Université de Mons

Faculté des Sciences

F.R.S.-FNRS

Distribution-Free and Calibrated Predictive Uncertainty in Probabilistic Machine Learning

Victor Dheur

A dissertation submitted in fulfillment of the requirements of the degree of *Docteur en Sciences*

December 2025

Advisor

Prof. Souhaib Ben Taieb
Mohamed bin Zayed University of Artificial Intelligence, United Arab Emirates
University of Mons, Belgium

Co-Advisor

Prof. Stéphane Dupont
University of Mons, Belgium

Members of the Jury

Prof. Jef Wijsen
University of Mons, Belgium
Prof. Pierre Geurts
University of Liège, Belgium
Prof. Matteo Sesia
University of Southern California, USA

Abstract

Machine learning models are increasingly deployed in high-stakes domains such as healthcare and autonomous systems, where decisions carry significant risks. Probabilistic machine learning is valuable in these settings, as it quantifies predictive uncertainty, notably by generating probabilistic predictions. We focus on regression, where the goal is to predict one or more continuous outputs given a set of inputs. In this context, we consider two main forms of uncertainty representation: predictive distributions, which assign probabilities to possible output values, and prediction sets, which are designed to contain the true output with a pre-specified probability. For these predictions to be reliable and informative, they must be calibrated and sharp, i.e., statistically consistent with observed data and concentrated around the true value.

In this thesis, we develop distribution-free regression methods to produce calibrated and sharp probabilistic predictions using neural network models. We consider both single-output and the less-explored multi-output regression settings. Specifically, we develop and study recalibration, regularization, and conformal prediction (CP) methods. The first adjusts predictions after model training, the second augments the training objective, and the last produces prediction sets with finite-sample coverage guarantees.

For single-output regression, we conduct a large-scale experimental study to provide a comprehensive comparison of these methods. The results reveal that post-hoc approaches consistently achieve superior calibration. We explain this finding by establishing a formal link between recalibration and CP, showing that recalibration also benefits from finite-sample coverage guarantees. However, the separate training and recalibration steps typically lead to degraded negative log-likelihood. To address this issue, we develop an end-to-end training procedure that incorporates the recalibration objective directly into learning, resulting in improved negative log-likelihood while maintaining calibration.

For multi-output regression, we conduct a comparative study of CP methods and introduce new classes of approaches that offer novel trade-offs between sharpness, compatibility with generative models, and computational efficiency. A key challenge in CP is achieving conditional coverage, which ensures that coverage guarantees hold for specific inputs rather than only on average. To address this, we propose a method that improves conditional coverage using conditional quantile regression, thereby avoiding the need to estimate full conditional distributions. Finally, for tasks requiring a full predictive density, we introduce a recalibration technique that operates in the latent space of invertible generative models such as conditional normalizing flows. This approach yields an explicit, calibrated multivariate probability density function. Collectively, these contributions advance the theory and practice of uncertainty quantification in machine learning, facilitating the development of more reliable predictive systems across diverse applications.

Acknowledgment

This thesis concludes a four-year endeavor, during which I have had the pleasure of working with many remarkable people.

I would first like to express my sincere gratitude to my supervisor, Souhaib Ben Taieb. His guidance and continuous support have been fundamental throughout this journey. Under his supervision, I have learned extensively about research, writing, and presentation. I am also thankful for his encouragement to pursue a PhD.

I would like to thank the members of my jury, Stéphane Dupont, Jef Wijsen, Pierre Geurts, and Matteo Sesia, for accepting this role and for dedicating time to examine my thesis. My thanks also go to my yearly committee members, Xavier Siebert, Benoît Frénay, and Pierre Geurts, for their useful and thoughtful advice over the years.

This research benefited from interactions with several collaborators. I am grateful for the opportunity to have worked with Matteo Fontana, Rafael Izbicki, Maxim Panov, Eric Moulines, Vincent Plassier, and Alexander Fishkov.

I also thank the members of the BDML lab at the University of Mons for the many great moments shared during group lunches, summer schools, and conferences. A special thanks to Tanguy and Sukanya for their companionship throughout these four years. My thanks also go to Yorick and Naomi, whom I had the opportunity to supervise and later work with as colleagues, as well as David and Hien. I also thank Faruk, Jad, Vinamr, Elnura, and Nabil for the shared adventures during my stay at MBZUAI.

More broadly, I would like to thank the professors of the UMONS computer science department, who indirectly contributed to this thesis by shaping my scientific curiosity and rigor. I am also grateful to the many people I met at conferences, during university courses, and at programming contests for their kindness and the memorable moments we shared.

Finally, I would like to thank my long-lasting friends for their support and the necessary distractions throughout these years. My deepest gratitude goes to my family, whose support began long before this thesis.

Contents

Abbreviations

CDF	Cumulative distribution function
QF	Quantile function
PDF	Probability density function
PMF	Probability mass function
PIT	Probability integral transform
UQ	Uncertainty quantification
MDN	Mixture density networks
DRF	Distributional random forest
KDE	Kernel density estimation
NF	Normalizing flow
ARFlow	Autoregressive normalizing flow
MQF^2	Multivariate quantile function forecaster
FM	Flow matching
NLL	Negative log-likelihood
QS	Quadratic score
PSS	Pseudo-spherical score
SS	Spherical score
CRPS	Continuous ranked probability score
HS	Hyvärinen score
KS	Kernel score
ES	Energy score
MMD	Maximum mean discrepancy
SQR	Simultaneous quantile regression
QR	Quantile recalibration
QREG	Quantile regularization
CP	Conformal prediction
SCP	Split conformal prediction
AR	Absolute residuals
CQR	Conformalized quantile regression
DCP	Distributional conformal prediction
HDR	Highest density region
HPD	Highest posterior density

C-HDR	Conformal HDR
PCE	Probabilistic calibration error
CCE	Conditional coverage error
WSC	Worst slab coverage
ACC	Asymptotic conditional coverage
TPP	Temporal point process
MTPP	Marked temporal point process
MCIF	Marked conditional intensity function
DR-CP	Density-Rank conformal prediction
STDQR	Spherically-transformed deep quantile regression
PCP	Probabilistic conformal prediction
HD-PCP	Highest-Density PCP
RCP	Rectified conformal prediction
HDR-R	HDR recalibration
LR	Latent recalibration

Notation

Common mathematical symbols

$[K]$	Set of integers $\{1, \dots K\}$
$\lvert A \rvert$	Cardinality (if A is finite) or Lebesgue measure (if A is measurable)
2^A	Powerset of A
$x^\top$	Transpose of vector or matrix x
$\lVert x \rVert$	Euclidean norm of x
$\langle x, y \rangle$	Euclidean inner product
I_d	$d \times d$ identity matrix
$\nabla_x y$	Gradient of y w.r.t. x
$\log(x)$, $\exp(x)$	Natural logarithm and exponential of x
$\odot$, $\oslash$	Elementwise product and division
$\arg\min_{\theta \in \Theta} \mathcal{R}(\theta)$	Set of minimizers of $\mathcal{R}$ over Θ

Probability

$\mathcal{X}, \mathcal{Y}$	Input and output spaces
p, d	Dimensions of $\mathcal{X}$ and $\mathcal{Y}$, respectively
X, Y	Input and output random variables
Y_i	Y indexed at dimension $i \in [d]$
$\mathcal{P}(\mathcal{Y})$	The set of probability distributions over $\mathcal{Y}$
P_X, P_Y	Marginal probability distributions of X and Y, respectively
$P_{X,Y}$	Joint probability distribution of X and Y
$P_{Y \mid X}$	Probability distribution of Y conditional on X
f_Y, p_Y, F_Y, Q_Y	PDF, PMF, CDF, and QF of Y
$f_{Y \mid X}, p_{Y \mid X}, F_{Y \mid X}, Q_{Y \mid X}$	PDF, PMF, CDF, and QF of Y conditional on X While f_Y is a PDF, $f_{Y \mid X}$ is a PDF-valued random variable; $f_{Y \mid X = x}$ is the (non-random) PDF of Y given $X = x$
$\mathcal{U}(0, 1)$	Standard uniform probability distribution
U	Standard uniform random variable
$\mathbb{E}_X[f(X)]$	Expectation of $f(X)$ w.r.t. X
$\mathbb{V}_X[f(X)]$	Variance of $f(X)$ w.r.t. X
$\mathbb{E}[f(X)]$	Expectation of $f(X)$ w.r.t. all random variables

$\mathbb{V}[f(X)]$	Variance of $f(X)$ w.r.t. all random variables
$X \sim P$	X is distributed according to P
$X \perp\!\!\!\perp Y$	X and Y are independent
$\overset{\text{d.}}{=}$	Equality in distribution
$\overset{\text{a.s.}}{=}$	Almost sure equality
$\mathbb{1}(E)$	Indicator function for the event E
$\mathcal{N}\left(x; \mu, \sigma^2\right)$	Univariate normal PDF at x with mean μ and variance σ^2
$\mathcal{N}\left(x; \mu, \Sigma\right)$	Multivariate normal PDF at x with mean μ and covariance Σ
$D_{\mathrm{KL}}\left(P \,\middle\|\, \hat{P}\right)$	Kullback-Leibler divergence of P w.r.t. $\hat{P}$

Supervised learning

$\mathcal{D}$	Full dataset
$\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{cal}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{test}}$	Training, calibration, validation, and test datasets
N	Cardinality of $\mathcal{D}_{\text{train}}$
n	Cardinality of $\mathcal{D}_{\text{cal}}$
$(X^{(i)}, Y^{(i)})$	i^{th} input-output pair (dataset clear from context)
$\mathcal{S}$	Prediction space
$\mathcal{H}$	Hypothesis space
Θ	Parameter space
h_θ	Parametric predictor with parameters $\theta \in \Theta$
$\hat{P}_{Y\|X}, \hat{f}_{Y\|X}, \hat{F}_{Y\|X}, \dots$	Estimates of $P_{Y\|X}, f_{Y\|X}, F_{Y\|X}, \dots$
$\hat{Y}$	Random variable with distribution $\hat{P}_{Y\|X}$
$\mathcal{L}$	Loss function
$\mathcal{R}$	Expected risk
$\hat{\mathcal{R}}$	Empirical risk
S	Scoring rule
$H(P)$	Generalized entropy of P induced by a scoring rule S
$D(\hat{P}, P)$	Divergence between P and $\hat{P}$ induced by a scoring rule S
ρ_α	Check function

Generative models

M	Number of mixture components
$\hat{\pi}_m, \hat{\mu}_m, \hat{L}_m, \hat{\Sigma}_m$	Weight, mean, Cholesky factor and covariance of component m
$\mathcal{Z}$	Latent space
Z	Latent random variable
$\hat{T}, \hat{T}^{-1}$	Conditional bijective transformation $\hat{T} : \mathcal{Z} \times \mathcal{X} \to \mathcal{Y}$ and its inverse
$\hat{v}$	Vector field

Calibration

α	Quantile level or miscoverage level $\alpha \in [0, 1]$
g	Pre-rank
G	Pre-rank random variable $G = g(X, Y)$ for the true output
$\hat{G}$	Pre-rank random variable $\hat{G} = g(X, \hat{Y})$ for the estimated output
$\hat{U}$	PIT random variable

$F_{\hat{U}}$	CDF of $\hat{U}$, corresponding to the ideal calibration map
$\Phi_{\text{EMP}}, \Phi_{\text{LIN}}, \Phi_{\text{KDE}}$	Empirical, linear and KDE-based calibration maps
$\text{HDR}_f(1-\alpha)$	HDR at level $1-\alpha$ under the PDF f
$\text{HPD}_f(y)$	HPD in y under the PDF f
λ	Regularization coefficient
β	Regularization coefficient for QRT
K, L	Number of samples from a model

Conformal prediction

s	Conformity score
$\hat{q}$	$(1-\alpha)$ empirical quantile of conformity scores on $\mathcal{D}_{\text{cal}}$
$\hat{R}(x)$	Prediction set for the input x
α_l, α_u	Lower and upper quantile levels
$\hat{l}_i(x), \hat{u}_i(x)$	Estimates of the α_l- and α_u-quantiles of $Y_i \vert X = x$

Decision-making

$\mathcal{A}$	Action space
δ	Decision rule
l	Decision loss
γ	Reliability gap

Marked temporal point processes

$\hat{R}_\tau$	Prediction set for the time
$\hat{R}_k$	Prediction set for the mark
t	Arrival time random variable
τ	Inter-arrival time random variable
k	Mark random variable
$\boldsymbol{h}$	History random variable
$\hat{f}(\tau, k \vert \boldsymbol{h})$	Joint density of inter-arrival time τ and mark k conditional on the history $\boldsymbol{h}$
λ_k^*	MCIF
Λ_k^*	Cumulative MCIF

Latent recalibration

$\rho_{\mathcal{Z}}$	Norm over $\mathcal{Z}$
$l_{\hat{T}}$	Latent norm w.r.t. $\hat{T}$
χ_d	Chi distribution with d degrees of freedom
r	Norm recalibration function
R	Radial transformation

Rectified conformal prediction

$\mathbb{T}$	Space of conformity scores (w.r.t. a conformity score s)
$\mathbf{s}$	Random variable $\mathbf{s} = s(X, Y)$
φ	Adjustment constant
f_t	Adjustment function parameterized by t

$\tilde{f}_v$	Adjustment function parameterized as $\tilde{f}_v(t) = f_t(v)$
$s_\varphi, \mathbf{s}_\varphi$	Adjusted score, and $\mathbf{s}_\varphi = s_\varphi(X, Y)$
$\tau_\star$	Conditional quantile of the adjusted scores (shorthand for $Q_{\mathbf{s}_\varphi \mid X}$)
$\tilde{s}_\star$, $\tilde{\mathbf{s}}_\star$	Rectified conformity score based on $\tau_\star$, and $\tilde{\mathbf{s}}_\star = \tilde{s}_\star(X, Y)$
$\mathcal{D}_\tau$	Dataset for conditional quantile estimation
$\hat{Q}_{\mathbf{s} \mid X}$	Estimate of the conditional quantile of the conformity scores
$\hat{\tau}$	Estimate of $\tau_\star$ (shorthand for $\hat{Q}_{\mathbf{s}_\varphi \mid X}$)
$\tilde{s}$, $\tilde{\mathbf{s}}$	Rectified conformity score based on $\hat{\tau}$, and $\tilde{\mathbf{s}} = \tilde{s}(X, Y)$
ϵ_τ	Conditional coverage error relative to the target level $1 - \alpha$.

CHAPTER 1

Introduction

Machine learning (ML) has become a central scientific discipline for building data-driven predictive systems. At its core, ML is the science of developing algorithms that can learn from data and generalize to unseen data. The increasing complexity and capacity of modern ML models, particularly neural networks (NNs), have enabled them to capture complex patterns in data, leading to unprecedented predictive accuracy on a wide range of tasks (Bommasani et al., 2021). In the context of regression, where the output variable is continuous, these models typically return a point prediction, a single value that does not quantify uncertainty. In contrast, probabilistic ML provides a more complete view by generating probabilistic predictions (Ghahramani, 2015; Murphy, 2022). This is necessary for optimal decision-making, exploration, and in settings where decisions carry tangible costs such as medical prognosis, financial risk assessment, and energy consumption forecasting (Abdar et al., 2021).

The field of uncertainty quantification (UQ) provides principles and tools to equip ML models with the ability to quantify the inherent randomness in the data as well as their own ignorance. A comprehensive UQ framework involves understanding sources of uncertainty (Hüllermeier and Waegeman, 2021; Gruber et al., 2023), learning representations of uncertainty (Gneiting and Katzfuss, 2014; Murphy, 2023) and quantifying uncertainty (Gneiting and Resin, 2023). Probabilistic predictions typically take two complementary forms. Predictive distributions assign a probability to each set of target values, a representation necessary for decision-makers who need to optimize utility for a downstream task or perform risk analysis. In contrast, prediction sets provide a more directly interpretable set of plausible target values designed to contain the true value with a pre-specified probability.

A predictive distribution is accurate and informative if it is calibrated and sharp. Calibration refers to the statistical consistency between the predictive distributions and the observations and can be defined according to different notions (Gneiting et al., 2007; Gneiting and Resin, 2023). The topic has gained renewed attention with the discovery that modern NN classifiers tend to be systematically miscalibrated and overconfident (Guo et al., 2017; C. Wang, 2025). *Probabilistic calibration* is an important notion which requires that prediction intervals achieve their stated coverage at all nominal levels on average. For instance, 90% prediction intervals should contain

the true outcome approximately 90% of the time. Sharpness refers to the concentration or informativeness of the predictive distributions and is a property of the prediction only. Proper scoring rules (Gneiting and Raftery, 2007; Waghmare and Ziegel, 2025) jointly assess calibration and sharpness, making them a standard tool for evaluating predictive distributions. However, by providing a single aggregate measure of performance, they do not provide insights into the sources of misspecification.

In this thesis, we study and develop algorithms to produce sharp, calibrated, and computationally efficient probabilistic predictions in NN regression. Our investigation centers on two families of principled and complementary methods: recalibration and conformal prediction (CP). A key attribute of these approaches is that they are distribution-free, meaning they do not rely on strong parametric assumptions about the underlying data-generating process. This generality has made them increasingly relevant for reliable decision-making across diverse domains, including smart grids (Arpogaus et al., 2023; Trotta et al., 2024), healthcare (Liou et al., 2024; Gamble et al., 2025; Yang et al., 2024), autonomous driving (J. Sun et al., 2023; Cao et al., 2024), and weather forecasting (Price et al., 2025; Allen et al., 2023). While our methods are broadly applicable, we primarily focus on tabular regression problems, progressing from the well-understood context of single-output regression to the more complex and less-explored domain of multi-output regression, where capturing dependencies between outputs is essential.

Recalibration aims to adjust the predictive distributions of a pre-trained model to correct statistical biases. A notable example is quantile recalibration (Kuleshov et al., 2018), which relies on a held-out dataset to improve probabilistic calibration. In parallel, CP provides a framework for constructing prediction sets with rigorous, finite-sample coverage guarantees (Vovk et al., 1999). For example, a 90% CP set is guaranteed to contain the true outcome with probability at least 90%. Importantly, in addition to developing new methods, we clarify connections between recalibration and CP. The following sections outline the specific challenges this thesis addresses and summarize our contributions.

1.1. Research Challenges and Contributions

Research Contribution 1 (Chapter 3) While NN miscalibration has been documented in classification settings (Guo et al., 2017; Minderer et al., 2021), there has been less research on probabilistic calibration for NN regression. A variety of methods have been proposed to improve their reliability. These methods broadly fall into three categories: post-hoc recalibration (e.g., quantile recalibration (Kuleshov et al., 2018)), CP (e.g., conformalized quantile regression (Romano et al., 2019)), and regularization (e.g., quantile regularization (Utpala and Rai, 2020)). It remains unclear how they compare in balancing probabilistic calibration with sharpness. Additionally, the calibration and CP literature have historically been studied separately despite their similarities, and thus theoretical links between them are not well established. Our first contribution addresses these issues by conducting a large-scale empirical study of probabilistic calibration on 57 tabular regression datasets. We also introduce novel differentiable recalibration and regularization objectives based on kernel density estimation, which provide further insights into the performance of these approaches. A finding of our study is the formal connection between recalibration and CP; we establish conditions under which quantile recalibration can be seen as a special case of CP, which helps explain their strong performance in improving probabilistic

calibration.

Research Contribution 2 (Chapter 4) Our initial study showed that post-hoc recalibration methods generally outperform regularization techniques in improving probabilistic calibration. However, the post-hoc step is completely independent of model training. This two-stage process is suboptimal, as the model is trained to jointly optimize for sharpness and calibration without any awareness of the final recalibration step (D.-B. Wang et al., 2021). Our second contribution introduces an end-to-end training procedure for NNs, quantile recalibration training (QRT), which integrates post-hoc calibration directly into the training process. QRT minimizes the negative log-likelihood (NLL) to encourage predictive accuracy, while ensuring the model's predictive distributions are probabilistically calibrated. This approach captures the paradigm of minimizing sharpness subject to calibration (Gneiting et al., 2007). This approach improves upon standard training, regularization and post-hoc baselines (Utpala and Rai, 2020; Kuleshov et al., 2018; Dheur and Ben Taieb, 2023) without introducing additional model parameters. Through a large-scale experiment, we show that QRT achieves improved NLL while maintaining the state-of-the-art calibration of post-hoc recalibration.

Research Contribution 3 (Chapter 5) Many real-world problems require predicting multiple, often dependent, variables. However, the extension of CP to multi-output tasks presents additional challenges, including complex output dependencies and high computational costs, and remains relatively underexplored. Simple approaches that combine univariate regions fail to capture dependencies between variables, leading to conservative and inflexibly shaped regions (Y. Zhou et al., 2024). Conversely, more sophisticated sample-based techniques can create sharper regions but suffer from high computational costs (Z. Wang et al., 2023). For our third contribution, we conduct a unified comparative study of nine conformal methods for constructing multivariate prediction regions, examining their properties and interconnections. We also introduce two classes of conformity scores that generalize their univariate counterparts. These scores ensure conditional coverage asymptotically while maintaining finite-sample marginal coverage. One class is compatible with any generative model, offering broad applicability, while the other is computationally efficient, leveraging the properties of invertible generative models. Finally, we conduct a large-scale study on 13 tabular datasets and one image dataset, comparing the different multi-output conformal methods across different sharpness and conditional coverage metrics, and different base predictors (multivariate quantile function forecaster (Kan et al., 2022), distributional random forests (Cevid et al., 2022), and Cholesky-based mixture density networks (Muschinski et al., 2022)). Additionally, we explore the construction of a joint prediction set for a bivariate output in the context of neural temporal point processes (Shchur et al., 2021).

Research Contribution 4 (Chapter 6) A primary limitation of CP is that it provides a guarantee that is only marginal, i.e., averaged over all inputs, and thus may undercover for specific subgroups of the data. While constructing prediction sets with exact conditional coverage is infeasible without additional distributional assumptions (Foygel Barber et al., 2021b), attempts to approximate it face their own issues. Methods based on partitioning the covariate space can lead to overly large prediction sets (Bian and Barber, 2023), while others require estimating the full conditional distribution of conformity scores, which is both computationally intensive and difficult to perform accurately (Izbicki et al., 2022). Our fourth contribution is to introduce

Rectified conformal prediction (RCP), a conformal method designed to enhance conditional coverage by refining conformity scores. RCP avoids the need to estimate the full conditional distribution of a multivariate response, relying instead on estimating only the conditional quantile of a univariate conformity score. By constructing a new conformity score whose quantile at a given coverage level is independent of inputs, RCP enhances conditional coverage while preserving marginal coverage guarantees. We provide a theoretical lower bound on the conditional coverage of the prediction sets generated by RCP as a function of the approximation error in estimating the conditional quantile of the conformity score distribution. Finally, we evaluate RCP against several conformal baselines based on point, interval, and distribution estimates, demonstrating improved performance, particularly in terms of conditional coverage metrics.

Research Contribution 5 (Chapter 7) For many downstream tasks such as risk assessment or decision-making under uncertainty, a full probability density function (PDF) is more useful than a prediction set (N. Klein, 2024). However, despite generating multivariate prediction sets with coverage guarantees, CP does not provide a full PDF. At the same time, recalibration methods are primarily limited to single-output settings, with the exception of HDR recalibration (Y. Chung et al., 2024). Yet, HDR recalibration has notable limitations: it does not yield an explicit predictive PDF and relies on computationally intensive sampling and binning at test time. To address this gap, our final contribution first introduces a notion of *latent calibration*, which assesses the probabilistic calibration of a conditional normalizing flow within its latent space. Based on this concept, we propose latent recalibration (LR), a post-hoc method that learns a transformation of the latent space to achieve finite-sample guarantees on latent calibration, with connections to CP (Fang et al., 2025; Dheur et al., 2025). An advantage of LR is that, unlike set-based or sampling-based approaches, it produces a recalibrated generative model with an explicit and computationally efficient multivariate PDF. We demonstrate empirically across 29 multi-output tabular datasets and one high-dimensional image dataset that LR consistently improves latent calibration, while also improving or maintaining the NLL.

1.2. Research Publications

The content of Chapters 3 to 7 is largely derived from research papers published in international peer-reviewed conferences and journals. For each publication, we provide a link to a public repository containing the codebase, enabling the reproduction of the experiments presented in this thesis.

- Chapter 3: A Study of Probabilistic Calibration in Neural Probabilistic Models
  - **Victor Dheur** and Souhaib Ben Taieb (2023). A Large-Scale Study of Probabilistic Calibration in Neural Network Regression. *The 40th International Conference on Machine Learning.*
    - **Repository**: `https://github.com/Vekteur/probabilistic-calibration-study`
- Chapter 4: Probabilistic Calibration by Design
  - **Victor Dheur** and Souhaib Ben Taieb. Probabilistic Calibration by Design for

Neural Network Regression (2024). *The 27th International Conference on Artificial Intelligence and Statistics.*

  * **Repository**: `https://github.com/Vekteur/quantile-recalibration-training`

- Chapter 5: Multi-Output Conformal Regression
  - **Victor Dheur**, Matteo Fontana, Yorick Estievenart, Naomi Desobry, and Souhaib Ben Taieb (2025). A Unified Comparative Study with Generalized Conformity Scores for Multi-Output Conformal Regression. *The 42nd International Conference on Machine Learning.*

    * **Repository**: `https://github.com/Vekteur/multi-output-conformal-regression`

  - **Victor Dheur***, Tanguy Bosser*, Rafael Izbicki, and Souhaib Ben Taieb (2024). Distribution-Free Conformal Joint Prediction Regions for Neural Marked Temporal Point Processes. *Machine Learning, Volume 113.*

    * **Repository**: `https://github.com/tanguybosser/conf_tpp`

- Chapter 6: Rectifying Conformity Scores
  - Vincent Plassier*, Alexander Fishkov*, **Victor Dheur***, Mohsen Guizani, Souhaib Ben Taieb, Maxim Panov, and Eric Moulines (2025). Rectifying Conformity Scores for Better Conditional Coverage. *The 42nd International Conference on Machine Learning.*

    * **Repository**: `https://github.com/stat-ml/rcp`

- Chapter 7: Latent Recalibration
  - **Victor Dheur**, Souhaib Ben Taieb (2025). Multivariate Latent Recalibration for Conditional Normalizing Flows. *The 39th Annual Conference on Neural Information Processing Systems.*

    * **Repository**: `https://github.com/Vekteur/latent-recalibration`

*Equal contribution

CHAPTER 2

Background

This chapter provides the foundational concepts essential for understanding the thesis and places them in their broader context. It establishes the mathematical and conceptual framework for reliable uncertainty quantification in probabilistic ML, progressing from the general principles of supervised learning to the specialized topics central to our investigation: calibration and conformal prediction.

We begin in Section 2.1 by defining the supervised learning problem and reviewing neural networks. Section 2.2 introduces the core principles of uncertainty quantification: its sources, representations, and key methods. Section 2.3 covers proper scoring rules, allowing both estimation and evaluation of predictive distributions. The subsequent sections dive into the two main axes of this work: Section 2.4 details the theory of calibration for assessing statistical consistency, while Section 2.5 introduces conformal prediction for constructing prediction sets with finite-sample coverage. Finally, Section 2.6 discusses connections between calibration and decision-making.

2.1. Machine Learning

Machine learning (ML) is a scientific discipline concerned with the development and study of algorithms that learn from data (Abu-Mostafa et al., 2012; Goodfellow et al., 2016; Murphy, 2023). Essentially, these algorithms build models that generalize beyond the specific data on which they were trained.

More broadly, ML is one of the core approaches within the field of artificial intelligence (AI), which aims to build systems capable of performing tasks that normally require human intelligence, such as reasoning, perception, or decision-making (Russell and Norvig, 2020). In this sense, ML provides the data-driven foundations that enable many modern AI applications.

ML tasks fall under different paradigms, the main ones being supervised learning, unsupervised learning, and reinforcement learning. In this thesis, we focus on supervised learning, in which the learning algorithm has access to a dataset consisting of paired *inputs* (also called features, covariates, or predictors) and *outputs* (also called targets, outcomes, responses, or labels).

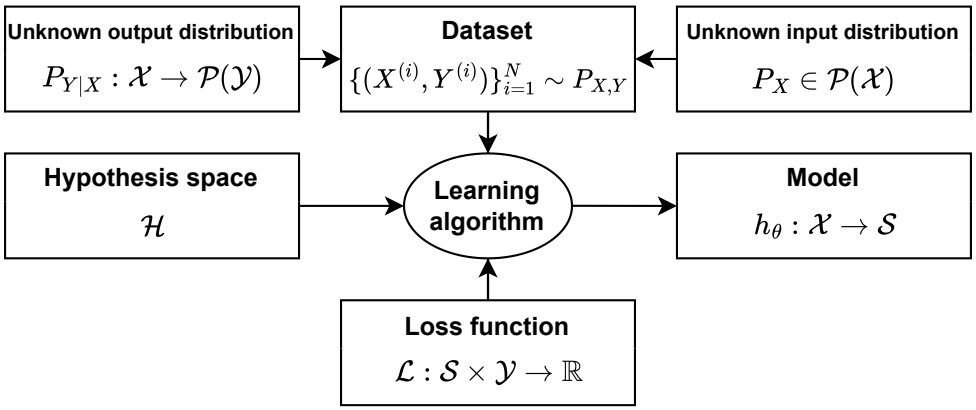


Figure 2.1: The learning diagram, adapted from Abu-Mostafa et al. (2012).

Section 2.1.1 introduces the formal framework of supervised learning, summarized in Figure 2.1, and Section 2.1.2 provides a brief overview of neural networks. For readers seeking a comprehensive overview, several foundational texts offer different perspectives. The recent works by Murphy (2022) and Murphy (2023) and the classic by Bishop (2006) provide a thorough introduction to probabilistic ML. For a perspective rooted more in statistical foundations, see Hastie (2009).

2.1.1 Supervised Learning

Inputs and outputs are represented as random variables X and Y taking values in spaces $\mathcal{X}$ and $\mathcal{Y}$. When $\mathcal{Y} = [K] = \{1, \dots, K\}$ is a finite set of K classes, the task is called *classification*. When $\mathcal{Y} \subseteq \mathbb{R}^d$, the task is called *regression*, and more precisely single-output regression when $d = 1$ and multi-output regression when $d > 1$. We assume that the pair (X, Y) follows an unknown joint distribution $P_{X,Y} = P_X P_{Y|X}$, decomposed into the input distribution P_X and the conditional output distribution $P_{Y|X}$. The space of distributions over a set A is denoted $\mathcal{P}(A)$. The distributions P_X and $P_{X,Y}$ belong to $\mathcal{P}(\mathcal{X})$ and $\mathcal{P}(\mathcal{X} \times \mathcal{Y})$, respectively. The conditional distribution $P_{Y|X}$ is interpreted as distribution-valued random variable whose value depends on X. Equivalently, $P_{Y|X}$ is a mapping $\mathcal{X} \to \mathcal{P}(\mathcal{Y})$ with $P_{Y|X=x} \in \mathcal{P}(\mathcal{Y})$.

The learning algorithm does not have access to $P_{X,Y}$ and instead observes a finite dataset $\mathcal{D}$ of input-output pairs independently sampled from $P_{X,Y}$:

$$\mathcal{D} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^{|\mathcal{D}|} \text{ where } (X^{(i)}, Y^{(i)}) \overset{\text{i.i.d.}}{\sim} P_{X,Y}. \tag{2.1}$$

Hypothesis space. In parametric learning, a model is represented by a hypothesis h_θ, a function governed by a set of parameters θ from a parameter space Θ. Each hypothesis maps an input $x \in \mathcal{X}$ to a prediction in a space $\mathcal{S}$. For example, $\mathcal{S}$ can be the target space $\mathcal{Y}$ for point predictions, or the space of probability distributions over $\mathcal{Y}$, denoted $\mathcal{P}(\mathcal{Y})$, for probabilistic predictions. The algorithm's search is confined to the hypothesis space $\mathcal{H}$, which is the set of all functions accessible by varying the parameters θ:

$$\mathcal{H} = \{\, h_\theta : \mathcal{X} \to \mathcal{S} \mid \theta \in \Theta \,\}. \tag{2.2}$$

Risk minimization. A loss function $\mathcal{L} : \mathcal{S} \times \mathcal{Y} \to \mathbb{R}$ evaluates predictions $h_\theta(x)$ given a realization y according to a real value $\mathcal{L}(h_\theta(x), y)$. The performance of a hypothesis h_θ is measured by its (expected) risk, defined as the expected loss over the data distribution:

$$\mathcal{R}(h_\theta) = \mathbb{E}[\mathcal{L}(h_\theta(X), Y)]\,. \tag{2.3}$$

Unless stated otherwise, expectations are over all random variables (here, X and Y). In univariate regression, a common loss function is the mean squared error $\mathcal{L}(\hat{y}, y) = (\hat{y} - y)^2$ for $\hat{y} = h_\theta(x) \in \mathbb{R}$. When $h_\theta(x) \in \mathcal{P}(\mathcal{Y})$ is a predictive distribution, $\mathcal{L}$ is known as a scoring rule (see Section 2.3).

Parameters yielding the minimum risk are denoted $\theta^* \in \arg\min_{\theta \in \Theta} \mathcal{R}(h_\theta)$. Since $P_{X,Y}$ is unknown, the learning algorithm seeks to minimize the empirical risk

$$\hat{\mathcal{R}}(h_\theta) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \mathcal{L}(h_\theta(X^{(i)}), Y^{(i)}) \tag{2.4}$$

as an approximation of the true risk. Parameters yielding the minimum empirical risk are denoted $\hat{\theta}^* \in \arg\min_{\theta \in \Theta} \hat{\mathcal{R}}(h_\theta)$.

Approximation-generalization trade-off. A key consideration in machine learning is the approximation-generalization trade-off. In order to find a hypothesis h with low risk $\mathcal{R}(h)$, the learning algorithm searches for parameters $\hat{\theta}$ ensuring two properties:

- $h_{\hat{\theta}}$ must generalize well: $\mathcal{R}(h_{\hat{\theta}}) \approx \hat{\mathcal{R}}(h_{\hat{\theta}})$.
- $h_{\hat{\theta}}$ must have a low empirical risk $\hat{\mathcal{R}}(h_{\hat{\theta}})$.

This trade-off is closely linked to the choice of the hypothesis space $\mathcal{H}$. If $\mathcal{H}$ is too limited, no hypothesis may be able to adequately capture the patterns in the data, a situation known as *underfitting*. Conversely, if $\mathcal{H}$ is too flexible, the model might learn the training data too well, including its noise, resulting in a low empirical risk but poor performance on new data. This phenomenon is known as *overfitting* (Vapnik, 1999). Finding a balance between these competing objectives is a central aspect of machine learning.

Machine learning models. A variety of model families have been developed to instantiate the hypothesis space $\mathcal{H}$. Classical approaches include linear models and kernel methods, which remain central due to their statistical foundations and efficiency (Hastie, 2009). For problems requiring expressive models, tree-based methods and NNs are widely used.

Tree-based methods partition the input space into regions that are easy to interpret and manipulate. Decision trees (Breiman et al., 1984) are simple yet expressive models that form the basis of more advanced ensembles. Random forests (Breiman, 2001) improve stability and accuracy by averaging predictions over many randomized trees, while boosting methods (Freund and Schapire, 1997) combine weak learners sequentially to reduce errors.

Neural networks (Goodfellow et al., 2016) constitute another major family. They are built from layers of non-linear transformations and have demonstrated remarkable performance across domains. A distinguishing feature of NNs is the diversity of architectures tailored to different modalities, including convolutional networks for images, recurrent and transformer models for

text, and multimodal architectures that integrate heterogeneous data. Importantly for this thesis, they are also very flexible in how they can parametrize distributions (see Section 2.2.3).

2.1.2 Neural Networks

This section offers a brief overview of NNs; it introduces no essential notation and may be skipped on first read. An in-depth treatment is provided by Goodfellow et al. (2016).

Neural networks are powerful function approximators (Cybenko, 1989; Goodfellow et al., 2016) inspired by biological brains (McCulloch and Pitts, 1943). The term *deep learning* was popularized in the mid-2000s to describe NNs with many layers, following breakthroughs that made it practical to train them effectively (Hinton et al., 2006). Among the many architectures, a foundational one is the *multilayer perceptron* (MLP). An MLP is composed of $L+1$ layers of size $m^{(0)}, \dots, m^{(L)}$, where $m^{(0)} = p$ is the size of the input layer and $m^{(L)}$ is the size of the output layer. It computes representations $z^{(0)}, \dots, z^{(L)}$ with $z^{(0)} = x$ as the input and $z^{(L)} = h_\theta(x)$ as the network output.

At each layer $l \in \{1, \dots, L\}$, with weights $W^{(l)} \in \mathbb{R}^{m^{(l)} \times m^{(l-1)}}$ and biases $b^{(l)} \in \mathbb{R}^{m^{(l)}}$, the representation is computed using a linear combination followed by a non-linear transformation ζ called an *activation function*:

$$z^{(l)} = \zeta\left(b^{(l)} + W^{(l)} z^{(l-1)}\right). \tag{2.5}$$

If all activations are linear, then $z^{(L)} = h_\theta(x)$ is a linear function of x. To model complex non-linear relationships, non-linear activations are used. A common choice is the rectified linear unit (ReLU):

$$\zeta(t) = \max\{0, t\}. \tag{2.6}$$

During training, the NN updates its parameters $\theta = \left\{\left(W^{(l)}, b^{(l)}\right)\right\}_{l=1}^{L}$ to minimize the empirical risk $\hat{\mathcal{R}}\left(h_\theta\right)$. Let $\eta > 0$ be the *learning rate*. The parameters are optimized using *gradient descent* as follows

$$\theta \leftarrow \theta - \eta \nabla_\theta \hat{\mathcal{R}}\left(h_\theta\right). \tag{2.7}$$

until a stopping criterion is met (for example, a fixed number of passes over the data or no further improvement on a validation set). Gradients $\nabla_\theta \hat{\mathcal{R}}\left(h_\theta\right)$ are efficiently computed by *backpropagation* (Rumelhart et al., 1986).

In practice, $\hat{\mathcal{R}}\left(h_\theta\right)$ is often estimated on a random mini-batch of training examples at each step (stochastic gradient descent), and adaptive optimizers such as Adam (Kingma and Ba, 2015) together with learning-rate schedules are commonly used. *Early stopping* is a common regularization technique that helps avoid overfitting. It monitors the validation loss on a held-out dataset and stops training when this validation performance ceases to improve for a number of epochs.

2.2. Uncertainty Quantification

Uncertainty quantification (UQ) provides the mathematical and conceptual tools to characterize and manage the unknowns inherent in data-driven modeling. This work focuses on *predictive uncertainty*, which refers to the uncertainty associated with a model's prediction. A robust

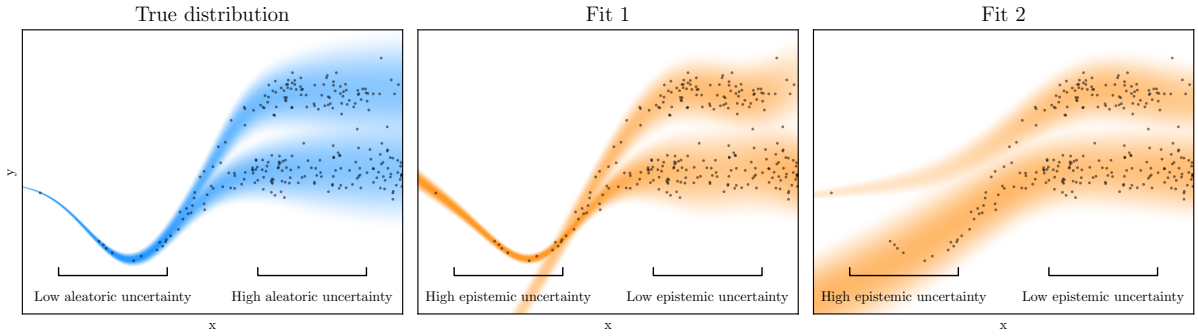


Figure 2.2: Illustrative example of the sources of uncertainty in a prediction task. For high x, the aleatoric uncertainty (inherent to the data) is high. For low x, the epistemic uncertainty is high due to lack of data and variations between model fits 1 and 2.

understanding of uncertainty is essential for building reliable ML systems. This section lays the groundwork for the thesis by first dissecting the fundamental sources from which uncertainty arises, surveying common representations of uncertainty, and finally presenting several UQ methods.

2.2.1 Sources of Uncertainty

To effectively characterize predictive uncertainty, it is essential to understand the sources from which it can arise. A fundamental distinction in ML is between *aleatoric* and *epistemic* uncertainty (Hüllermeier and Waegeman, 2021; Gruber et al., 2023). These two uncertainty types represent fundamentally different aspects of uncertainty.

Aleatoric and epistemic uncertainty. Aleatoric uncertainty stems from inherent randomness in the data-generating process $P_{Y|X}$. It is considered irreducible because even a perfect model with infinite data could not eliminate it. In contrast, epistemic uncertainty, or model uncertainty, arises from the model's limited knowledge of the underlying process, due to factors such as insufficient training data or an inadequate model specification. This type of uncertainty is, in principle, reducible by providing more data or improving the model.

Example 1. Figure 2.2 illustrates the interplay between aleatoric and epistemic uncertainty, where predictive uncertainty is represented by a conditional probability density function (PDF). The PDF is represented as a shading where darker colors correspond to a higher PDF values. The leftmost panel displays the true conditional PDF $f_{Y|X}$ (blue shading), from which the training data (black dots) are sampled. The true distribution itself exhibits varying levels of randomness. For small x, the distribution is unimodal with low variance, corresponding to low aleatoric uncertainty. In contrast, for large x, the distribution is bimodal with high variance, indicating high aleatoric uncertainty.

The center and right panels depict the predictive distributions of two different models, $\hat{f}^{(1)}_{Y|X}$ and $\hat{f}^{(2)}_{Y|X}$, trained on the same data. By comparing their behavior, we can diagnose the epistemic uncertainty:

- For high x values, the abundance of data strongly constrains any reasonable model. As a result, both $\hat{f}^{(1)}_{Y|X}$ and $\hat{f}^{(2)}_{Y|X}$ converge to nearly identical solutions that successfully capture the bimodal aleatoric uncertainty. This agreement between models is usually interpreted as low epistemic uncertainty.
- For small x (left of the figure), due to the lack of data, the models are free to extrapolate in different, albeit plausible, ways. This is visually represented by the stark disagreement between $\hat{f}^{(1)}_{Y|X}$ and $\hat{f}^{(2)}_{Y|X}$. Their divergence reveals high epistemic uncertainty in this region of the input space. An ideal uncertainty-aware model would not commit to a single hypothesis but would instead signal this high epistemic uncertainty. For example, Bayesian NNs consider $\theta \in \Theta$ as a random variable and learn a distribution $\hat{P}_\theta$ over the model parameters. Denoting a predictive PDF parameterized by $\omega \in \Theta$ as $\hat{f}^{(\omega)}_{Y|X}$, they produce a posterior predictive PDF $\hat{f}_{Y|X=x}(y) = \int_\Theta \hat{f}^{(\omega)}_{Y|X=x}(y)\hat{P}_\theta(d\omega)$ whose spread should ideally increase in case of higher epistemic uncertainty.

Aleatoric and epistemic uncertainty in practice. While the distinction between aleatoric and epistemic uncertainty is theoretically clear, its practical application can be ambiguous (Hüllermeier and Waegeman, 2021). For instance, if one assumes a fully deterministic universe (Laplace's demon), all uncertainty in real-world applications could theoretically be considered reducible and thus epistemic. In practice, however, aleatoric uncertainty is defined relative to a given set of observed variables, representing the variability that cannot be explained by them. Formally, aleatoric uncertainty can be identified with the true conditional distribution $P_{Y|X}$, while epistemic uncertainty is all the remaining uncertainty that arises in predictive modeling (Gruber et al., 2023). Epistemic uncertainty can be further divided into model uncertainty (corresponding to the choice of model family) and estimation uncertainty (corresponding to the correct estimation of model parameters).

UQ methods for epistemic uncertainty. Several methods aim to quantify and disentangle these uncertainties. Prominent approaches include Bayesian NNs (Jospin et al., 2022; Fortuin, 2022), Monte Carlo dropout (Yarin Gal and Ghahramani, 2016), and ensembling (Lakshminarayanan, Pritzel, et al., 2017; G. Huang et al., 2017). These methods often provide robust uncertainty estimates but come with significant computational overhead. Evidential deep learning (Sensoy, Kaplan, et al., 2018; Amini et al., 2020) has been proposed as a more efficient alternative. However, the statistical interpretation of the uncertainty estimates from this method has been debated (M. Shen et al., 2024).

In-distribution and out-of-distribution learning. A further distinction in predictive modeling is between *in-distribution* (ID) and *out-of-distribution* (OOD) learning scenarios (Moreno-Torres et al., 2012; Ye et al., 2021). In-distribution learning refers to the regime where both training and test data are assumed to be drawn from the same distribution, which is the setting we consider in Section 2.1.1. In this case, the dominant source of uncertainty is typically aleatoric, since the model is evaluated under conditions for which it has been trained (Kendall and Gal, 2017). In contrast, out-of-distribution learning arises when the test data distribution differs systematically from the training distribution. In this case, the risk w.r.t. the training data distribution differs from the risk w.r.t. the test data distribution. OOD scenarios are particularly

challenging because they introduce additional uncertainty that requires explicit mechanisms for detecting OOD data and robust generalization (Ovadia et al., 2019).

In this work, we restrict our focus to in-distribution learning, where the accurate modeling of aleatoric uncertainty is of primary importance. Epistemic uncertainty still plays a role in practice, especially in regions of the input space with scarce data, but capturing aleatoric uncertainty is the central objective in the setting considered here.

2.2.2 Representations of Uncertainty

To be useful, predictive uncertainty must be captured in a concrete mathematical form. The choice of representation depends on the task requirements and the desired level of granularity. This section introduces the notation for several key representations used throughout this work. In this section, since we only focus on the output representation and not the dependence on X, we consider unconditional quantities.

Predictive distributions

A standard way of representing predictive uncertainty is via a (probability) distribution $\hat{P}_Y \in \mathcal{P}(\mathcal{Y})$. Figure 2.3 presents representations of a specific discrete (first row) and continuous (second row) distribution. The first column shows the probability mass function (PMF) $\hat{p}_Y : \mathcal{Y} \to [0, 1]$ and PDF $\hat{f}_Y : \mathcal{Y} \to \mathbb{R}_+$. The second column shows the CDF $\hat{F}_Y : \mathcal{Y} \to [0, 1]$, which is defined both for discrete and continuous distributions as $\hat{F}_Y(y) = \mathbb{P}(Y \leqslant y)$. The third column shows the quantile function (QF) $\hat{Q}_Y : [0, 1] \to \mathcal{Y}$. Unless mentioned otherwise, it corresponds to the left-quantile function:

$$\hat{Q}_Y(\alpha) = \inf\{y \in \mathcal{Y} : \alpha \leqslant \hat{F}_Y(y)\}. \tag{2.8}$$

For some generative models like diffusion models (Ho et al., 2020) or generative adversarial networks (Goodfellow et al., 2014), such precise representations are not available. Instead, the predictive uncertainty is represented via K samples $\hat{Y}^{(1)}, \ldots, \hat{Y}^{(K)} \sim \hat{P}_Y$ from the estimated distribution. In the fourth column, $K = 50$ samples are represented as points on their empirical CDF.

Prediction sets

Prediction sets, denoted $\hat{R} \subseteq \mathcal{Y}$, represent an alternative way to represent predictive uncertainty. For a given nominal miscoverage level $\alpha \in (0, 1)$, prediction sets are desired to satisfy coverage, i.e., $\mathbb{P}(Y \in \hat{R}) \geqslant 1 - \alpha$ (Vovk et al., 2005). Multiple prediction sets can satisfy this property. In single-output regression, a well-known example is the equal-tailed interval (Brehmer and Gneiting, 2021):

$$\left[\hat{Q}_Y(\alpha/2), \hat{Q}_Y(1 - \alpha/2)\right]. \tag{2.9}$$

The fourth column of Figure 2.3 shows a prediction set containing the three values with the highest mass (first row) and an equal-tailed interval (second row).

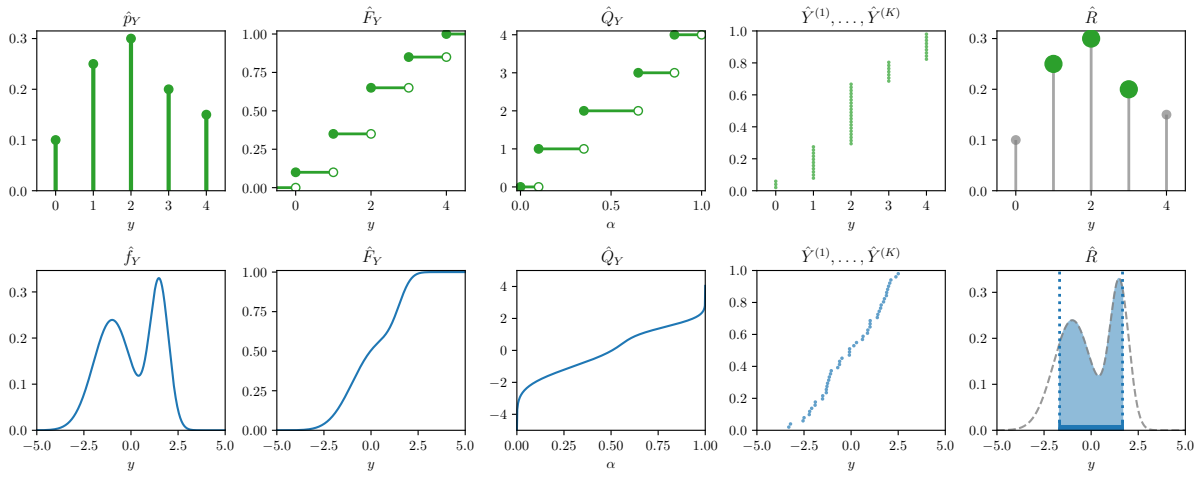


Figure 2.3: Standard representations of a discrete distribution (first row) and a continuous distribution (second row).

Scalar summaries

For certain tasks, a single scalar measure of predictive uncertainty is desired. Let $\hat{Y} \sim \hat{P}_Y$. Common choices include the variance $\mathbb{V}[\hat{Y}] = \mathbb{E}[(\hat{Y} - \mathbb{E}[\hat{Y}])^2]$, the (differential) entropy $H(\hat{Y}) = -\mathbb{E}[\log \hat{f}_Y(\hat{Y})]$, or the volume of a prediction set $|\hat{R}|$ (Hüllermeier and Waegeman, 2021). Section 2.3.1 introduces the generalized entropy of strictly proper scoring rules, which offers alternatives.

Second-order representations

The distribution-, set- or scalar-valued representations above are known as first-order. Second-order representations aim to quantify what the model itself does not know, which allows quantifying epistemic uncertainty (Section 2.2.1). Such approaches include distributions over predictive distributions, which are standard in Bayesian inference (Blundell et al., 2015; Jospin et al., 2022), and sets of distributions, also known as credal sets (Hüllermeier et al., 2024).

2.2.3 Uncertainty Quantification Methods

In this section, we present UQ methods studied in this thesis. We focus on in-distribution learning and our main interest lies in capturing *aleatoric uncertainty*. We consider the modeling approaches themselves, while loss functions are discussed separately in Section 2.3. Our considered models are instances of conditional *generative models*.

Generative models constitute a broad family of UQ methods (Murphy, 2023; Bond-Taylor et al., 2021). They aim to learn the underlying distribution of the training data, thereby enabling the generation of new samples. In particular, *conditional* generative models estimate the conditional distribution $P_{Y|X}$, making them natural tools for uncertainty quantification.

Traditional approaches rely on estimating the parameters of a fixed family of distributions or on nonparametric methods such as kernel density estimation. More recent deep generative models

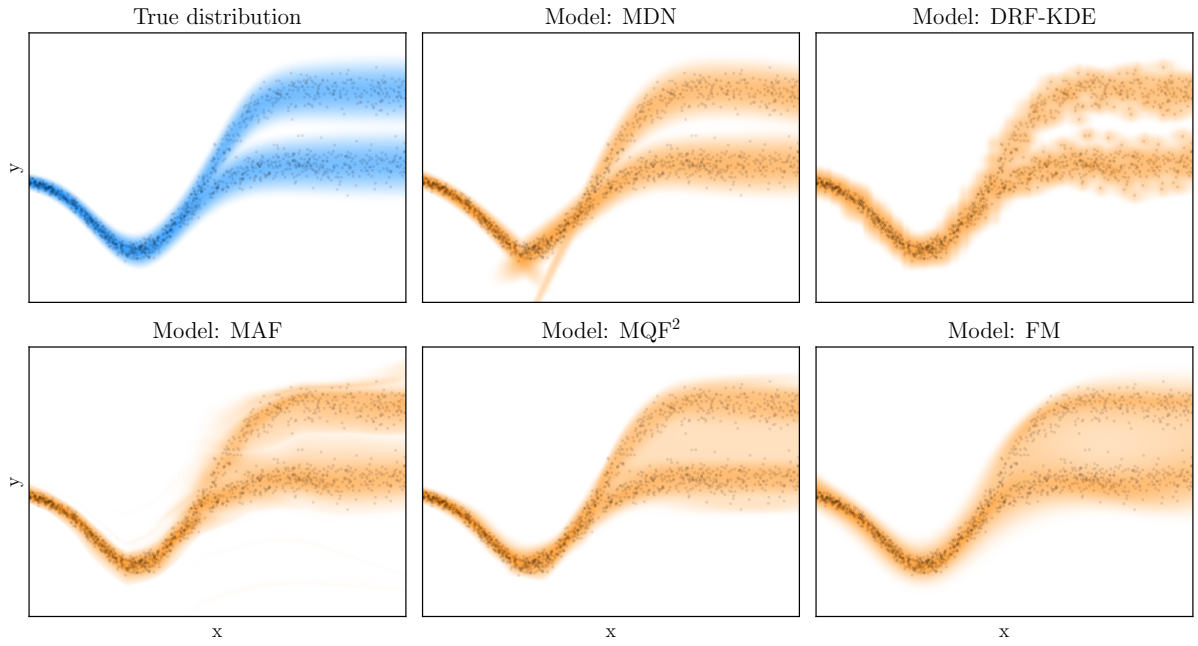


Figure 2.4: Illustrative predictions from UQ methods considered in this thesis.

scale to high-dimensional output spaces $\mathcal{Y}$, with major families including variational autoencoders (VAEs) (Kingma and Welling, 2014; D. J. Rezende et al., 2014), generative adversarial networks (GANs) (Goodfellow et al., 2014; Arjovsky et al., 2017), normalizing flows (NFs) (Dinh et al., 2017; Kobyzev et al., 2021), autoregressive models (Oord et al., 2016), energy-based models (EBMs) (LeCun et al., 2006; Y. Song and Kingma, 2021), and diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Y. Song et al., 2020).

Among these, we consider parametric methods, which are often sufficiently expressive in low-dimensional spaces, and deep generative approaches. Among deep generative approaches, normalizing flows and flow matching are of particular interest in this thesis because they yield an explicit PDF, from which strictly proper scoring rules and calibration metrics can be computed.

Figure 2.4 illustrates predictive PDFs obtained from several methods discussed in this section. As in Figure 2.2, the training data are represented as black dots. The leftmost panel shows the true PDF $f_{Y|X}$ (blue), while the remaining panels display the PDF $\hat{f}_{Y|X}$ learned by different models. Other deep generative models such as Glow (Kingma and Dhariwal, 2018) and TarFlow (Zhai et al., 2025) that only appear in one chapter are not discussed here.

Mixture Density Networks (MDNs)

A straightforward way to represent predictive uncertainty with NNs is to let the NN output the parameters of a chosen family of probability distributions. For instance, given $x \in \mathcal{X}$, suppose a NN produces two outputs $\hat{\mu}(x) \in \mathbb{R}$ and $\hat{\rho}(x) \in \mathbb{R}$. Applying the softplus function yields a positive scale parameter, $\hat{\sigma}(x) = \log(1 + e^{\hat{\rho}(x)}) > 0$. These parameters define a conditional Gaussian PDF,

$$\hat{f}_{Y|X=x}(y) = \mathcal{N}(y; \hat{\mu}(x), \hat{\sigma}(x)).$$

The specific distributional family can be chosen based on prior knowledge.

This idea originates with Bishop (1994), who further suggest using a mixture of distributions (e.g., Gaussians), leading to the concept of *Mixture Density Networks*. They argue that with a sufficiently expressive NN (a property implied by the universal approximation theorem) and enough mixture components, such models can approximate any conditional PDF $f_{Y|X}$ arbitrarily well.

For multi-output regression, we adopt an extension of this approach, also described in Muschinski et al. (2022), where a mixture of M multivariate Gaussian components is parameterized. Given $x \in \mathcal{X}$, for each mixture component $m \in [M]$, the NN outputs the logit $\hat{z}_m(x) \in \mathbb{R}$ (for the categorical distribution over the mixture components), the mean vector $\hat{\mu}_m(x) \in \mathbb{R}^d$ (component location), and the lower triangular Cholesky factor $\hat{L}_m(x) \in \mathbb{R}^{d \times d}$ (representing the scale of the covariance matrix).

The normalized mixture weights are obtained via the softmax function:

$$\hat{\pi}_m(x) = \frac{\exp(\hat{z}_m(x))}{\sum_{j=1}^{M} \exp(\hat{z}_j(x))}$$

and the covariance matrix is given by

$$\hat{\Sigma}_m(x) = \hat{L}_m(x)\hat{L}_m(x)^\top,$$

ensuring it is positive semidefinite. The conditional density of $y \in \mathcal{Y}$ given $x \in \mathcal{X}$ is therefore

$$\hat{f}_{Y|X=x}(y) = \sum_{m=1}^{M} \hat{\pi}_m(x) \cdot \mathcal{N}(y; \hat{\mu}_m(x), \hat{\Sigma}_m(x)).$$

Distributional Random Forests (DRFs)

DRFs (Cevid et al., 2022) are built upon the random forest algorithm. For any given test point $x \in \mathcal{X}$, they estimate the full conditional distribution $P_{Y|X}$ by learning weights over instances in the training dataset $\mathcal{D}_{\text{train}} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^{|\mathcal{D}_{\text{train}}|}$. A forest of trees $\{T_k\}_{k=1}^{K}$ is grown with splits that maximize a discrepancy of the outputs across candidate child nodes; the resulting leaf containing x in tree k is denoted $L_k(x)$. This induces nonnegative weights

$$\hat{w}_x(X^{(i)}) = \frac{1}{K} \sum_{k=1}^{K} \frac{\mathbb{1}\left(X^{(i)} \in L_k(x)\right)}{|L_k(x)|}, \tag{2.10}$$

that sum to one: $\sum_{i=1}^{|\mathcal{D}_{\text{train}}|} \hat{w}_x(X^{(i)}) = 1$. The weight is large if $X^{(i)}$ often falls in the same leaf as x and those leaves are small. Then, one can estimate the conditional distribution as

$$\hat{P}_{Y|X=x} = \sum_{i=1}^{|\mathcal{D}_{\text{train}}|} \hat{w}_x(X^{(i)}) \delta_{Y^{(i)}}, \tag{2.11}$$

where $\delta_{Y^{(i)}}$ is the point mass at $Y^{(i)}$.

At each parent node N_P, for disjoint candidate splits $N_L \cup N_R = N_P$, DRF selects the split that maximizes a two-sample discrepancy between the empirical distributions of Y in N_L and N_R. The default is the maximum mean discrepancy (MMD) with a Gaussian kernel, so that samples within each leaf share a similar output distribution, while child nodes differ strongly from each other. This output-based criterion yields leaves with similar output distribution and thus induces neighborhoods tailored to distributional similarity. For efficiency, the MMD is computed via a fast randomized approximation, allowing all thresholds in a node to be scanned quickly.

Given $x \in \mathcal{X}$, various conditional quantities can be approximated based on the weights $\hat{w}_x(X^{(i)})$. To obtain a conditional predictive PDF, a natural approach is to produce a Gaussian mixture with each component centered on a training point $Y^{(i)}$ and weighted by $\hat{w}_x(X^{(i)})$:

$$\hat{f}_{Y|X=x}(y) = \sum_{i=1}^{|\mathcal{D}_{\text{train}}|} \hat{w}_x(X^{(i)}) \cdot \mathcal{N}(y; Y^{(i)}, \sigma^2 I_d),$$

where σ is a hyperparameter. Besides σ, hyperparameters include the minimum node size, the number of trees, and the splitting criterion.

Normalizing Flows (NFs)

NFs (Papamakarios et al., 2021) provide a powerful and flexible method for modeling complex distributions over continuous random variables. A conditional NF is defined as a parameterized bijection $\hat{T} : \mathcal{Z} \times \mathcal{X} \to \mathcal{Y}$ between a latent space $\mathcal{Z} \subseteq \mathbb{R}^d$ and an output space $\mathcal{Y} \subseteq \mathbb{R}^d$ given an input $x \in \mathcal{X}$. For any $y \in \mathcal{Y}$ and $x \in \mathcal{X}$, the transformation satisfies

$$\hat{T}(\hat{T}^{-1}(y; x); x) = y.$$

Given $x \in \mathcal{X}$, a latent variable $Z \sim \mathcal{N}(0, I_d)$ is mapped to $\hat{Y} = \hat{T}(Z; x)$. The conditional PDF is obtained via the change-of-variables formula:

$$\hat{f}_{Y|X=x}(y) = f_Z(\hat{T}^{-1}(y; x)) \cdot \left|\det(\nabla_y \hat{T}^{-1}(y; x))\right|,$$

where f_Z denotes the density of Z and $\nabla_y \hat{T}^{-1}(y; x) = \left(\frac{\partial \hat{T}^{-1}(y;x)}{\partial y_1}, \ldots, \frac{\partial \hat{T}^{-1}(y;x)}{\partial y_d}\right) \in \mathbb{R}^{d \times d}$ is the Jacobian of $\hat{T}^{-1}(\cdot; x)$. In practice, the likelihood is typically computed in log-scale for numerical stability.

Composition of normalizing flows. In practice, $\hat{T}$ is built as a composition of K simple invertible mappings, each with a tractable Jacobian determinant. Let $\hat{T}_k : \mathbb{R}^d \times \mathcal{X} \to \mathbb{R}^d$ be the k-th transformation. Define the sequence

$$z_0 = y, \qquad z_k = \hat{T}_k^{-1}(z_{k-1}; x), \quad k = 1, \ldots, K.$$

Thus $\hat{T}^{-1} = \hat{T}_K^{-1} \circ \cdots \circ \hat{T}_1^{-1}$ and equivalently $y = \hat{T}(z_K; x)$. Applying the change-of-variables formula repeatedly yields

$$\hat{f}_{Y|X=x}(y) = f_Z(z_K) \cdot \prod_{k=1}^{K} \left|\det(\nabla_{z_{k-1}} \hat{T}_k^{-1}(z_{k-1}; x))\right|.$$

Coupling flows (Dinh et al., 2017) implement $\hat{T}_k$ by splitting the index set into two disjoint subsets $A \cup B = [d]$. We denote by z_A and z_B the input z indexed by A and B, respectively. The model uses two NNs

$$\hat{\mu} : \mathbb{R}^{|A|} \times \mathcal{X} \to \mathbb{R}^{|B|}, \qquad \hat{\rho} : \mathbb{R}^{|A|} \times \mathcal{X} \to \mathbb{R}^{|B|}.$$

and $\hat{\sigma}(z_A; x) = \exp(\hat{\rho}(z_A; x))$ ensures positive outputs. In an *affine coupling* layer, the forward transformation from an input z to an output y is defined as

$$y_A = z_A, \qquad y_B = z_B \odot \hat{\sigma}(z_A; x) + \hat{\mu}(z_A; x),$$

where $\odot$ denotes elementwise multiplication. The inverse transformation from y to z is

$$z_A = y_A, \qquad z_B = (y_B - \hat{\mu}(y_A; x)) \oslash \hat{\sigma}(y_A; x),$$

where $\oslash$ denotes elementwise division. Because the Jacobian of the inverse transformation is block-triangular, its determinant simplifies to

$$\left|\det(\nabla_y \hat{T}_k^{-1}(y; x))\right| = \prod_{i \in B} \frac{1}{\hat{\sigma}_i(y_A; x)}.$$

By alternating the partition and stacking many coupling layers, NFs can model highly flexible conditional PDFs while maintaining computational efficiency.

Masked autoregressive flows (MAFs) (Papamakarios, Pavlakou, et al., 2017) implement $\hat{T}_k^{-1}$ as an autoregressive affine transformation. For each coordinate $i \in [d]$, the model uses an autoregressive NN (e.g., MADE, (Germain et al., 2015)) to parameterize

$$\hat{\mu}_i : \mathbb{R}^{i-1} \times \mathcal{X} \to \mathbb{R}, \qquad \hat{\rho}_i : \mathbb{R}^{i-1} \times \mathcal{X} \to \mathbb{R},$$

and $\hat{\sigma}_i(y_{<i}; x) = \exp(\hat{\rho}_i(y_{<i}; x))$ ensures positive outputs. The masking mechanism ensures that $\hat{\mu}_i(y_{<i}; x)$ and $\hat{\sigma}_i(y_{<i}; x)$ depend only on the preceding coordinates $y_{<i} = (y_1, \ldots, y_{i-1})$ and the conditioning variable x.

The inverse transformation for each coordinate is then

$$z_i = \frac{y_i - \hat{\mu}_i(y_{<i}; x)}{\hat{\sigma}_i(y_{<i}; x)}, \qquad i = 1, \ldots, d.$$

The forward transformation $\hat{T}$, from z to y, is obtained by inverting the expression above:

$$y_i = z_i \cdot \hat{\sigma}_i(y_{<i}; x) + \hat{\mu}_i(y_{<i}; x), \qquad i = 1, \ldots, d.$$

This computation is sequential, as calculating y_i requires the values of $y_1, \ldots, y_{i-1}$.

Since this mapping is triangular in y, the Jacobian of $\hat{T}_k^{-1}$ is also triangular, which makes the determinant computation efficient:

$$\left|\det(\nabla_y \hat{T}_k^{-1}(y; x))\right| = \prod_{i=1}^{d} \frac{1}{\hat{\sigma}_i(y_{<i}; x)}.$$

Beyond affine transformations, more flexible transformations using splines have been developed (Durkan et al., 2019; Dolatabadi et al., 2020). A detailed overview of NF architectures can be found in Papamakarios et al. (2021).

Multivariate quantile function forecasters (MQF^2) (Kan et al., 2022) are a generalization of QFs to multivariate outputs. Given $x \in \mathcal{X}$, MQF^2 satisfies the property of cyclical monotonicity, defined as

$$(\hat{T}(z;x) - \hat{T}(z';x))^\top (z - z') \geqslant 0 \tag{2.12}$$

for any $z, z' \in \mathcal{Z}$. When $d = 1$, this reduces to the standard property that the quantile function is increasing, i.e., $\hat{T}(z;x) \leqslant \hat{T}(z';x)$ if $z < z'$. To guarantee the property of cyclical monotonicity, Kan et al. (2022) define the quantile function as a *convex potential flow* (C.-W. Huang et al., 2021).

A convex potential flow parameterizes the bijective transformation $\hat{T}$ via a strongly convex *potential* whose gradient yields the inverse map. Given $x \in \mathcal{X}$, let the model define a scalar potential $\hat{V} : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}$ where $\hat{V}(\cdot; x)$ is strongly convex for every fixed x. The associated transformation is the gradient

$$\hat{T}^{-1}(y;x) = \nabla_y \hat{V}(y;x) \in \mathbb{R}^d.$$

To ensure convexity in y, $\hat{V}(\cdot; x)$ is parameterized by an *input-convex neural network* (ICNN, Amos et al., 2017). A small quadratic term $\frac{\alpha}{2}\|y\|^2$ enforces strong convexity of $\hat{V}(\cdot; x)$.

Since $\hat{V}(\cdot; x)$ is strongly convex, $\hat{T}^{-1}(\cdot; x)$ is a bijection $\mathcal{Y} \to \mathcal{Z}$ with inverse $\hat{T}(\cdot; x)$. Generating new conditional samples requires inverting $\hat{T}^{-1}(\cdot; x)$. Given $z \in \mathcal{Z}$, one recovers $\hat{T}(z;x)$ as the unique minimizer of the convex objective

$$\hat{T}(z;x) \in \arg\min_{y \in \mathcal{Y}} \left\{ \hat{V}(y;x) - z^\top y \right\}. \tag{2.13}$$

Indeed, since $\hat{V}(\cdot; x)$ is differentiable and strongly convex, the minimum is attained when

$$\nabla_y \left(\hat{V}(y;x) - z^\top y \right) = 0 \iff \nabla_y \hat{V}(y;x) = z \iff \hat{T}^{-1}(y;x) = z \iff y = \hat{T}(z;x). \tag{2.14}$$

In practice, (2.13) can be solved efficiently with gradient-based convex optimization (e.g., L-BFGS (Liu and Nocedal, 1989)), and strong convexity ensures convergence to a unique solution.

The Hessian $\nabla_y^2 \hat{V}(y;x)$ is positive definite, with

$$\left| \det(\nabla_y \hat{T}^{-1}(y;x)) \right| = \det(\nabla_y^2 \hat{V}(y;x)).$$

One approach to compute this determinant is by explicitly forming the Hessian $H = \nabla_y^2 \hat{V}(y;x)$. Concretely, $\hat{T}^{-1}(y;x) = \nabla_y \hat{V}(y;x)$ is first evaluated with a single forward pass and backpropagation, and then the Hessian is formed as

$$\left(\frac{\partial \hat{T}^{-1}(y;x)}{\partial y_1}, \ldots, \frac{\partial \hat{T}^{-1}(y;x)}{\partial y_d} \right) \in \mathbb{R}^{d \times d},$$

requiring d additional backpropagations. Finally, the determinant can be computed explicitly in $O(d^3)$. Hence, the overall computational complexity is $O\left(d\,C_{\text{backprop}} + d^3\right)$ where C_{backprop} denotes the cost of a single backpropagation. This brute-force approach is practical for small d but becomes prohibitive as d grows; more efficient strategies for large d are discussed in C.-W. Huang et al. (2021).

This approach is closely related to the *optimal transport* theory. With a sufficiently expressive ICNN, the learned map $\hat{T}^{-1}(\cdot; x)$ can approximate the optimal transport map that pushes the conditional distribution $P_{Y|X=x}$ to the latent distribution P_Z while minimizing the average squared Euclidean distance $\int_{\mathcal{Y}} \|y - \hat{T}^{-1}(y; x)\|_2^2 \, dP_{Y|X=x}(y)$. Brenier's theorem states that for distributions on $\mathbb{R}^d$, such an optimal map is unique and is given by the gradient of a convex function. The potential $\hat{V}(\cdot; x)$ learned by the model serves as an approximation of this convex function.

Flow Matching (FM)

Flow matching (FM, Lipman et al., 2022) is a recent generative modeling paradigm which has rapidly been gaining popularity. The key motivation behind FM is to combine the strengths of normalizing flows (NFs) and diffusion models while alleviating their main limitations. NFs enable exact likelihood estimation and efficient sampling but often suffer from limited expressiveness due to architectural constraints. Diffusion models, on the other hand, offer remarkable expressiveness and stability but typically require slow iterative sampling and do not provide tractable likelihoods. FM addresses these issues by framing generative modeling as the learning of a continuous-time flow that transports noise to data, enabling both efficient training and fast sampling while retaining theoretical connections to likelihood-based methods.

Given $x \in \mathcal{X}$, we model the conditional predictive PDF $\hat{f}_{Y|X=x}$ using a transformation defined by the ordinary differential equation (ODE) $\frac{d\tilde{y}}{dt} = \hat{v}\left(t, \tilde{y}, x\right)$, with a NN-parameterized vector field $\hat{v} : [0,1] \times \mathcal{Y} \times \mathcal{X} \to \mathcal{Y}$. Training uses the linear interpolant between latent $z \sim \mathcal{N}(0, I)$ and data $y \sim f_{Y|X=x}$,

$$\tilde{y}(t) = (1-t)\, z + t y,$$

whose target velocity is constant w.r.t. t:

$$\frac{d}{dt}\tilde{y}(t) = y - z.$$

The FM objective is

$$\min_{\hat{v}} \; \mathbb{E}\left\|\hat{v}\left(t, (1-t)\, Z + tY, X\right) - (Y - Z)\right\|^2 \tag{2.15}$$

where the expectation is over $t \sim \mathcal{U}(0,1)$, $(X, Y) \sim P_{X,Y}$, and $Z \sim \mathcal{N}(0, I)$. After training, forward numerical integration generates samples from $\hat{f}_{Y|X=x}$ (Chen et al., 2018):

$$\hat{T}\left(z; x\right) = \tilde{y}\left(1\right) = z + \int_0^1 \hat{v}\left(t, \tilde{y}\left(t\right), x\right) dt, \qquad \tilde{y}\left(0\right) = z. \tag{2.16}$$

Reverse-time integration encodes y into its latent z:

$$\hat{T}^{-1}\left(y; x\right) = \tilde{y}\left(0\right) = y + \int_1^0 \hat{v}\left(t, \tilde{y}\left(t\right), x\right) dt, \qquad \tilde{y}\left(1\right) = y. \tag{2.17}$$

For $y \in \mathcal{Y}$, set $z = \hat{T}^{-1}(y; x)$. The log-likelihood follows from the instantaneous change-of-variables formula along the unique ODE path $\tilde{y}(t)$ with $\tilde{y}(0) = z$ and $\tilde{y}(1) = y$:

$$\log \hat{f}_{Y|X=x}(y) = \log f_Z(z) - \int_0^1 \operatorname{Tr}\left(\nabla_{\tilde{y}} \hat{v}(t, \tilde{y}(t), x)\right) dt. \tag{2.18}$$

The trace of the Jacobian can be computed using d backpropagations, which can be prohibitive if d is large. It can also be efficiently approximated using Hutchinson's estimator:

$$\operatorname{Tr}\left(\nabla_{\tilde{y}} \hat{v}(t, \tilde{y}(t), x)\right) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)}\left[\epsilon^\top \left(\nabla_{\tilde{y}} \hat{v}(t, \tilde{y}(t), x)\right) \epsilon\right] \approx \frac{1}{K} \sum_{k=1}^{K} \epsilon_k^\top \left(\nabla_{\tilde{y}} \hat{v}(t, \tilde{y}(t), x)\right) \epsilon_k, \tag{2.19}$$

with independent $\epsilon_k \sim \mathcal{N}(0, I)$, which enables practical likelihood computation.

2.3. Scoring Rules

Scoring rules (Gneiting and Raftery, 2007; Waghmare and Ziegel, 2025) provide a means to evaluate predictive distributions. Proper scoring rules were first developed in meteorology as tools to evaluate probabilistic forecasts. The quadratic score introduced by Brier (1950) and McCarthy's (1956) formalization of propriety laid the foundation for ensuring that forecasters were incentivized to state their true beliefs. General characterizations were later provided by Hendrickson and Buehler (1971) and Gneiting and Raftery (2007) establishing proper scoring rules as a central concept in statistical theory. Since then, the framework has been expanded beyond meteorology into fields such as economics, where it is used for belief elicitation (Schotter and Trevino, 2014) and machine learning, where it defines principled loss functions (Kan et al., 2022; Si et al., 2023; Shao et al., 2024; De Bortoli et al., 2025).

Today, proper scoring rules serve two main purposes: estimation and evaluation. On the estimation side, minimizing proper scoring rules generalizes maximum likelihood estimation. On the evaluation side, proper scoring rules provide a scalar measure of *predictive accuracy* by evaluating predictive distributions against observed outcomes. They are widely applied in domains ranging from weather and energy forecasting to finance and epidemiology, where they provide principled means to compare predictive performance (Gneiting and Katzfuss, 2014).

For conciseness, in this section, unless mentioned otherwise, we restrict ourselves to unconditional distributions over the outcome Y. Furthermore, we omit the index Y, i.e., $P = P_Y$, $\hat{P} = \hat{P}_Y$, $F = F_Y$, etc. Let $\bar{\mathbb{R}}$ denote the extended real line $\mathbb{R} \cup \{-\infty, +\infty\}$. Given a predictive distribution $\hat{P} \in \mathcal{P}(\mathcal{Y})$ and a realization $y \sim P$, a scoring rule S returns a real number $S(\hat{P}, y) \in \bar{\mathbb{R}}$. The expected value of a scoring rule w.r.t. the true distribution is denoted

$$S(\hat{P}, P) = \mathbb{E}\Big[S(\hat{P}, Y)\Big] = \int_{\mathcal{Y}} S(\hat{P}, y) P(dy). \tag{2.20}$$

Note that we use S to denote both the scoring rule and its expectation; the arguments of the function disambiguate between the two. Following the ML literature, we treat scoring rules as loss functions, implying that their expected value (also known as the risk function) should be minimized.

Propriety. A commonly desired property of a scoring rule is propriety (and, more strongly, *strict* propriety). A scoring rule is proper if

$$S(P, P) \leqslant S(\hat{P}, P), \tag{2.21}$$

for any $\hat{P} \in \mathcal{P}(\mathcal{Y})$. It is *strictly proper* if, additionally, $S(P, P) = S(\hat{P}, P)$ implies $\hat{P} = P$.

Example 2. In ML, the most commonly used strictly proper scoring rule is the NLL, also known as logarithmic score, given by $S(\hat{P}, y) = -\log \hat{f}(y)$.

In practice, for a dataset $\mathcal{D} = \{Y^{(1)}, \ldots, Y^{(|\mathcal{D}|)}\}$ the expected value is usually estimated by its empirical average:

$$\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} S(\hat{P}, Y^{(i)}), \tag{2.22}$$

which is an unbiased and consistent estimator of $S(\hat{P}, P)$. Strictly proper scoring rules allow comparing predictive distributions $\hat{P}_1, \ldots, \hat{P}_M$, and the ones with the lowest average score are considered the most accurate. When used as loss functions, they encourage the model's predictions to converge to the true distribution. Estimation using (2.22) is known as optimum score estimation, with maximum likelihood estimation being a special case when S is the NLL.

Remark. If S is a (strictly) proper scoring rule, $c > 0$ is a constant, and $\psi : \mathcal{Y} \to \bar{\mathbb{R}}$ is any function, then

$$S'(\hat{P}, y) = cS(\hat{P}, y) + \psi(y) \tag{2.23}$$

is minimized by the same $\hat{P}$ and is thus also (strictly) proper.

2.3.1 Divergence and generalized entropy

Any scoring rule induces a generalized entropy

$$H(P) = S(P, P), \tag{2.24}$$

which is (strictly) concave on $\mathcal{P}(\mathcal{Y})$ when S is (strictly) proper (Gneiting and Raftery, 2007). It also induces a divergence

$$D(\hat{P}, P) = S(\hat{P}, P) - H(P), \tag{2.25}$$

satisfying $D(\hat{P}, P) \geqslant 0$ if S is proper and uniquely minimized at $\hat{P} = P$ if S is strictly proper.

By definition,

$$S(\hat{P}, P) = D(\hat{P}, P) + H(P). \tag{2.26}$$

When S is strictly proper, the first term is zero if and only if $\hat{P} = P$. Since this term reflects model suboptimality, it relates to epistemic uncertainty and Kull and Flach (2015) call it epistemic loss. The second term only depends on the true distribution P and thus on the inherent uncertainty in the output. In contrast to the epistemic loss, it is related to aleatoric uncertainty.

In practice, only the predictive distribution $\hat{P}$ is known. It serves as the basis for estimating the properties of the true distribution P, including its inherent aleatoric uncertainty. Decreasing $D(\hat{P}, P)$ can then be interpreted both as decreasing epistemic uncertainty, and improving the estimation of aleatoric uncertainty.

2.3.2 Well-Known Strictly Proper Scoring Rules

Many scoring rules have been designed for both classification and regression problems, that is, for discrete and continuous distributions. In this thesis, we focus on scoring rules for continuous distributions. Well-known strictly proper scoring rules (Gneiting and Raftery, 2007) for continuous distributions are presented in Table 2.1.

Uncertainty representation requirements. Scoring rules can depend on various representations of the predictive distribution $\hat{P}$. The NLL (Good, 1952), quadratic score (QS, Brier, 1950) and pseudospherical score (PSS, Good et al., 1971) require a predictive PDF $\hat{f}$. The continuous ranked probability score (CRPS, Matheson and Winkler, 1976) has two equivalent definitions in terms of the CDF $\hat{F}$ or QF $\hat{Q}$. The QF version of the CRPS uses the check function $\rho_\alpha(u) = u \cdot (\alpha - \mathbb{1}(u < 0))$. The kernel score (KS, Gneiting and Raftery, 2007) requires independent samples $\hat{Y}, \tilde{Y} \sim \hat{P}$ from the predictive distribution. The Hyvärinen score (HS, Hyvärinen, 2005) depends on the Laplace operator $\Delta g(y) = \sum_{i=1}^{d} \frac{\partial^2 g(y)}{\partial y_i^2}$. The CRPS is only compatible with univariate outcomes ($d = 1$) while the others are compatible with multivariate outcomes ($d \geqslant 1$).

Conditions for strict propriety. Under technical measure-theoretic constraints detailed in Gneiting and Raftery (2007), the NLL, QS and CRPS are strictly proper. The HS is also strictly proper (Hyvärinen (2005)). The PSS is strictly proper when $\gamma > 1$. When $\gamma = 2$, the PSS is known as the spherical score (SS). When $\gamma \to 1$, the PSS tends to the NLL. The KS is strictly proper when $k : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is positive definite and universal (Gretton et al., 2012). When $k(y, y') = -\|y - y'\|^\beta$ with $\beta \in (0, 2)$, the KS is known as the *energy score* (ES). The ES is equal to the CRPS when $\beta = 1$ and $d = 1$.

Divergence and generalized entropy. Table 2.1 also details the divergence and generalized entropy (introduced in Section 2.3.1) of each scoring rule. For the NLL, the divergence is the KL divergence:

$$D(\hat{P}, P) = D_{\mathrm{KL}}\Big(P \,\Big\|\, \hat{P}\Big) = \mathbb{E}\Big[\log f(Y) - \log \hat{f}(Y)\Big] \tag{2.27}$$

and the entropy is the differential entropy.

For the HS, the divergence is the Fisher divergence:

$$D(\hat{P}, P) = D_{\mathrm{F}}\Big(\hat{P} \,\Big\|\, P\Big) = \tfrac{1}{2}\mathbb{E}\Big[\|\nabla \log f(Y) - \nabla \log \hat{f}(Y)\|^2\Big]. \tag{2.28}$$

For the KS, the divergence is the squared Maximum Mean Discrepancy (MMD) with kernel k:

$$\mathrm{MMD}_k^2(\hat{P}, P) = \mathbb{E}_{\hat{Y}, \tilde{Y} \sim \hat{P}}\Big[k(\hat{Y}, \tilde{Y})\Big] + \mathbb{E}_{Y', Y'' \sim P}\big[k(Y', Y'')\big] - 2\mathbb{E}_{\hat{Y} \sim \hat{P}, Y' \sim P}\Big[k(\hat{Y}, Y')\Big]. \tag{2.29}$$

Given a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$, another common equivalent definition for the MMD is the following:

$$\mathrm{MMD}_k(\hat{P}, P) = \sup_{\|g\|_{\mathcal{H}_k} \leqslant 1} \Big\{\mathbb{E}_{\hat{Y} \sim \hat{P}}[g(\hat{Y})] - \mathbb{E}_{Y' \sim P}[g(Y')]\Big\}. \tag{2.30}$$

Table 2.1: Well-known strictly proper scoring rules with their associated divergence and generalized entropy.

	Scoring rule $S(\hat{P}, y)$	**Divergence** $D(\hat{P}, P)$	**Gen. entropy** $H(P)$
NLL	$-\log \hat{f}(y)$	$D_{\mathrm{KL}}\Big(P \,\Big\|\, \hat{P}\Big)$	$-\mathbb{E}[\log f(Y)]$
QS	$\|\hat{f}\|_2^2 - 2\hat{f}(y)$	$\int_{\mathcal{Y}} (\hat{f}(y) - f(y))^2 dy$	$-\|f\|_2^2$
PSS	$-\hat{f}(y)^{\gamma-1} / \|\hat{f}\|_\gamma^{\gamma-1}$	$\|f\|_\gamma - \int_{\mathcal{Y}} f(y)\hat{f}(y)^{\gamma-1} dy / \|\hat{f}\|_\gamma^{\gamma-1}$	$-\|f\|_\gamma$
CRPS	$\int_{\mathcal{Y}} (\hat{F}(z) - \mathbb{1}(z \geqslant y))^2 dz$	$\int_{\mathcal{Y}} (\hat{F}(y) - F(y))^2 dy$	$\int_{\mathcal{Y}} F(y)(1 - F(y)) dy$
CRPS	$2\int_0^1 \rho_\alpha(y - \hat{Q}(\alpha)) d\alpha$	$2\int_0^1 \mathbb{E}[\rho_\alpha(Y - \hat{Q}(\alpha)) - \rho_\alpha(Y - Q(\alpha))] d\alpha$	$2\int_0^1 \left(\alpha - \frac{1}{2}\right) Q(\alpha) d\alpha$
HS	$\Delta \log \hat{f}(y) + \frac{1}{2}\|\nabla \log \hat{f}(y)\|^2$	$D_{\mathrm{F}}\Big(\hat{P} \,\Big\|\, P\Big)$	$-\frac{1}{2}\mathbb{E}[\|\nabla \log f(Y)\|^2]$
KS	$\mathbb{E}_{\hat{Y}, \tilde{Y} \sim \hat{P}}[k(\hat{Y}, \tilde{Y}) - 2k(\hat{Y}, y)]$	$\mathrm{MMD}_k^2(\hat{P}, P)$	$-\mathbb{E}_{Y', Y'' \sim P}[k(Y', Y'')]$

Computational considerations. The computational cost for a strictly proper scoring rule can vary widely. When $\hat{f}$ is available, the NLL is practical as it only requires one evaluation of the predictive PDF. The QS, CRPS, and PSS all require evaluating an integral, which can require numerical approximations. The HS requires evaluating the Laplacian of $\log \hat{f}(y)$, which in the general case requires running backpropagation d times, which does not scale well. Denoising score matching (Vincent, 2011) provides an efficient approximation. The KS typically requires approximating the expectations $\mathbb{E}\Big[k(\hat{Y}, \tilde{Y})\Big]$ and $\mathbb{E}\Big[k(\hat{Y}, y)\Big]$ based on samples. However, closed-form expressions often exist, avoiding potentially costly and imprecise numerical approximations.

Illustrative example for Gaussian mixtures. To accurately compare the strictly proper scoring rules presented in Table 2.1, we derived closed-form expressions for the QS, SS (i.e., PSS with $\gamma = 2$), HS, and KS with an RBF kernel in the context of multivariate Gaussian mixture predictions. These expressions, which we did not find in the literature, are detailed in Section B.1 along with their computational complexities. Overall, the HS and NLL are the fastest to evaluate. As an illustrative example, Figure 2.5 shows the predictions of a MDN with 3 mixture components trained with each of these scoring rules on a synthetic dataset ($d = p = 1$). In each case, we keep the same architecture and only vary the loss function. The training dataset is shown as black dots and the conditional predictive PDF is shown as orange shades. In this specific example, the NLL, QS, and SS are able to capture the true distribution quite precisely, except that the PDF is too high in certain areas. The PDF of the HS is slightly worse because it does not capture the upper mode well. KS-RBF provides the worst PDF because it does not capture the true uncertainty correctly for both low- and high-uncertainty regions. This example illustrates that, while strictly proper scoring rules are theoretically guaranteed to be minimized by the true distribution, the predicted and true distribution empirically differ with varying degrees.

Practical recommendations. There is no universally best strictly proper scoring rule; the choice depends on the task, the available representation of $\hat{P}$, and computational constraints. While minimizing strictly proper scoring rules under suitable conditions converges to the true distribution, in practice a scoring rule S emphasizes different statistical properties depending on

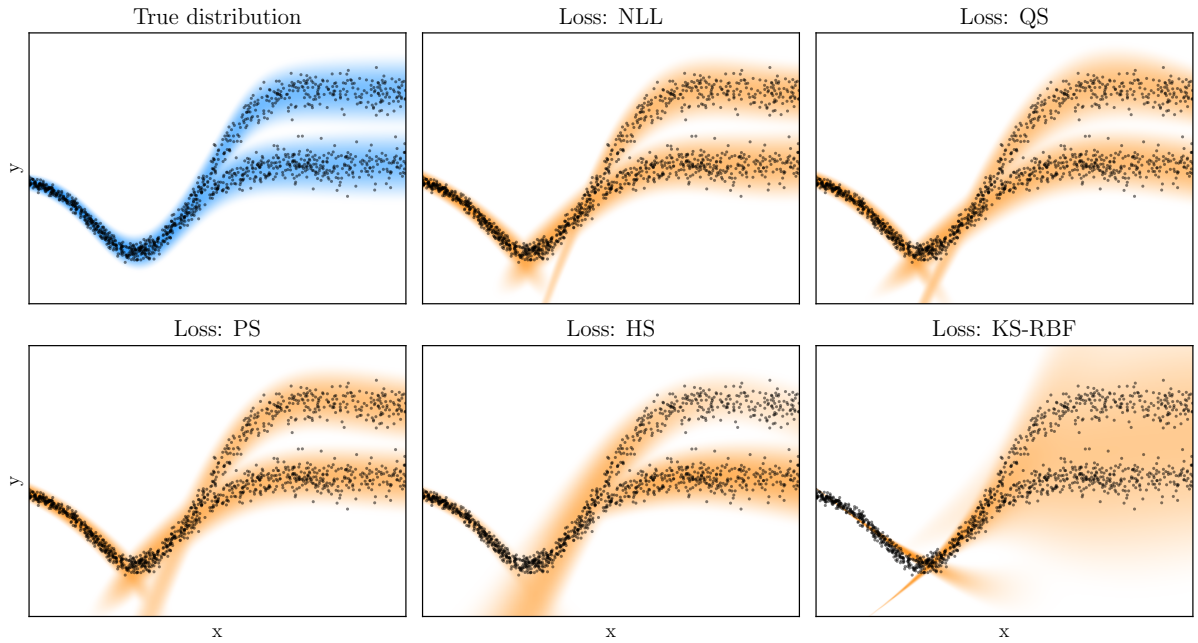


Figure 2.5: Example of predictive PDFs produced by a MDN trained using various strictly proper scoring rules.

its associated divergence D.

The most immediate constraint is the form in which the predictive distribution $\hat{P}$ is available:

- If $\hat{f}$ can be evaluated efficiently, the NLL is a natural and computationally cheap choice. The HS is another possibility, and is uniquely suited for energy-based models, where the PDF is known only up to a constant (Gustafsson et al., 2020).
- For univariate predictions, if $\hat{F}$ or $\hat{Q}$ is available, the CRPS is a common choice. It does not require a PDF and is well-defined for discrete, continuous or mixed distributions.
- If only samples $\{\hat{Y}_i\}_{i=1}^{K} \sim \hat{P}$ are available, the KS is a possibility.

H. Du (2021) argues that the NLL should be prioritized due to its unique properties. The NLL is the only strictly proper scoring rule that is local, meaning its value only depends on $\hat{f}(y)$ (or $\hat{p}(y)$), the density (or mass) assigned to the actual outcome. As a consequence, it has a direct interpretation related to information theory, and it can be shown that it ranks models consistently under any bijective transformation of Y.

H. Du (2021) also recommend that non-local scoring rules should be used with more caution than the NLL. In particular, Pinson and Tastu (2013) find that the energy score is highly sensitive to errors in the mean but can have poor discrimination ability for errors in the scale or multivariate dependence structure.

Buchweitz et al. (2025) show that the NLL, CRPS, QS, SS and ES penalize underestimating or overestimating a given parameter of the target distribution (such as the scale) in different ways, giving insights on what they incentivize.

Overall, since each scoring rule provides a different, incomplete summary of the predictive distribution quality, relying on a single score can be misleading. A more robust evaluation strategy is to report a small set of complementary scores. This provides a multi-faceted view of the model's performance.

2.4. Calibration

While training with strictly proper scoring rules encourages accurate predictions, it does not guarantee that the resulting predictions are reliable or calibrated, i.e., statistically aligned with the true distribution of the observations (Gneiting et al., 2007). This issue is particularly relevant under limited data or model misspecification, and it has gained renewed attention with the observation that modern neural network classifiers are often systematically miscalibrated and overconfident (Guo et al., 2017).

Calibration is inherently multi-faceted; no single universal definition exists. Instead, several notions have been proposed, each with distinct theoretical and practical trade-offs. A minimal requirement is that an *ideal* model (the oracle that predicts the true data-generating distribution) satisfies the notion. In this section, we focus on calibration notions and calibration methods for regression, with emphasis on probabilistic calibration and its multivariate generalization, *pre-rank* calibration.

2.4.1 Unconditional Calibration Notions

Following the framework of Marx et al. (2023), we define *unconditional* calibration notions as properties that can be expressed as the equality in distribution between two random variables: a *predictive variable* that depends on the model's prediction, and a *target variable* that does not. Table 2.2 lists several notions considered below.

Table 2.2: Unconditional calibration notions (with $U \sim \mathcal{U}(0,1)$ standard uniform).

Calibration notion	**Dimension**	**Predictive variable**	**Target variable**
Probabilistic calibration	$d = 1$	$\hat{F}_{Y\|X}(Y)$	U
Pre-rank calibration	$d \geqslant 1$	$F_{\hat{G}\|X}(G)$	U
HDR-calibration	$d \geqslant 1$	$\mathrm{HPD}_{\hat{f}_{Y\|X}}(Y)$	U
Marginal calibration	$d \geqslant 1$	$\hat{Y}$	Y

Probabilistic calibration (Gneiting et al., 2007) is a fundamental calibration notion for single-output regression ($d = 1$). It builds on the probability integral transform (PIT). Recall that $F_{Y|X}$ is a CDF-valued random variable whose value depends on X. Assuming $F_{Y|X=x}$ is continuous for any $x \in \mathcal{X}$, the PIT states

$$F_{Y|X}(Y) \sim \mathcal{U}(0,1). \tag{2.31}$$

Proof. Consider the random variable $\tilde{U} = F_{Y|X}(Y)$ and $u \in [0,1]$. We can show that $\tilde{U}$ is

uniformly distributed on $[0,1]$:

$$\begin{aligned} F_{\tilde{U}}(u) &= \mathbb{P}\left(F_{Y|X}(Y) \leqslant u\right) \\ &= \mathbb{E}\left[\mathbb{P}\left(F_{Y|X}(Y) \leqslant u \mid X\right)\right] && \text{(Law of total expectation)} \\ &= \mathbb{E}\left[\mathbb{P}\left(Y \leqslant Q_{Y|X}(u) \mid X\right)\right] && \text{(Invertibility of } F_{Y|X}\text{)} \\ &= \mathbb{E}\left[F_{Y|X}(Q_{Y|X}(u))\right] \\ &= \mathbb{E}[u] \\ &= u. \end{aligned}$$

□

Definition 1 (Probabilistic calibration)**.** A model is probabilistically calibrated if its predictive CDF satisfies

$$\hat{F}_{Y|X}(Y) \sim \mathcal{U}(0,1). \tag{2.32}$$

This property is also sometimes called quantile calibration in the literature (Kuleshov et al., 2018). To better understand this fundamental property, we also consider equivalent characterizations. In terms of the predictive QF $\hat{Q}$,

$$\mathbb{P}\left(Y \leqslant \hat{Q}_{Y|X}(\alpha)\right) = \alpha \quad \forall \alpha \in [0,1], \tag{2.33}$$

i.e., all quantile levels are calibrated. Equivalently, all predictive intervals are calibrated:

$$\mathbb{P}\left(\hat{Q}_{Y|X}(\alpha_1) \leqslant Y \leqslant \hat{Q}_{Y|X}(\alpha_2)\right) = \alpha_2 - \alpha_1 \quad \forall \alpha_1, \alpha_2 \in [0,1] \text{ with } \alpha_1 \leqslant \alpha_2. \tag{2.34}$$

The following definition, closer to the definition of Gneiting et al. (2007), is equivalent to probabilistic calibration and, assuming $\mathcal{X}$ is finite, allows an exact computation in the oracle setting (i.e., p_X and $F_{Y|X}$ are known):

$$\mathbb{E}_X\Big[F_{Y|X}(\hat{Q}_{Y|X}(\alpha))\Big] = \sum_{x \in \mathcal{X}} p_X(x) F_{Y|X=x}(\hat{Q}_{Y|X=x}(\alpha)) = \alpha \quad \forall \alpha \in [0,1]. \tag{2.35}$$

By introducing a random variable $U \sim \mathcal{U}(0,1)$, probabilistic calibration can be generalized to non-continuous distributions:

$$\hat{F}_{Y|X}(Y_-) + U \cdot (\hat{F}_{Y|X}(Y) - \hat{F}_{Y|X}(Y_-)) \sim \mathcal{U}(0,1). \tag{2.36}$$

where $\hat{F}_{Y|X}(y_-) = \lim_{y' \uparrow y} \hat{F}_{Y|X}(y')$ denotes the left-hand limit. This definition handles discrete values by randomly distributing the probability mass at each discrete point over the corresponding vertical jump in the step-wise CDF, ensuring the PIT is uniformly distributed. However, since we focus on continuous predictive distributions, this more general definition is not necessary in this thesis.

Pre-rank calibration (Allen et al., 2024) generalizes probabilistic calibration to multi-output regression ($d \geqslant 1$). Consider a function $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, which we call *pre-rank*, that projects input-output pairs to a real value, and set $G = g(X,Y)$. Assuming $F_{G|X=x}$ is continuous for any $x \in \mathcal{X}$, by the PIT,

$$F_{G|X}(G) \sim \mathcal{U}(0,1). \tag{2.37}$$

Definition 2 (Pre-rank calibration)**.** Let $\hat{Y} \sim \hat{P}_{Y|X}$ and the corresponding pre-rank variable $\hat{G} = g(X, \hat{Y})$. A model is pre-rank calibrated w.r.t. the pre-rank g if

$$F_{\hat{G}|X}(G) \sim \mathcal{U}(0, 1). \tag{2.38}$$

For single-output regression with $g(x, y) = y$, pre-rank calibration reduces to probabilistic calibration because $F_{\hat{G}|X}(G) = \hat{F}_{Y|X}(Y)$. But in the general case, for a fixed $x \in \mathcal{X}$, $F_{\hat{G}|X=x}$ needs to be estimated, usually using sampling:

$$F_{\hat{G}|X=x}(G) \approx \frac{1}{K} \sum_{i=1}^{K} \mathbb{1}(\hat{G}^{(i)} \leqslant G), \tag{2.39}$$

where $\hat{Y}^{(1)}, \ldots, \hat{Y}^{(K)} \sim \hat{P}_{Y|X=x}$ and $\hat{G}^{(i)} = g(x, \hat{Y}^{(i)})$.

A notable property of pre-rank calibration is that it is invariant under strictly monotone transformations of the pre-rank.

Proposition 1 (Invariance of pre-rank calibration under monotone transformations). Consider a strictly increasing or decreasing function $t : \mathbb{R} \to \mathbb{R}$. The model is pre-rank calibrated w.r.t. g if and only if it is pre-rank calibrated w.r.t. $t \circ g$.

Proof. Since the model is pre-rank calibrated, $F_{\hat{G}|X}(G) \sim \mathcal{U}(0, 1)$. If t is strictly increasing, then $F_{t(\hat{G})|X}(t(G)) = F_{\hat{G}|X}(G) \sim \mathcal{U}(0, 1)$. If t is strictly decreasing, then $F_{t(\hat{G})|X}(t(G)) = 1 - F_{\hat{G}|X}(G) \sim \mathcal{U}(0, 1)$. In both cases the result is uniform on $[0, 1]$. Using similar arguments, the converse is true. □

In order to interpret multivariate predictions under multiple angles, Allen et al. (2024) proposed several pre-ranks, such as the mean $g(x, y) = \mu_y = \frac{1}{d} \sum_{i=1}^{d} y_i$, the variance $g(x, y) = \sigma_y^2 = \frac{1}{d} \sum_{i=1}^{d} (y_i - \mu_y)^2$ or a variogram-based pre-rank to measure the dependence among dimensions $g(x, y) = -\frac{\gamma_y(h)}{\sigma_y^2}$ with $\gamma_y(h) = \frac{1}{2(d-h)} \sum_{j=1}^{d-h} |y_j - y_{j+h}|^2$. They note that every scoring rule is also a valid pre-rank.

HDR-calibration is a notable special case of pre-rank calibration.

Definition 3 (HDR-calibration)**.** A model is HDR-calibrated (Y. Chung et al., 2024) if it is pre-rank calibrated w.r.t. the pre-rank $g(x, y) = \hat{f}_{Y|X=x}(y)$.

By Proposition 1, it equivalently corresponds to pre-rank calibration w.r.t. the log-likelihood $g(x, y) = \log \hat{f}_{Y|X=x}(y)$ or the NLL $g(x, y) = -\log \hat{f}_{Y|X=x}(y)$.

The name originates from highest density regions (HDRs, Hyndman, 1996). The HDR w.r.t. $\hat{f}_{Y|X=x}$ at level $1 - \alpha$ corresponds to the set of values with the highest density such that $\hat{Y} \sim \hat{P}_{Y|X=x}$ has a probability of at least $1 - \alpha$ to be in the set:

$$\mathrm{HDR}_{\hat{f}_{Y|X=x}}(1 - \alpha) = \{y \in \mathcal{Y} : \hat{f}_{Y|X=x}(y) \geqslant t_{1-\alpha}\}$$
$$\text{where } t_{1-\alpha} = \sup\{t : \mathbb{P}(\hat{f}_{Y|X}(\hat{Y}) \geqslant t \mid X = x) \geqslant 1 - \alpha\}.$$

Assuming $F_{\hat{G}|X=x}$ is bijective, the threshold $t_{1-\alpha}$ can be equivalently written as a function of the conditional QF of $\hat{G}$:

$$t_{1-\alpha} = \sup\{t : \mathbb{P}(\hat{G} \geqslant t \mid X = x) \geqslant 1-\alpha\} \tag{2.40}$$
$$= \sup\{t : F_{\hat{G}|X=x}(t) \leqslant \alpha\} \tag{2.41}$$
$$= Q_{\hat{G}|X=x}(\alpha), \tag{2.42}$$

where we use the definition of the upper quantile function in the last step.

An important concept related to the HDR is the highest predictive density (HPD; Box and Tiao (1992)), defined as

$$\mathrm{HPD}_{\hat{f}_{Y|X=x}}(y) = 1 - F_{\hat{G}|X=x}(\hat{f}_{Y|X=x}(y)). \tag{2.43}$$

For all $y \in \mathcal{Y}$, the HPD is involved in the following equivalence:

$$y \in \mathrm{HDR}_{\hat{f}_{Y|X=x}}(1-\alpha) \iff \hat{f}_{Y|X=x}(y) \geqslant Q_{\hat{G}|X=x}(\alpha) \tag{2.44}$$
$$\iff F_{\hat{G}|X=x}(\hat{f}_{Y|X=x}(y)) \geqslant \alpha \tag{2.45}$$
$$\iff \mathrm{HPD}_{\hat{f}_{Y|X=x}}(y) \leqslant 1-\alpha. \tag{2.46}$$

By definition, the HDR satisfies $\mathbb{P}(\hat{Y} \in \mathrm{HDR}_{\hat{f}_{Y|X=x}}(1-\alpha) \mid X = x) \geqslant 1-\alpha$ for all $x \in \mathcal{X}$. This property only depends on the predictive density $\hat{f}_{Y|X=x}$ and not on the true conditional distribution $P_{Y|X}$.

In contrast, HDR-calibration requires

$$\mathbb{P}(Y \in \mathrm{HDR}_{\hat{f}_{Y|X}}(1-\alpha)) = \mathbb{P}(\mathrm{HPD}_{\hat{f}_{Y|X}}(Y) \leqslant 1-\alpha) = 1-\alpha, \tag{2.47}$$

which depends on the true conditional distribution $P_{Y|X}$.

Definition 4 (Marginal calibration)**.** A model is marginally calibrated (Gneiting et al., 2007) if the marginal distribution of the predictive variable $\hat{Y}$ matches the marginal distribution of the true outcome variable Y, i.e.,

$$Y \overset{\text{d.}}{=} \hat{Y}, \tag{2.48}$$

with $\overset{\text{d.}}{=}$ denoting equality in distribution.

In the literature, marginal calibration has received less attention than probabilistic calibration. A notable exception is the copula-based approach proposed by N. Klein et al. (2021). This calibration notion, originally proposed for single-output regression (Gneiting et al., 2007), is directly generalizable to multi-output regression. However, in contrast to pre-rank calibration, it becomes more challenging to assess and visualize for large output dimensions d.

Gneiting and Resin (2023) provide a unified framework for assessing unconditional calibration w.r.t. a specific functional, $T : \mathcal{P}(\mathcal{Y}) \to \mathbb{R}$. A functional is a property of a distribution, such as its mean, median, a specific quantile, or a specific moment. In contrast to the calibration notions in Table 2.2 which evaluate probabilistic predictions, this concept allows evaluating point predictions targeting a specific functional of the ideal distribution.

The framework is built upon an *identification function* $V : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, which is associated with the functional T. It is assumed that $V(\cdot, y)$ is increasing and left-continuous for any $y \in \mathbb{R}$. This function is designed such that, for any distribution $\hat{P} \in \mathcal{P}(\mathcal{Y})$, $T(\hat{P}) \in \mathbb{R}$ satisfies $\mathbb{E}_{\hat{Y} \sim \hat{P}}[V(T(\hat{P}), \hat{Y})] = 0$. In the following, for a concise presentation, we assume that this value is unique.

Definition 5. A model is unconditionally T-calibrated if

$$\mathbb{E}\Big[V(T(\hat{P}_{Y|X}), Y)\Big] = 0. \tag{2.49}$$

In contrast to other unconditional calibrations considered here, the equality is between two scalars, not two random variables.

Two prominent examples illustrate this general definition:

- **Unconditional mean calibration**: If $T(\hat{P}) = \mathbb{E}_{\hat{Y} \sim \hat{P}}\Big[\hat{Y}\Big]$ is the mean functional, the corresponding identification function is $V(x, y) = x - y$. Unconditional T-calibration for the mean thus requires $\mathbb{E}\Big[T(\hat{P}_{Y|X})\Big] = \mathbb{E}_{Y \sim P_Y}[Y]$, which is the standard condition that predictions are unconditionally unbiased.
- **Unconditional quantile calibration**: If $T(\hat{P}) = \hat{Q}(\alpha)$ is the α-quantile functional, the corresponding identification function is $V(x, y) = \mathbb{1}(y \leqslant x) - \alpha$. In this case, unconditional T-calibration requires $\mathbb{P}\left(Y \leqslant T(\hat{P}_{Y|X})\right) = \alpha$. Assuming $\hat{F}_{Y|X=x}$ is continuous for any $x \in \mathcal{X}$, unconditional α-quantile calibration for all $\alpha \in [0, 1]$ is probabilistic calibration (Gneiting and Resin, 2023).

2.4.2 Reliability diagrams

Reliability diagrams are standard tools for visualizing calibration. In the context of probabilistic and pre-rank calibration, they plot the empirical CDF of the predictive variable against its theoretical uniform target. For a perfectly calibrated model, the curve aligns with the diagonal, representing the identity function.

Figure 2.6 illustrates these diagrams for five simple predictive PDFs (first panel) where neither the true nor the predictive distributions depend on X. The true distribution is a standard Gaussian ($\mathcal{N}(0, 1)$), while the predictive distributions are chosen to be overconfident, underconfident, right-biased, or left-biased. The reliability diagrams (second and third panels) reveal how these miscalibrations manifest differently for each notion.

For probabilistic calibration, over- and underconfident predictions result in characteristic S-shaped curves. Biased predictions, in contrast, shift the curve consistently above or below the diagonal. For HDR-calibration, the patterns differ: overconfidence leads to a curve above the diagonal, while underconfidence leads to a curve below it. Notably, both left- and right-biased predictions result in identical curves below the diagonal because the pre-rank depends only on the value of the predictive density, not the location of the observation.

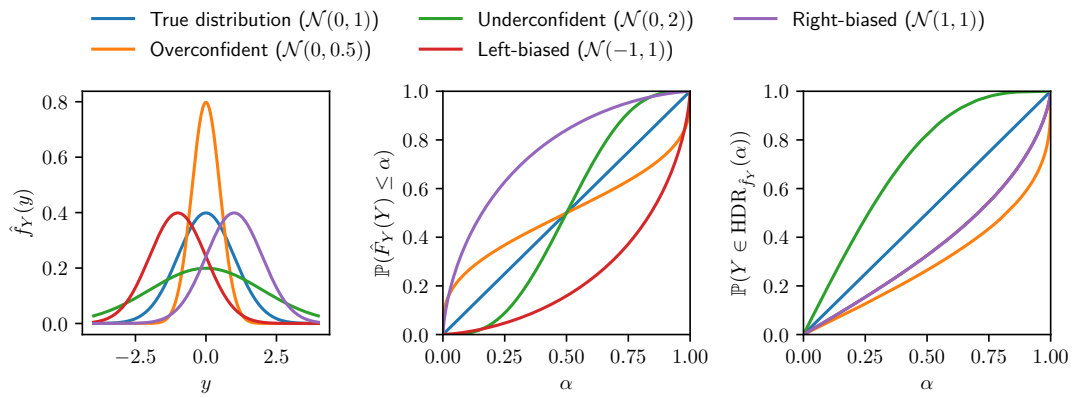


Figure 2.6: Panel 1 shows different predictive PDFs $\hat{f}_Y$ independent of X, while Panels 2 and 3 are the corresponding reliability diagrams for probabilistic calibration and HDR-calibration. The true distribution is assumed to be standard Gaussian ($\mathcal{N}(0,1)$).

2.4.3 Unconditional Calibration Methods

Beyond diagnostics of calibration, calibration methods update the predictive distributions to improve calibration. Calibration methods can be divided into recalibration methods, which act after training (i.e., post-hoc), and regularization methods, which act during training. In this section, we focus on recalibration methods due to their importance in this thesis. Regularization methods are discussed within their relevant chapters.

Various recalibration methods have been proposed in classification, with Platt scaling (Platt et al., 1999) and temperature scaling (Guo et al., 2017) being simple and well established. Other methods include Dirichlet calibration (Kull et al., 2019) and spline-based methods (Gupta et al., 2021).

In single-output regression, quantile recalibration (QR, Kuleshov et al., 2018) is a simple and effective method to achieve probabilistic calibration.

Quantile Recalibration (QR)

Let $\hat{U} = \hat{F}_{Y|X}(Y)$, and define the CDF $F_{\hat{U}}$ of $\hat{U}$ as the calibration map. QR defines the recalibrated CDF as $\hat{F}' = F_{\hat{U}} \circ \hat{F}$. By construction, $\hat{F}'_{Y|X}(Y)$ is uniformly distributed over $[0,1]$. Specifically, for any $\alpha \in [0,1]$:

$$\mathbb{P}(\hat{F}'_{Y|X} \leqslant \alpha) = \mathbb{P}(\hat{F}_{Y|X} \leqslant F_{\hat{U}}^{-1}(\alpha)) = F_{\hat{U}}(F_{\hat{U}}^{-1}(\alpha)) = \alpha. \tag{2.50}$$

Algorithm 1 details the recalibration procedure, where the empirical CDF of $\hat{U}$, denoted Φ_{EMP}, is used as the calibration map. In Chapter 3, we consider differentiable calibration maps, yielding a recalibrated predictive PDF.

In multi-output regression, the only proposed recalibration method is HDR recalibration (Y. Chung et al., 2024), which targets HDR calibration. In contrast to QR which yields a CDF and possibly a PDF (Utpala and Rai, 2020), HDR recalibration can only yield samples from the

Algorithm 1 Quantile recalibration.

Input: Calibration dataset $\mathcal{D}_{\text{cal}}$, base predictor with predictive CDF $\hat{F}_{Y|X}$, test input x_{test}.
Calibration:
for $(x^{(i)}, y^{(i)}) \in \mathcal{D}_{\text{cal}}$
  $\hat{U}_i \leftarrow \hat{F}_{Y|X=x^{(i)}}\left(y^{(i)}\right)$
Define $\Phi_{\text{EMP}}(u) = \frac{1}{|\mathcal{D}_{\text{cal}}|} \sum_{i=1}^{|\mathcal{D}_{\text{cal}}|} \mathbb{1}\left(\hat{U}_i \leqslant u\right)$ // `Calibration map`
Prediction:
Define $\hat{F}'_{Y|X=x_{\text{test}}} = \Phi_{\text{EMP}} \circ \hat{F}_{Y|X=x_{\text{test}}}$
return recalibrated predictive CDF $\hat{F}'_{Y|X=x_{\text{test}}}$

recalibrated distribution. Since it creates samples from the recalibration distribution from an initial pool of samples, it is called sampling-based. Instead of presenting HDR recalibration, we propose a direct generalization of this method that we call pre-rank recalibration, which targets pre-rank calibration and is also sampling-based. HDR recalibration is a special case when the pre-rank is the density $g(x, y) = \hat{f}_{Y|X=x}(y)$ or a strictly monotone transformation of the density as in Proposition 1.

Pre-rank Recalibration

Pre-rank recalibration adapts the idea of QR to multi-output settings by operating on a scalar pre-rank $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. This is achieved by learning a calibration map from calibration data and then resampling predictions accordingly.

Calibration step. For each calibration pair $(X^{(i)}, Y^{(i)}) \in \mathcal{D}_{\text{cal}}$, K samples $\{\hat{Y}^{(k,i)}\}_{k=1}^{K}$ are generated from the model's predictive distribution $\hat{P}_{Y|X=X^{(i)}}$. Each sample is mapped to a pre-rank

$$\hat{G}^{(k,i)} = g\left(X^{(i)}, \hat{Y}^{(k,i)}\right),$$

yielding an empirical estimate of the conditional CDF $\hat{F}_{\hat{G}|X=X^{(i)}}$. The estimated PIT value of the observed pre-rank value $g(X^{(i)}, Y^{(i)})$, which should ideally be uniformly distributed, is defined as

$$\hat{U}_i = \hat{F}_{\hat{G}|X=X^{(i)}}\left(g\left(X^{(i)}, Y^{(i)}\right)\right).$$

The collection $\{\hat{U}_i\}_{i=1}^{|\mathcal{D}_{\text{cal}}|}$ defines the empirical CDF $\hat{F}_{\hat{U}}$, which will be used as calibration map in the prediction step.

Prediction step. Given a new input x_{test}, K samples $\{\hat{Y}^{(k)}\}_{k=1}^{K}$ are drawn from $\hat{P}_{Y|X=x_{\text{test}}}$, with pre-ranks

$$\hat{G}^{(k)} = g\left(x_{\text{test}}, \hat{Y}^{(k)}\right).$$

Sorting these values gives an ordered list $\hat{G}^{(\pi(1))} \leqslant \cdots \leqslant \hat{G}^{(\pi(K))}$. Pre-rank recalibration resamples from the set of candidate samples $\{\hat{Y}^{(k)}\}_{k=1}^{K}$ such that their pre-rank approximately becomes uniformly distributed. More precisely, B bins corresponding to samples with similar pre-ranks

Algorithm 2 Pre-rank recalibration.

Input: Calibration dataset $\mathcal{D}_{\text{cal}}$, pre-rank $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, number of samples K, number of bins B, base predictor with predictive distribution $\hat{P}_{Y|X}$, test input x_{test}.
Calibration:
for $(X^{(i)}, Y^{(i)}) \in \mathcal{D}_{\text{cal}}$
  for $k = 1$ to K
    $\hat{Y}^{(k,i)} \sim \hat{P}_{Y|X=X^{(i)}}$
    $\hat{G}^{(k,i)} \leftarrow g\left(X^{(i)}, \hat{Y}^{(k,i)}\right)$
  Define $\hat{F}_{\hat{G}|X=X^{(i)}}(c) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\left(\hat{G}^{(k,i)} \leqslant c\right)$
  $\hat{U}_i \leftarrow \hat{F}_{\hat{G}|X=X^{(i)}}\left(g\left(X^{(i)}, Y^{(i)}\right)\right)$
Define $\hat{F}_{\hat{U}}(u) = \frac{1}{|\mathcal{D}_{\text{cal}}|} \sum_{i=1}^{|\mathcal{D}_{\text{cal}}|} \mathbb{1}\left(\hat{U}_i \leqslant u\right)$ // `Calibration map`
Prediction:
for $k = 1$ to K
  $\hat{Y}^{(k)} \sim \hat{P}_{Y|X=x_{\text{test}}}$
  $\hat{G}^{(k)} \leftarrow g\left(x_{\text{test}}, \hat{Y}^{(k)}\right)$
Define a permutation π such that $\hat{G}^{(\pi(1))} \leqslant \cdots \leqslant \hat{G}^{(\pi(K))}$
$\mathcal{S}' \leftarrow \varnothing$ // `Initial set of samples`
for $b = 1$ to B
  $n_b \leftarrow \lfloor K\hat{F}_{\hat{U}}(\frac{b}{B}) \rfloor - \lfloor K\hat{F}_{\hat{U}}(\frac{b-1}{B}) \rfloor$ // `Number of resamples`
  if $n_b > 0$
    $\mathcal{B}_b \leftarrow \left\{ \left\lfloor \frac{K(b-1)}{B} \right\rfloor + 1, \ldots, \left\lfloor \frac{Kb}{B} \right\rfloor \right\}$
    $\mathcal{S}_b \leftarrow \{\hat{Y}^{(\pi(k))}\}_{k \in \mathcal{B}_b}$ // `Samples pool`
    $\{\tilde{Y}^{(k)}\}_{k=1}^{n_b} \sim P_{\mathcal{S}_b}$ // `Resampling with replacement`
    $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{\tilde{Y}^{(k)}\}_{k=1}^{n_b}$
return recalibrated predictive samples $\mathcal{S}'$

are created, and the number of resamples n_b within each bin is defined by $\hat{F}_{\hat{U}}$. The union of these resampled subsets forms the set of recalibrated predictive samples $\mathcal{S}'$. Algorithm 2 details the exact recalibration procedure.

While effective in attaining pre-rank calibration, this approach has notable drawbacks: (1) it does not yield a closed-form recalibrated density $\hat{f}'_{Y|X}$; (2) duplicate samples may occur due to resampling; (3) discretization of bins introduces approximation error; and (4) it is computationally expensive since K candidate samples must be drawn for every recalibrated prediction.

Figure 2.7 shows an example of applying QR and HDR recalibration on a miscalibrated base predictor (panel 2). For both recalibration methods (panels 3 and 4), we observe that samples from both methods approach the true distribution (panel 1). Panel 3 also shows the recalibrated predictive PDF using the kernel density estimation approach in Section 3.5.1.

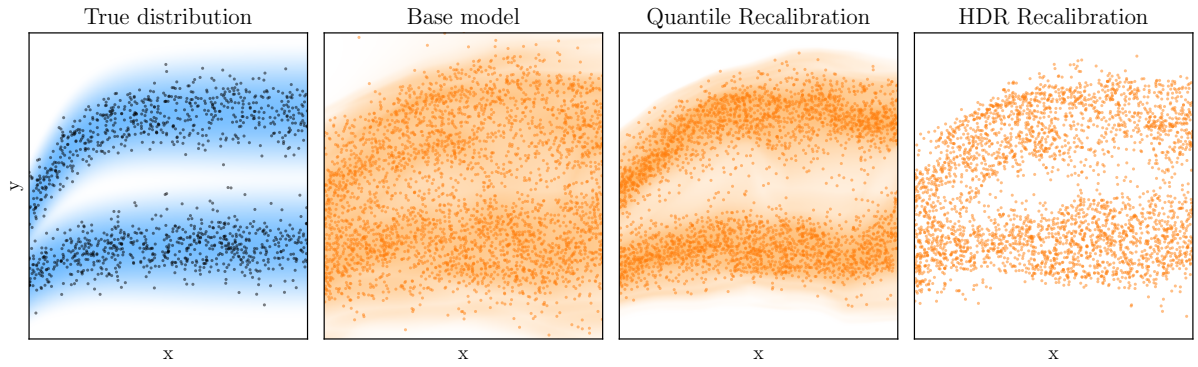


Figure 2.7: Example of predictive PDFs for the ideal model (Panel 1), a miscalibrated model (Panel 2), the quantile recalibrated model (Panel 3) and the HDR recalibrated model (Panel 4). Black dots on the Panel 1 represent the true data, while orange dots on successive panels represent samples from the respective models. The blue shade on Panel 1 represents the true PDF while orange shades on Panels 2 and 3 represents predictive PDFs.

2.4.4 Conditional Calibration Notions

Conditional calibration notions generalize unconditional ones to capture stronger forms of calibration. This section introduces several such notions, emphasizing the key concept of auto-calibration. Table 2.3 provides a summary of these conditional calibration notions.

Table 2.3: Conditional calibration notions.

Calibration notion	**Definition**
Ideal model	$\hat{P}_{Y\|X} \overset{\text{a.s.}}{=} P_{Y\|X}$
Auto-calibration	$\hat{P}_{Y\|X} \overset{\text{a.s.}}{=} P_{Y\|\hat{P}_{Y\|X}}$
Conditional T-calibration	$T(\hat{P}_{Y\|X}) \overset{\text{a.s.}}{=} T\left(P_{Y\|T(\hat{P}_{Y\|X})}\right)$
Threshold calibration	$\mathbb{P}(\hat{F}_{Y\|X}(Y) \leqslant c \mid \hat{F}_{Y\|X}(y_0) \leqslant \alpha) = c \quad \forall y_0 \in \mathcal{Y}, \alpha \in [0,1], c \in [0,1]$
Group calibration	$P_{Y\|X\in B} = \hat{P}_{Y\|X\in B} \quad \forall B \in \mathcal{B}$

Auto-calibration

In essence, auto-calibration (Tsyplakov, 2013) requires the model to be marginally calibrated conditional on its own predictions. We limit the discussion to an informal presentation, while a more formal discussion with proofs is provided in Section B.2.

Definition 6 (Auto-calibration)**.** A model $\hat{P}$ is *auto-calibrated* (w.r.t. Y) if the distribution of Y conditional on $\hat{P}$ is $\hat{P}$, i.e.,

$$P_{Y|\hat{P}} \overset{\text{a.s.}}{=} \hat{P}. \tag{2.51}$$

Intuitively, it implies that the model is truthful in the sense that the predictive uncertainty reflects the actual uncertainty. While this notion of calibration is strong, it does not imply that the model is ideal. For example, the marginal distribution of Y, also known as the climatological forecast $\hat{P} = P_Y$, is auto-calibrated but not informative.

Given a model with predictive distribution $P_{Y|X}$, we can produce a unique auto-calibrated version of this model as follows.

Proposition 2 (Auto-calibrated version of any model). Let $\hat{P}$ be a conditional distribution and let $\bar{P}$ be the distribution of Y conditional on $\hat{P}$:

$$\bar{P} = P_{Y|\hat{P}}.$$

Then $\bar{P}$ is auto-calibrated w.r.t. Y.

An auto-calibrated model has the interesting property that its average scoring rule is equal to its average generalized entropy. Thus, the expected score of an auto-calibrated model can be evaluated from the predictive distribution alone.

Lemma 1 (Sharpness identity for auto-calibrated predictions)**.** Let $\bar{P}$ be auto-calibrated w.r.t. Y. Then

$$\mathbb{E}[S(\bar{P}, Y)] = \mathbb{E}[H(\bar{P})].$$

A well-known decomposition (DeGroot and Fienberg, 1981; Bröcker, 2009; Kull and Flach, 2015) involving the auto-calibrated is given by the following theorem.

Theorem 1. Let $\hat{P}$ be a conditional distribution, $\bar{P} = P_{Y|\hat{P}}$ be its auto-calibrated version, and P be the true conditional distribution. Then,

$$\mathbb{E}[S(\hat{P}, Y)] = \mathbb{E}[D(\hat{P}, \bar{P})] + \mathbb{E}[D(\bar{P}, P)] + \mathbb{E}[S(P, Y)]. \tag{2.52}$$

Kull and Flach (2015) extend this decomposition with a term corresponding to the marginally calibrated version of $\hat{P}$ in the context of classification, with added conditions on the specific scoring rule as a function of the calibration method.

Theorem 1 does not require S to be proper. However, for a proper scoring rule, $\mathbb{E}[D(\hat{P}, \bar{P})] \geqslant 0$, hence $\mathbb{E}[S(\hat{P}, Y)] \geqslant \mathbb{E}[D(\bar{P}, P)] + \mathbb{E}[S(P, Y)] = \mathbb{E}[S(\bar{P}, Y)]$. This shows that the auto-calibrated version of a model, in addition to improving truthfulness, also improves the predictive accuracy according to any proper scoring rule.

While auto-calibration is a useful property, it is difficult to achieve in practice because the space of output distributions $\mathcal{P}(\mathcal{Y})$ can be inherently as complex as the input space $\mathcal{X}$, especially if $d > 1$. Instead of conditioning on the full predictive distribution, various weaker conditions with varying degrees of achievability have appeared in the literature.

Other conditional calibration notions

Conditional T-calibration (Gneiting and Resin, 2023) simplifies the requirement of auto-calibration by first projecting the conditional using a functional $T : \mathcal{P}(\mathcal{Y}) \to \mathbb{R}$. Threshold calibration (Sahoo et al., 2021) is discussed in details in Section 2.6. Given pre-defined disjoint bins $\mathcal{B} = B_1, \ldots, B_m$, group calibration (Pleiss et al., 2017) requires the model to be calibrated within each bin.

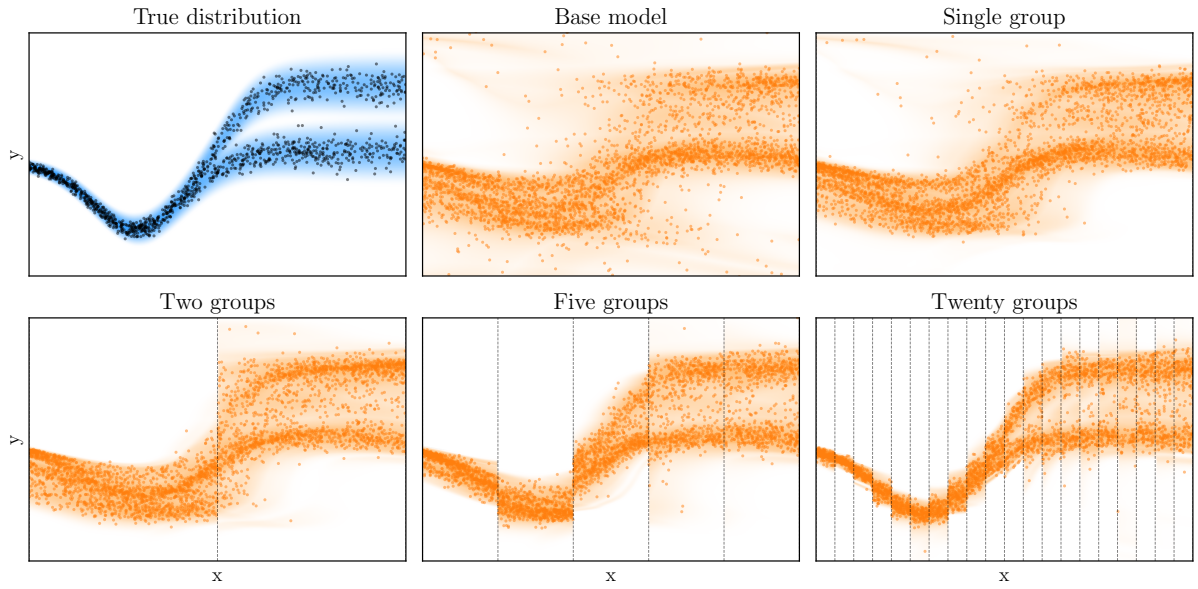


Figure 2.8: Example of predictive PDFs given by a group-recalibrated model.

2.4.5 Conditional Calibration Methods

In this section, we briefly discuss existing methods aiming for conditional calibration notions.

Kuleshov and Deshpande (2022) propose a post-hoc method for auto-calibration in single-output regression. First, the predictive distribution is projected to a low-dimensional space $\mathbb{R}^{d'}$ using a projection function $\psi : \mathcal{P}(\mathcal{Y}) \to \mathbb{R}^{d'}$. Then, a conditional calibration map, defined by $R_{\hat{P}} = F_{Y|\psi(\hat{P})}$, is estimated using conditional density estimation. If $R_{\hat{P}}$ is estimated correctly and $\psi(\hat{P})$ is sufficiently representative of $\hat{P}$, the recalibrated predictive CDF $\hat{F}'_{Y|X} = R_{\hat{F}_{Y|X}} \circ \hat{F}_{Y|X}$ is nearly auto-calibrated. Allen et al. (2025) also proposes an auto-calibration method in single-output regression, but they rely on binning.

A simple way of achieving group calibration is to apply any recalibration method separately within each group. The drawback is that the number of calibration points within each group becomes smaller as the number of groups increases. Figure 2.8 provides an example of applying QR within either 1, 2, 5 or 20 groups that partition $\mathcal{X}$. This model has been only trained for five epochs and is thus highly miscalibrated. In this example, it is clear that increasing the number of groups to 20 improves the quality of the predictive distribution. However, grouping should be used with caution when the number of calibration points is small.

With the aforementioned calibration methods, we only discussed asymptotic calibration guarantees. In the next section, we discuss a framework that is weaker in the sense that it produces prediction sets, and not full predictive distributions, but also stronger in the sense that it provides finite-sample guarantees. In Chapters 3 and 7, we will also see cases in which calibration methods and conformal prediction provide the same guarantees.

2.5. Conformal Prediction

Conformal prediction (CP) is a framework for constructing prediction sets with distribution-free, finite-sample coverage under the assumption of exchangeability (Vovk et al., 2005; Shafer and Vovk, 2008; Fontana et al., 2023; Angelopoulos and Bates, 2023; Angelopoulos et al., 2024a). Given a target miscoverage level $\alpha \in (0, 1)$ and a new input $x \in \mathcal{X}$, CP produces a prediction set $\hat{R}(x) \subseteq \mathcal{Y}$ that contains the true outcome y with probability at least $1 - \alpha$.

Originally introduced by Vladimir Vovk and colleagues in the mid-1990s, CP remained a fairly niche topic for much of its history, until recently gaining wide attention in the ML community due to its compatibility with black-box predictors. In Section 2.5.1, we focus on split conformal prediction (SCP, Papadopoulos et al., 2002), a particularly important variant of CP in ML due to its better computational efficiency, at the expense of requiring additional data. To provide an understanding of how SCP can be generalized beyond controlling coverage, we also present conformal risk control (Angelopoulos et al., 2024b).

2.5.1 Split Conformal Prediction

SCP (Papadopoulos et al., 2002) is a computationally efficient and popular variant of CP. It follows the simple procedure detailed in Algorithm 3. First, the available data $\mathcal{D}$ is divided into two disjoint subsets: a training set $\mathcal{D}_{\text{train}}$ and a calibration set $\mathcal{D}_{\text{cal}}$. A base predictor h_θ is obtained by training on $\mathcal{D}_{\text{train}}$ using a learning algorithm $\mathscr{A}$. This base predictor is then used to define a conformity score function $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, which measures the agreement between an input-output pair (x, y). By convention, smaller scores indicate a better fit. The conformity scores are computed for all points in the calibration set, yielding a set of scores $\mathcal{S}_{\text{cal}} = \{s(x_{\text{cal}}, y_{\text{cal}}) \mid (x_{\text{cal}}, y_{\text{cal}}) \in \mathcal{D}_{\text{cal}}\}$. The threshold $\hat{q}$ is then set as the $\lceil (|\mathcal{D}_{\text{cal}}| + 1)(1 - \alpha) \rceil$-th smallest value in $\mathcal{S}_{\text{cal}} \cup \{+\infty\}$. For a new input x, the prediction set is formed by all possible outputs y whose conformity score does not exceed this threshold:

$$\hat{R}(x) = \{y \in \mathcal{Y} : s(x, y) \leqslant \hat{q}\}. \tag{2.53}$$

Algorithm 3 Split Conformal Prediction (SCP).

Input: Dataset $\mathcal{D}$, miscoverage level α, learning algorithm $\mathscr{A}$, conformity score function s, test input x.
Calibration:
$\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{cal}} \leftarrow$ Split $\mathcal{D}$
$h_\theta \leftarrow \mathscr{A}(\mathcal{D}_{\text{train}})$
$\mathcal{S}_{\text{cal}} \leftarrow \{s(x_{\text{cal}}, y_{\text{cal}}; h_\theta) \mid (x_{\text{cal}}, y_{\text{cal}}) \in \mathcal{D}_{\text{cal}}\}$ `// We omit` h_θ `in the following.`
$k_\alpha \leftarrow \lceil (|\mathcal{D}_{\text{cal}}| + 1)(1 - \alpha) \rceil$
$\hat{q} \leftarrow k_\alpha$th smallest value in the set $\mathcal{S}_{\text{cal}} \cup \{+\infty\}$
Prediction:
$\hat{R}(x) \leftarrow \{y \in \mathcal{Y} \mid s(x, y) \leqslant \hat{q}\}$
return $\hat{R}(x)$

2.5.2 Marginal Coverage Guarantee

The marginal coverage guarantee of SCP hinges on the assumption of exchangeability.

Definition 7 (Exchangeability)**.** A finite sequence of random variables $(Z_1, \ldots, Z_n)$ is exchangeable if its joint probability distribution is invariant under any permutation of the indices. That is, for any permutation π,

$$Z_1, \ldots, Z_n \overset{\text{d.}}{=} Z_{\pi(1)}, \ldots, Z_{\pi(n)}. \tag{2.54}$$

Informally, this means the order of the data points carries no information. A sufficient condition for exchangeability is that the data points are drawn independently and identically distributed (i.i.d.), and a necessary condition is that the data points are identically distributed. Marginal coverage ensures that, on average, the generated prediction sets will contain the true outcome with the desired probability.

Theorem 2 (Marginal Coverage)**.** If the input-output pair (X, Y) and the data points in $\mathcal{D}_{\text{cal}}$ are drawn exchangeably, then the prediction set $\hat{R}(X)$ from SCP satisfies:

$$\mathbb{P}(Y \in \hat{R}(X)) \geqslant 1 - \alpha, \tag{2.55}$$

where the probability is taken over the joint distribution of (X, Y) and $\mathcal{D}_{\text{cal}}$.

This result is powerful because it holds regardless of the underlying data distribution, the choice of model h_θ, or the conformity score, as long as the exchangeability assumption is met. A proof is provided in Angelopoulos and Bates (2021).

Tightness of the marginal coverage. The guarantee in (2.55) averages over the randomness of both the input-output pair (X, Y) and the calibration set $\mathcal{D}_{\text{cal}}$. It is natural to ask about the distribution of the coverage for a fixed calibration set. In this case, the coverage is a random variable whose distribution can be precisely characterized.

Theorem 3. Let $k_\alpha = \lceil (1 - \alpha)(|\mathcal{D}_{\text{cal}}| + 1) \rceil$. If the input-output pair (X, Y) and the data points in $\mathcal{D}_{\text{cal}}$ are drawn i.i.d., assuming no ties among conformity scores, the coverage conditional on the calibration set $\mathcal{D}_{\text{cal}}$ follows a Beta distribution:

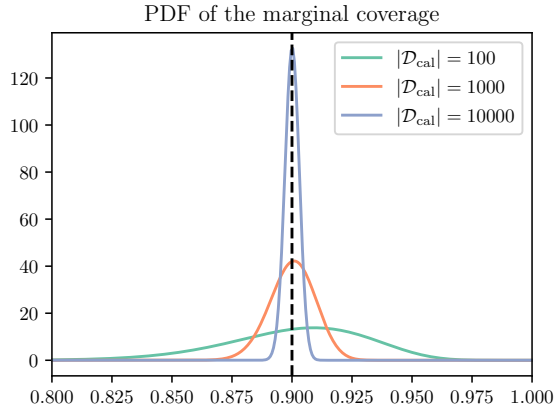
$$\mathbb{P}(Y \in \hat{R}(X) \mid \mathcal{D}_{\text{cal}}) \sim \text{Beta}(k_\alpha, |\mathcal{D}_{\text{cal}}| + 1 - k_\alpha). \tag{2.56}$$

This result, visualized in Figure 2.9a, shows that the coverage conditional on $\mathcal{D}_{\text{cal}}$ is concentrated around $1 - \alpha$, and this concentration tightens as $|\mathcal{D}_{\text{cal}}|$ increases. A proof is provided in Angelopoulos et al. (2024a).

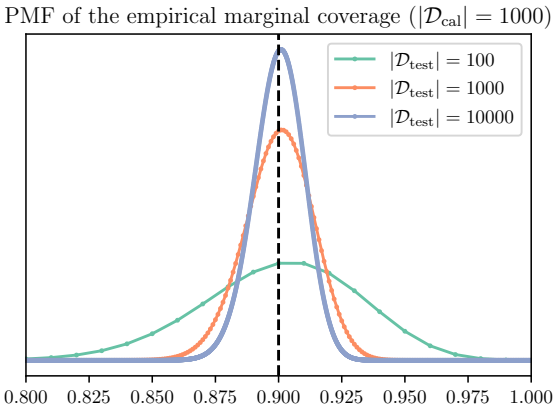
Tightness of the empirical marginal coverage. In practice, marginal coverage is often evaluated empirically on a finite test set $\mathcal{D}_{\text{test}}$:

$$|\mathcal{D}_{\text{test}}|^{-1} \sum_{(X^{(i)}, Y^{(i)}) \in \mathcal{D}_{\text{test}}} \mathbb{1}(Y^{(i)} \in \hat{R}(X^{(i)})). \tag{2.57}$$

Conveniently, the distribution of the empirical marginal coverage can also be characterized precisely.



(a) PDF of the marginal coverage conditional on the calibration dataset.



(b) PMF of the empirical marginal coverage conditional on the calibration and test datasets.

Figure 2.9: Distribution of the marginal coverage of any SCP method.

Theorem 4. Let $k_\alpha = \lceil (1-\alpha)(|\mathcal{D}_{\text{cal}}| + 1) \rceil$. If the input-output pair (X, Y) and the data points in $\mathcal{D}_{\text{cal}}$ are i.i.d., assuming no ties among conformity scores,

$$\sum_{(X^{(i)},Y^{(i)}) \in \mathcal{D}_{\text{test}}} \mathbb{1}(Y^{(i)} \in \hat{R}(X^{(i)})) \sim \text{BetaBinom}(|\mathcal{D}_{\text{test}}|, k_\alpha, |\mathcal{D}_{\text{cal}}| + 1 - k_\alpha). \tag{2.58}$$

As shown in Figure 2.9b, this concentration tightens as both the size of the calibration and test sets, $|\mathcal{D}_{\text{cal}}|$ and $|\mathcal{D}_{\text{test}}|$, increase. A proof is provided in Angelopoulos and Bates (2021).

2.5.3 Desired Properties of Prediction Sets

While marginal coverage is guaranteed, two other properties are highly desirable for prediction sets to be useful in practice.

Conditional coverage. Ideally, the prediction set should achieve *conditional coverage* at the level $1 - \alpha$, i.e.,

$$\mathbb{P}(Y \in \hat{R}(X) \mid X) \geqslant 1 - \alpha \tag{2.59}$$

holds almost surely. Intuitively, this ensures the guarantee holds uniformly across different regions of the input space, preventing the method from being overconfident for some inputs and underconfident for others simply to satisfy marginal coverage. This is a much stronger requirement than marginal coverage (2.55), and as Foygel Barber et al. (2021b) demonstrate, it is impossible to achieve without making additional assumptions about the data-generating process.

Sharpness. Beyond coverage, prediction sets should be *sharp*: as small (or informative) as possible subject to the coverage constraint. A trivial method that always returns $\mathcal{Y}$ attains 100% coverage but is useless. Sharpness is typically measured by the expected size $\mathbb{E}\left[|\hat{R}(X)|\right]$. Meaningful sharpness comparisons must always be paired with a verification of coverage (at least marginal, ideally conditional).

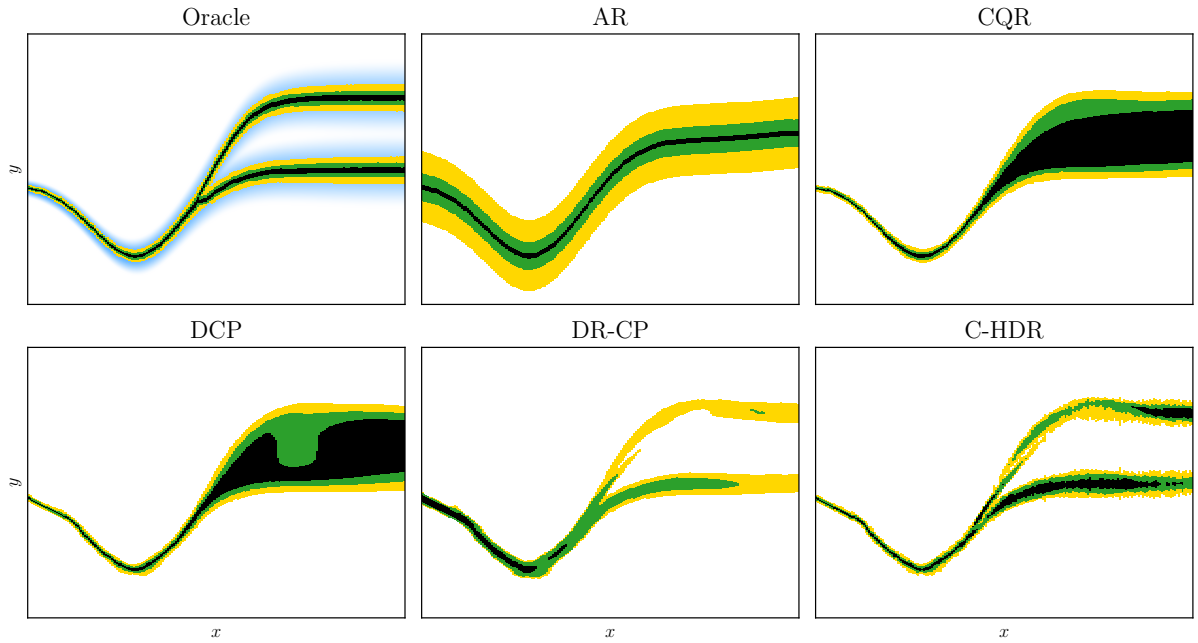


Figure 2.10: Example of prediction sets produced by conformal methods. The black, green and yellow regions represent coverage levels 20%, 50% and 80%, respectively.

2.5.4 Conformal Prediction Methods

The choice of conformity score s is central to the performance of conformal prediction methods. Different scores correspond to different ways of quantifying the agreement between a candidate label y and the input x, and lead to different prediction sets $\hat{R}(x)$ via (2.53). For single-output regression, several well-established conformity scores have been proposed, summarized in Table 2.4.

Table 2.4: Well-known conformity scores for single-output regression ($\mathcal{Y} = \mathbb{R}$).

Method	Conformity score $s(x, y)$	Prediction set $\hat{R}(x)$
AR	$\lvert y - \hat{\mu}(x)\rvert$	$[\hat{\mu}(x) - \hat{q}, \hat{\mu}(x) + \hat{q}]$
CQR	$\max\{\hat{Q}_{Y\mid X=x}(\alpha_{\text{lo}}) - y,\ y - \hat{Q}_{Y\mid X=x}(\alpha_{\text{hi}})\}$	$[\hat{Q}_{Y\mid X=x}(\alpha_{\text{lo}}) - \hat{q},\ \hat{Q}_{Y\mid X=x}(\alpha_{\text{hi}}) + \hat{q}]$
DCP	$\left\lvert \hat{F}_{Y\mid X=x}(y) - \frac{1}{2}\right\rvert$	$[\hat{F}^{-1}_{Y\mid X=x}(-\hat{q} + 1/2), \hat{F}^{-1}_{Y\mid X=x}(\hat{q} + 1/2)]$
HPD-split	$\text{HPD}_{\hat{f}_{Y\mid X=x}}(y)$	$\text{HDR}_{\hat{f}_{Y\mid X=x}}(\hat{q})$

Absolute Residuals (AR). The AR approach (Vovk et al., 2005) builds prediction sets by calibrating deviations from a point predictor $\hat{\mu}(x)$. It can perform well when residuals are roughly symmetric and homoscedastic but it does not adapt to skewness or heteroskedasticity.

Conformalized Quantile Regression (CQR). CQR (Romano et al., 2019) adapts to skewness and heteroskedasticity and thus typically yields tighter sets than AR, provided the

quantile estimates are sufficiently well estimated. In case quantiles cross, i.e., $\hat{Q}_{Y|X=x}(\alpha_{\text{lo}}) - \hat{q} > \hat{Q}_{Y|X=x}(\alpha_{\text{hi}}) + \hat{q}$, the interval is empty.

Distributional Conformal Prediction (DCP). DCP (Chernozhukov et al., 2021) relies on a predictive CDF and, similarly to CQR, adapts to skewness and heteroskedasticity. Compared to CQR, it avoids quantile crossing.

Conformalized Highest Density Region (HPD-Split). HPD-split (Izbicki et al., 2022) relies on a predictive PDF and uses the HPD, defined in (2.43), as conformity score. Compared to previous methods, since it returns superlevel sets of the PDF, it can produce non-connected sets and adapt to multimodality.

2.5.5 Conformal Risk Control

Beyond miscoverage, one can control the *expected value of a task-specific loss* over prediction sets. Multiple approaches have been proposed, including learn then test (Angelopoulos et al., 2025), risk-controlling prediction sets (Bates et al., 2021), and conformal risk control (CRC, Angelopoulos et al., 2024b). In this section, we focus on CRC, which is a generalization of SCP.

Consider the task of producing a prediction set over a space $\mathcal{Y}'$, which can be different from $\mathcal{Y}$. Let $l : 2^{\mathcal{Y}'} \times \mathcal{Y} \to (-\infty, B]$ be an upper-bounded loss that evaluates a prediction set over $\mathcal{Y}'$ compared to an output in $\mathcal{Y}$. We consider a nested family of sets indexed by a scalar $\lambda \in \Lambda \subseteq \mathbb{R}$, built from the conformity score $s : \mathcal{X} \times \mathcal{Y}' \to \mathbb{R}$:

$$\hat{R}_\lambda(x) = \{y' \in \mathcal{Y}' : s(x, y') \leqslant \lambda\}, \tag{2.60}$$

so that larger λ yields larger (more conservative) sets. Define, for $(X^{(i)}, Y^{(i)}) \in \mathcal{D}_{\text{cal}}$, the loss functions

$$L_i(\lambda) = l\big(\hat{R}_\lambda(X^{(i)}), Y^{(i)}\big), \qquad i = 1, \dots, |\mathcal{D}_{\text{cal}}|, \tag{2.61}$$

and the empirical risk $\hat{\mathcal{R}}_{\text{cal}}(\lambda) = \frac{1}{|\mathcal{D}_{\text{cal}}|} \sum_{i=1}^{|\mathcal{D}_{\text{cal}}|} L_i(\lambda)$. We assume $L_i(\lambda)$ is non-increasing and right-continuous in λ, and that $\sup_\lambda L_i(\lambda) \leqslant B$.

Example 3. Consider tumor segmentation for an image of dimension $d_h \times d_w$. Inputs in $\mathcal{X} = [0,1]^{d_h \times d_w}$ correspond to images to segment, while outputs in $\mathcal{Y} = 2^{[d_h] \times [d_w]}$ correspond to the set of pixels in the segmentation mask. The goal is to create a prediction set over $\mathcal{Y}' = [d_h] \times [d_w]$, corresponding to the set of pixels that are the most likely to contain the tumor. Given a model $h_\theta : \mathcal{X} \to [0,1]^{d_h \times d_w}$ that returns the probability of each pixel to contain a tumor, the conformity score $s(x, y') = -h_\theta(x)_{y'}$ defines nested prediction sets $\hat{R}_\lambda(x) = \{y' \in \mathcal{Y}' : -h_\theta(x)_{y'} \leqslant \lambda\}$ containing pixels whose assigned probability is the highest. Angelopoulos et al. (2024b) propose controlling the false negative rate (FNR), defined as:

$$l(C, y) = 1 - \frac{|y \cap C|}{|y|}. \tag{2.62}$$

Intuitively, this loss penalizes the fraction of tumor pixels that are missed by the prediction set C.

Risk-control guarantee. Given a target risk level $\alpha \in (-\infty, B)$, choose

$$\hat{\lambda} \;=\; \inf\left\{\lambda \in \Lambda :\; \frac{n}{n+1}\hat{\mathcal{R}}_{\text{cal}}(\lambda) + \frac{B}{n+1} \leqslant \alpha\right\}. \tag{2.63}$$

Since $\hat{\mathcal{R}}_{\text{cal}}(\lambda)$ is a non-increasing function of λ, $\hat{\lambda}$ can be found using the bisection method. If the input-output pair (X, Y) and the data points in $\mathcal{D}_{\text{cal}}$, $\hat{R}_{\hat{\lambda}}$ are drawn exchangeably, then

$$\mathbb{E}\Big[l\big(\hat{R}_{\hat{\lambda}}(X), Y\big)\Big] \;\leqslant\; \alpha. \tag{2.64}$$

Moreover, under mild continuity and i.i.d. assumptions, the procedure is tight up to $O(1/n)$, i.e., $\mathbb{E}\Big[l(\hat{R}_{\hat{\lambda}}(X), Y)\Big] \geqslant \alpha - 2B/(n+1)$.

In Example 3, by calibrating λ using conformal risk control, one can guarantee that the expected FNR across new patients does not exceed the target level α. However, it should be kept in mind that the guarantee is marginal, and not conditional on features $x \in \mathcal{X}$.

Finally, Algorithm 4 summarizes the CRC procedure.

Algorithm 4 Conformal Risk Control.

Input: Dataset $\mathcal{D}$, risk budget α, learning algorithm $\mathscr{A}$, conformity score function s, bounded monotone loss $l(\cdot, \cdot)$ with upper bound B, test input x_{test}.
Calibration:
$\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{cal}} \leftarrow$ Split $\mathcal{D}$
$h_\theta \leftarrow \mathscr{A}(\mathcal{D}_{\text{train}})$
Define $\hat{R}_\lambda(x) = \{y' \in \mathcal{Y}' \mid s(x, y'; h_\theta) \leqslant \lambda\}$ `// We omit` h_θ `in the following.`
Define $\hat{\mathcal{R}}_{\text{cal}}(\lambda) = \frac{1}{|\mathcal{D}_{\text{cal}}|}\sum_{(x_i, y_i) \in \mathcal{D}_{\text{cal}}} l\big(\hat{R}_\lambda(x_i), y_i\big)$
$\hat{\lambda} \leftarrow \inf\left\{\lambda \in \Lambda :\; \frac{|\mathcal{D}_{\text{cal}}|}{|\mathcal{D}_{\text{cal}}|+1}\hat{\mathcal{R}}_{\text{cal}}(\lambda) + \frac{B}{|\mathcal{D}_{\text{cal}}|+1} \leqslant \alpha\right\}$ `// Using bisection`
Prediction:
$\hat{R}(x_{\text{test}}) \leftarrow \hat{R}_{\hat{\lambda}}(x_{\text{test}})$
return $\hat{R}(x_{\text{test}})$

Reduction to SCP. If, in addition, $\mathcal{Y}' = \mathcal{Y}$ and $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is the usual conformity score over labels, then taking $l(C, y) = \mathbb{1}(y \notin C)$ yields $L_i(\lambda) = \mathbb{1}(s(X^{(i)}, Y^{(i)}) > \lambda)$, which is bounded by 1 and non-increasing in λ. Plugging this into (2.63) with $B = 1$ gives

$$\hat{\lambda} = \inf\left\{\lambda \in \Lambda : \frac{1}{n+1}\sum_{i=1}^{n} \mathbb{1}(s(X^{(i)}, Y^{(i)}) > \lambda) + \frac{1}{n+1} \leqslant \alpha\right\} \tag{2.65}$$

$$= \inf\left\{\lambda \in \Lambda : \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(s(X^{(i)}, Y^{(i)}) \leqslant \lambda) \geqslant \frac{(n+1)(1-\alpha)}{n}\right\} \tag{2.66}$$

Thus, $\hat{\lambda}$ is the same quantile $\hat{q}$ used by SCP, and (2.64) reduces to marginal coverage.

CRC inherits the distribution-free, finite-sample properties of SCP, while allowing l to measure application-specific desiderata.

2.6. Calibration for Decision-Making

The previous sections discussed different approaches to model predictive uncertainty. However, the ultimate goal is often to convert these probabilistic predictions into actions to solve real-world problems. Decision theory provides a principled framework for reasoning and acting under uncertainty. In this section, we review the notion of threshold calibration introduced in Sahoo et al. (2021), allowing decision makers to confidently estimate the loss of any threshold decision rule. Thus, the goal is not increase the utility, but instead to allow decision makers to be more confident in their decision.

2.6.1 The Decision-Making Framework

A desirable property of a probabilistic model is that a decision maker can trust its uncertainty estimates to inform their actions. We consider tasks where the decision maker selects an action from an action space $\mathcal{A}$. Formally, the decision maker's strategy is a decision rule $\delta : \mathcal{X} \to \mathcal{A}$ that maps an input x to an action $\delta(x)$. The goal is to select a rule that minimizes a loss function $l : \mathcal{X} \times \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$.

Because the true label y is unobserved at decision time, the decision maker aims to minimize the expected loss, where the expectation is taken over the predictive distribution $\hat{P}_{Y|X}$. We denote conditional samples from this distribution as $\hat{Y} \sim \hat{P}_{Y|X}$. The optimal strategy under this assumption is the Bayes decision rule.

Definition 8. Bayes Decision Rule

The Bayes decision rule δ^* is the rule that minimizes the expected loss under the predictive distribution:

$$\delta^*(x) = \arg\min_{a \in \mathcal{A}} \mathbb{E}\Big[l(X, \hat{Y}, a) \mid X = x \Big]. \tag{2.67}$$

For a decision rule to be deployed confidently, the decision maker needs an accurate estimate of its true performance. We can quantify the accuracy of the predicted loss with the reliability gap, which measures the difference between the predicted expected loss and the true expected loss.

Definition 9. Reliability Gap

For a given decision rule δ and loss function l, the reliability gap is:

$$\gamma(\delta, l) = \Big| \mathbb{E}\Big[l(X, \hat{Y}, \delta(X)) \Big] - \mathbb{E}[l(X, Y, \delta(X))] \Big|. \tag{2.68}$$

A reliability gap of zero means the decision maker can perfectly estimate the consequences of their decisions before deployment. Sahoo et al. (2021) show that, when a model is auto-calibrated, $\gamma(\delta, l) = 0$ for any rule and loss.

2.6.2 Threshold Calibration

Achieving auto-calibration is intractable in the general case. Sahoo et al. (2021) introduced a weaker calibration notion that guarantees a zero reliability gap for a more practical class of decision problems. Their original approach is based on probabilistic calibration and hence

limited to univariate target spaces $\mathcal{Y}$ ($d = 1$). However, in addition to reviewing their method, we propose a direct generalization to pre-rank calibration with pre-rank $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, allowing the application of their method to multi-output regression. Recall the definition of pre-rank calibration:

$$\mathbb{P}(F_{\hat{G}|X}(G) \leqslant c) = c \quad \forall c \in [0,1] \tag{2.69}$$

with $\hat{G} = g(X, \hat{Y})$ and $G = g(X, Y)$.

Consider a setting where the decision maker makes a binary decision, $\mathcal{A} = \{0, 1\}$, based on whether the pre-rank exceeds a predefined threshold $t \in \mathbb{R}$. The costs associated with each outcome-action pair are given by a cost matrix $\{c_{i,j}\}_{i \in \{0,1\}, j \in \{0,1\}}$. The threshold loss function is:

$$l_t(x, y, a) = c_{\mathbb{1}(g(x,y) > t), a}. \tag{2.70}$$

***H*1.** We make the standard assumption that a correct action is strictly less costly than an incorrect one. This implies that the cost of a true positive is less than a false negative ($c_{1,1} < c_{1,0}$) and the cost of a true negative is less than a false positive ($c_{0,0} < c_{0,1}$).

The Bayes decision rule δ^* minimizes the expected loss under the predictive distribution $\mathbb{E}\Big[l_t(X, \hat{Y}, \delta^*(X))\Big]$ pointwise for each input x. It corresponds to a threshold decision rule with a fixed threshold given by Lemma 2.

Definition 10. Threshold decision rule

Let $x \in \mathcal{X}$. A threshold decision rule is a decision rule of the form

$$\delta_\alpha(x) = \mathbb{1}\left(F_{\hat{G}|X=x}(t) \leqslant \alpha\right). \tag{2.71}$$

To find it, we compute the expected loss for each action $a \in \{0, 1\}$ under the predictive distribution $\hat{F}_{Y|X=x}$ and choose the action with the minimum expected loss.

Lemma 2. Under **H**1, the Bayes Decision Rule for a threshold loss function is a threshold decision rule. Specifically,

$$\delta^*(x) = \mathbb{1}\left(F_{\hat{G}|X=x}(t) \leqslant \frac{c_{1,0} - c_{1,1}}{c_{1,0} - c_{1,1} + c_{0,1} - c_{0,0}}\right). \tag{2.72}$$

The proof is in Section B.3.1.

Definition 11. Threshold calibration

Let $\hat{U} = F_{\hat{G}|X}(G)$. A model is threshold calibrated if:

$$\mathbb{P}(\hat{U} \leqslant c \mid F_{\hat{G}|X}(t) \leqslant \alpha) = c \quad \forall t \in \mathbb{R}, \alpha \in [0,1], c \in [0,1]. \tag{2.73}$$

This property ensures that the prediction is calibrated on the subsets of the data defined by any threshold decision rule. As the following theorem shows, this is precisely the condition needed to guarantee a zero reliability gap.

Theorem 5. Let $\mathcal{L}$ be the space of all threshold loss functions and Δ be the space of all threshold decision rules. Assume that, for any $x \in \mathcal{X}$, $F_{\hat{G}|X=x}$ is strictly increasing. A predictive distribution $\hat{F}_{Y|X}$ satisfies threshold calibration if and only if

$$\gamma(\delta, l) = 0 \quad \forall \delta \in \Delta, l \in \mathcal{L}. \tag{2.74}$$

The proof is in Section B.3.2.

Sahoo et al. (2021) further detailed an algorithm to achieve threshold calibration, which we omit for conciseness. Overall, this section highlights the potential of calibration for reliable decision-making, including in the multi-output setting.

CHAPTER 3

A Study of Probabilistic Calibration in Neural Probabilistic Models

This chapter is based on the following paper:

> **Victor Dheur** and Souhaib Ben Taieb (2023). A Large-Scale Study of Probabilistic Calibration in Neural Network Regression. *The 40th International Conference on Machine Learning.*

3.1. Introduction

As established in Chapter 1, it is standard practice in probabilistic forecasting to assess whether models are *probabilistically calibrated* (Gneiting et al., 2007). This property states that all quantiles must be calibrated, i.e., the frequency of realizations below these quantiles must match the corresponding quantile level. Additionally, predictive distributions should be sufficiently sharp (i.e., concentrated around the realizations) and leverage the information in the inputs. However, the miscalibration of modern neural networks poses a significant challenge to their reliability in real-world applications. In the classification setting, Guo et al. (2017) found that common neural architectures trained on image and text data were miscalibrated, sparking increased interest in neural network calibration. In a follow-up study, Minderer et al. (2021) showed that more recent neural architectures demonstrate improved calibration. Yet, calibration for neural probabilistic regression models has received less attention compared to classification. Therefore, it remains uncertain whether the same results apply to the regression setting.

Chapter 2 laid the necessary theoretical groundwork for addressing this issue in the context of regression. Section 2.3 presented a framework for evaluating probabilistic predictions through strictly proper scoring rules such as the NLL and CRPS. Proper scoring rules provide a principled way to rank predictive distributions by assigning numerical penalties that jointly reward calibration and sharpness. However, they may obscure specific deficiencies since two scoring rules can achieve similar scores with different statistical properties. Probabilistic calibration, defined in Section 2.4, offers by contrast a more transparent diagnostic of whether predictive distributions

are reliable. Thus, while strictly proper scoring rules are indispensable for overall comparisons, explicit checks of probabilistic calibration complement them by revealing where systematic biases remain hidden. Furthermore, we surveyed different post-hoc approaches: quantile recalibration (QR, Section 2.4.3) and split conformal prediction (SCP, Section 2.5.1). In this chapter, we also consider regularization methods, which have been shown to perform well in the classification setting (Karandikar et al., 2021; Popordanoska et al., 2022; Yoon et al., 2023).

In Dheur and Ben Taieb (2023), we make the following main contributions:

1. We conduct the largest empirical study to date on probabilistic calibration of neural regression models using 57 tabular datasets (Sections 3.4 and 3.6). We consider multiple state-of-the-art calibration methods (Section 3.5), including post-hoc recalibration, conformal prediction, and regularization methods, with various scoring rules and predictive models.
2. Building on QR, we propose a new differentiable calibration map using kernel density estimation, which provides improved NLL compared to baselines. We also introduce two new regularization objectives based on the probabilistic calibration error (Section 3.5).
3. We show that QR is a special case of SCP, providing an explanation for its superior probabilistic calibration (Section 3.6).

3.2. Background

We consider the supervised setting (Section 2.1.1), and in particular the single-output regression setting. Recall that the target variable $Y \in \mathcal{Y} \subseteq \mathbb{R}$ depends on an input variable $X \in \mathcal{X} \subseteq \mathbb{R}^p$. Our objective is to approximate the conditional (oracle) distribution $P_{Y|X}$ using training data $\mathcal{D} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^N$ where $(X^{(i)}, Y^{(i)}) \overset{\text{i.i.d.}}{\sim} P_{X,Y}$.

A probabilistic predictor h_θ with parameters $\theta \in \Theta$ maps an input $x \in \mathcal{X}$ to a predictive distribution over $\mathcal{Y}$. We denote its cumulative distribution function (CDF) as $\hat{F}_{Y|X=x}$, its quantile function (QF) as $\hat{Q}_{Y|X=x}$, and its probability density function (PDF) as $\hat{f}_{Y|X=x}$. Similarly, the marginal CDF, QF, or PDF of a random variable R is denoted by F_R, Q_R, or f_R, respectively.

Probabilistic calibration. Given an input $x \in \mathcal{X}$, the model is ideal if its predictive distribution matches the true conditional distribution $P_{Y|X}$. However, learning the oracle distribution based on finite data is not possible without additional (strong) assumptions (Foygel Barber et al., 2021b). To avoid additional assumptions, we can instead enforce certain desirable properties that are attainable in practice and that the oracle distribution exhibits. One such property is probabilistic calibration (Gneiting et al., 2007).

Let $\hat{U} = \hat{F}_{Y|X}(Y) \in [0,1]$ denote the probability integral transform (PIT) of Y conditional on X. Recall that the model is *probabilistically calibrated* (also known as PIT-calibrated) if $\forall \alpha \in [0,1]$,

$$F_{\hat{U}}(\alpha) = \mathbb{P}\left(\hat{U} \leqslant \alpha\right) = \alpha. \tag{3.1}$$

Let $U \in [0,1]$ be standard uniform and independent of $\hat{U}$. The left and right hand sides of (3.1) correspond to the CDF of $\hat{U}$ and U, respectively, at α. This illustrates that the uniformity of

the PIT is equivalent to probabilistic calibration (Dawid, 1984).

Since the oracle distribution is probabilistically calibrated, it is sensible to require this property from any competent model. However, probabilistic calibration, though necessary, is not sufficient for making accurate probabilistic predictions. Additionally, as discussed by Gneiting and Resin (2023), probabilistic calibration primarily addresses unconditional aspects of predictive performance and is implied by more robust conditional notions of calibration, such as auto-calibration.

Probabilistic calibration error. The most common approach for evaluating probabilistic calibration is to consider distances of the form $\int_0^1 |F_{\hat{U}}(\alpha) - F_U(\alpha)|^p d\alpha$ where $p > 0$. The particular cases of $p = 1$ and $p = 2$ are known as the 1-Wasserstein distance and Cramér–von Mises distance, respectively. Let $\hat{U}_i = \hat{F}_{Y|X=X^{(i)}}(Y^{(i)})$ be an i.i.d. PIT realization. We denote the empirical PIT CDF as $\hat{F}_{\hat{U}}(\alpha) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\hat{U}_i \leqslant \alpha)$. A common approach to assess probabilistic calibration using a discrete approximation is to evaluate it at M equidistant values $\alpha_1 < \cdots < \alpha_M$ by computing

$$\mathrm{PCE}_p(\hat{F}_{Y|X}; \mathcal{D}) = \frac{1}{M}\sum_{j=1}^{M} \left| \alpha_j - \hat{F}_{\hat{U}}(\alpha_j) \right|^p . \tag{3.2}$$

This metric has been previously employed in the literature with $p = 1$ (S. Zhao et al., 2020; T. Zhou et al., 2021), and with $p = 2$ (Kuleshov et al., 2018; Utpala and Rai, 2020). Unlike the classical definition of the p-norm, we do not exponentiate by $\frac{1}{p}$ in (3.2) to maintain consistency with prior literature. In the subsequent sections, we focus our analysis on PCE_1 and use the abbreviation PCE for brevity.

One limitation of scalar metrics like PCE is their inability to provide detailed information regarding calibration errors at individual quantile levels, $\alpha_1, \ldots, \alpha_M$. Instead, PIT reliability diagrams offer a visual assessment of probabilistic calibration across all quantile levels by plotting the empirical CDF of the PIT $\hat{U}$. These diagrams display the right side of (3.1) against its left side, with a perfectly calibrated model represented by a diagonal line (asymptotically). Figure 3.2 provides examples of such reliability diagrams, which have been used in studies by Pinson and Hagedorn (2012) and Kuleshov et al. (2018).

3.3. Related Work

Post-hoc recalibration approaches involve adjusting the predictions of a trained model based on a separate calibration dataset. For classification problems, many such methods have been proposed (Ananya Kumar et al., 2019; Gupta et al., 2021). Among them, temperature scaling (Guo et al., 2017) is a simple, popular, and effective method that adjusts predictive confidence while maintaining accuracy. For regression tasks, quantile recalibration (Kuleshov et al., 2018) is a method that transforms predictive distributions using a calibration map to achieve probabilistic calibration. Conformal prediction, pioneered by Vovk et al. (2005), is a general approach that generates prediction sets with a finite-sample coverage guarantee (Vovk et al., 2020). Multiple approaches based on this framework have been applied with neural networks, including conformal quantile regression (Romano et al., 2019) and distributional conformal prediction (Izbicki et al., 2020; Chernozhukov et al., 2021). The latter has been shown to be closely related to quantile

recalibration (Marx et al., 2022; Dheur and Ben Taieb, 2023). Furthermore, post-hoc approaches have also been proposed aiming to target stronger, conditional notions of calibration such as auto-calibration (H. Song et al., 2019; Kuleshov and Deshpande, 2022).

Regularization approaches aim to improve calibration during training by incorporating regularization or modifying the loss function. This can be achieved through general methods such as ensembling (Lakshminarayanan, Pritzel, et al., 2017), mixup (Zhang et al., 2018), label smoothing (Müller et al., 2019), or penalizing high-confidence predictions (Pereyra et al., 2017). Numerous regularization objectives have been proposed for both classification (Aviral Kumar et al., 2018; Karandikar et al., 2021; Popordanoska et al., 2022; Yoon et al., 2023) and regression. In regression, some methods utilize regularization to target conditional notions of calibration (S. Zhao et al., 2020; Feldman et al., 2021). T. Zhou et al. (2021) introduced an alternative loss function involving the simultaneous training of two neural networks. Pearce et al. (2018), Y. Chung et al. (2021), and Thiagarajan et al. (2020) proposed objectives allowing control over the trade-off between the coverage and sharpness of prediction intervals, while quantile regularization (Utpala and Rai, 2020) specifically targets probabilistic calibration. Marx et al. (2023) proposed a unified framework presenting many existing notions of calibration as distribution matching constraints. Specifically, they use unbiased estimates of the maximum mean discrepancy, yielding effective regularization objectives. While these regularization methods can improve calibration, they may negatively impact other accuracy metrics, especially when combined with post-hoc recalibration. For instance, Karandikar et al. (2021) and Yoon et al. (2023) reported selecting hyperparameters that minimize calibration error while decreasing accuracy by about 1%.

3.4. Are Neural Regression Models Probabilistically Calibrated?

We conduct an extensive empirical study to evaluate the probabilistic calibration of neural regression models. To this end, we calculate the *probabilistic calibration error* defined in (3.2) for various state-of-the-art models across multiple benchmark datasets.

Benchmark datasets. We analyze a total of 57 datasets, including 27 from the OpenML curated benchmark (Grinsztajn et al., 2022), 18 from the AutoML Repository (Gijsbers et al., 2019), and 12 from the UCI Machine Learning Repository (Dua and Graff, 2017).

These datasets are widely used in the evaluation of probabilistic neural regression models and uncertainty quantification, as evidenced by previous studies such as Fakoor et al. (2023), Y. Chung et al. (2021), T. Zhou et al. (2021), Utpala and Rai (2020), and Yarin Gal and Ghahramani (2016).

Figure 3.1 provides an overview of the utilization of these datasets in previous studies. To the best of our knowledge, our study represents the most comprehensive assessment of probabilistic calibration for neural regression models.

Neural probabilistic regression models. We consider three UQ methods. The first predicts a parametric distribution, where the parameters are obtained as outputs of a hypernetwork. Previous studies have often focused on the Gaussian distribution (Lakshminarayanan, Pritzel, et al., 2017; Utpala and Rai, 2020; S. Zhao et al., 2020). To introduce more flexibility, we consider the mixture density network defined in Section 2.2.3 (Bishop, 1994). We have two variants of this model depending on the scoring rule used for training: the negative log-likelihood (NLL) or

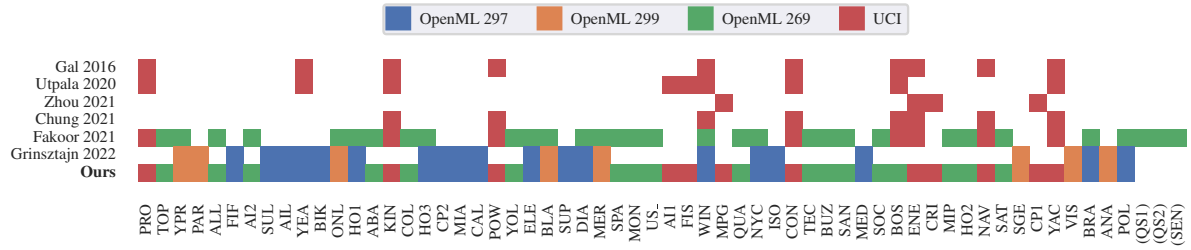


Figure 3.1: Multiple regression benchmark datasets with references. Datasets inside parentheses have not been considered in this study due to categorical outputs or no valid input column after preprocessing. Full dataset names are available in Table A.1.

the continuous ranked probability score (CRPS). These models are denoted as MIX-NLL and MIX-CRPS, respectively. For MIX-CRPS, we use a closed-form expression of the CRPS of a mixture of Gaussians introduced in Grimit et al. (2006).

The second model predicts quantiles of the distribution (Tagasovska and Lopez-Paz, 2019; Y. Chung et al., 2021; Feldman et al., 2021). Specifically, given an input $x \in \mathcal{X}$ and a quantile level $\alpha \in [0, 1]$, the model outputs a quantile $\hat{Q}_{Y|X=x}(\alpha)$. The loss function is the quantile score evaluated at $\alpha \sim \mathcal{U}(0, 1)$, which is asymptotically equivalent to the CRPS (Bracher et al., 2021). We denote this model as SQR-CRPS, where SQR stands for simultaneous quantile regression (Tagasovska and Lopez-Paz, 2019).

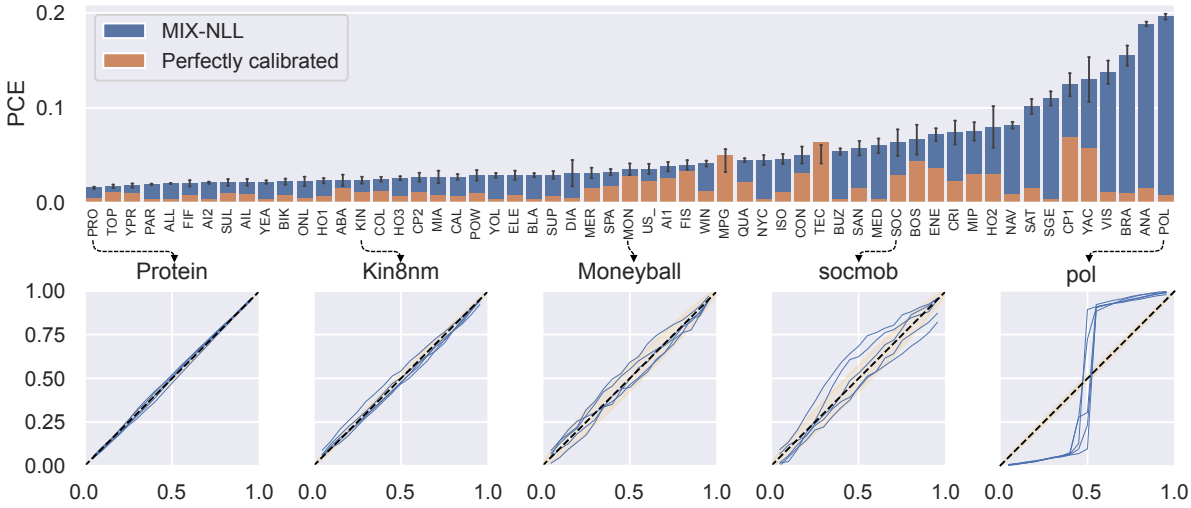


Figure 3.2: The top row shows the PCE for different datasets with one standard error (error bar). The bottom row gives examples of PIT reliability diagrams for five datasets.

Experimental setup. We adopt the large-sized regime introduced by Grinsztajn et al. (2022), which involves truncating the datasets to a maximum of 50,000 examples. Among the 57 datasets, the number of training examples ranges from 135 to 50,000, and the number of features ranges

from 3 to 3,611[1]. Each of the 57 datasets is divided into four sets: training (65%), validation (10%), calibration (15%), and test (10%). We normalize the input X and target Y using the mean and standard deviation from the training split. The final predictions are then transformed back to the original scale. For our neural network models, we use the same fully-connected architecture as previous studies conducted by Kuleshov et al. (2018), Y. Chung et al. (2021), and Fakoor et al. (2023). Further details regarding the model hyperparameters can be found in Section C.3.

Results. In Figure 3.2, we consider the PCE averaged over $\kappa = 5$ random train-validation-test splits $\mathcal{D}_1, \ldots, \mathcal{D}_\kappa$, denoted $\overline{\text{PCE}}(\cdot) = \frac{1}{\kappa}\sum_{i=1}^{\kappa} \text{PCE}(\cdot, \mathcal{D}_\kappa)$. The first row displays $\overline{\text{PCE}}(\hat{F}_{Y|X})$ for MIX-NLL in blue on each of the 57 datasets. For comparison, $\overline{\text{PCE}}(F_{Y|X})$ for a perfectly calibrated model, i.e. with uniformly distributed PITs is shown in orange, further averaged over 10^4 simulated values. While the model is probabilistically calibrated, the PCE is not zero because it is evaluated on a finite test dataset and thus the CDF $\hat{F}_{\hat{U}}$ is not exactly the identity function. The second row presents reliability diagrams for $\kappa = 5$ datasets, with 90% consistency bands as in Gneiting et al. (2023). Similar information is provided for MIX-CRPS and SQR-CRPS in Figures C.7 and C.8 in Section C.2.4, respectively. Additionally, reliability diagrams for all datasets can be found in Figure C.10 in Section C.2.6.

The analysis reveals that $\overline{\text{PCE}}$ is generally high across many datasets, although there are significant variations between datasets. To test the statistical significance of these results, 10^4 samples were generated from the sampling distribution of $\overline{\text{PCE}}$ under the null hypothesis of probabilistic calibration. The resulting sampling distribution for all datasets is presented in Section C.2.5.

By computing the p-value associated with a one-sided test in the upper tail of the distribution (as illustrated in Section C.2.5), it was observed that most datasets have a p-value of zero. This indicates that the average PCE obtained for the considered model is higher than all the simulated average PCEs of the probabilistically calibrated model. Applying a threshold of 0.01 and a Holm correction for the 57 hypothesis tests, the null hypothesis is rejected for 11 datasets out of the 57.

Overall, the results indicate that the neural probabilistic models considered in this study are generally not probabilistically calibrated on a significant number of benchmark tabular datasets. In Section 3.6, we will further explore how calibration methods can substantially improve the PCE of neural regression models.

3.5. Calibration Methods

We begin by discussing the three main families of calibration methods in single-output regression: QR, SCP, and regularization-based calibration. Following that, we introduce two novel variants of regularization-based calibration. For a general introduction to recalibration and conformal prediction methods, we refer the reader to Sections 2.4.3 and 2.5.4. This section further details how to estimate a smooth calibration map. Additionally, compared to Section 2.5.4 which focuses on prediction sets, this section discusses an approach to estimate calibrated quantiles using SCP.

QR and SCP are post-hoc methods, meaning they are applied after model training. These

[1]Please refer to Table A.1, for a detailed summary of each dataset.

approaches utilize a separate calibration dataset $\mathcal{D}_{\text{cal}} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^{n}$, where $(X^{(i)}, Y^{(i)}) \overset{\text{i.i.d.}}{\sim} P_{X,Y}$. In contrast, regularization-based calibration operates directly during training and relies solely on the training data $\mathcal{D}_{\text{train}}$. For a fair comparison, the calibration dataset $\mathcal{D}_{\text{cal}}$ of post-hoc methods is included in the training dataset of regularization methods.

3.5.1 Quantile Recalibration

Recall that QR aims to transform a potentially miscalibrated CDF $\hat{F}_{Y|X}$ into a probabilistically calibrated CDF $\hat{F}'_{Y|X} = F_{\hat{U}} \circ \hat{F}_{Y|X}$, using the calibration map $F_{\hat{U}}$ which represents the CDF of the PITs for $\hat{F}_{Y|X}$. In practice, $F_{\hat{U}}$ needs to be estimated from data. Kuleshov et al. (2018) proposed estimating it using isotonic regression, while Utpala and Rai (2020) showed that computing the empirical CDF is an equivalent and simpler method. Specifically, given a set of PIT values $\hat{U}_i = \hat{F}_{Y|X=X^{(i)}}(Y^{(i)}), i = 1, \ldots, n$, the calibration map Φ_{EMP} is computed as:

$$\Phi_{\text{EMP}}(\alpha; \{\hat{U}_i\}_{i=1}^{n}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(\hat{U}_i \leqslant \alpha), \tag{3.3}$$

where $\alpha \in [0, 1]$.

Similarly to Utpala and Rai (2020), we also consider a linear and continuous calibration map Φ_{LIN}, which corresponds to a linear interpolation between the points

$$\{(0, 0), (\hat{U}_{(1)}, {}^{1}\!/_{n+1}), \ldots, (\hat{U}_{(n)}, {}^{n}\!/_{n+1}), (1, 1)\},$$

where $\hat{U}_{(k)}$ is the kth order statistic of $\hat{U}_1, \ldots, \hat{U}_n$.

In addition, we propose a calibration map based on kernel density estimation (KDE), denoted as Φ_{KDE}. The key idea is to use a relaxed approximation of the indicator function, which allows us to make the PIT CDF (3.3) differentiable. Specifically, we compute

$$\mathbb{1}_\tau(a \leqslant b) = \sigma(\tau(b - a)) \approx \mathbb{1}(a \leqslant b),$$

where $\tau > 0$ is a hyperparameter and $\sigma(x) = \frac{1}{1+e^{-x}}$ denotes the sigmoid function. The resulting smoothed empirical CDF is given by

$$\Phi_{\text{KDE}}(\alpha; \{\hat{U}_i\}_{i=1}^{n}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_\tau(\hat{U}_i \leqslant \alpha). \tag{3.4}$$

This corresponds to estimating the CDF $F_{\hat{U}}$ using KDE based on n realizations of $\hat{U}$ ($\{\hat{U}_i\}_{i=1}^{n}$). Since σ is the CDF of the logistic distribution, we use the PDF of the logistic distribution as the kernel in the KDE. Algorithm 5 summarizes this method.

Algorithm 5 Quantile recalibration

Input: Predictive CDF $\hat{F}_{Y|X}$ and $\mathcal{D}_{\text{cal}} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^{n}$.
Compute $\hat{U}_i = \hat{F}_{Y|X=X^{(i)}}(Y^{(i)}) \quad (i = 1, \ldots, n)$
Compute a calibration map Φ: either Φ_{EMP}, Φ_{LIN}, or Φ_{KDE}
Return: Recalibrated CDF $\hat{F}'_{Y|X} = \Phi \circ \hat{F}_{Y|X}$

3.5.2 Split Conformal Prediction

Let us assume the input-output pair (X, Y) and the data points in our calibration dataset $\mathcal{D}_{\text{cal}} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^{n}$ are drawn exchangeably from $P_{X,Y}$[2]. Given a predictive model h_θ, a miscoverage level $\alpha \in [0, 1]$ and an input $x \in \mathcal{X}$, Section 2.5 showed that SCP constructs a prediction set $\hat{R}(x) \subseteq \mathcal{Y}$ satisfying the property:

$$\mathbb{P}\left(Y \in \hat{R}(X)\right) = \frac{\lceil (n+1)(1-\alpha) \rceil}{n+1} \tag{3.5}$$

$$\approx 1 - \alpha. \tag{3.6}$$

This is achieved using a conformity score $s(x, y)$ with $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, which intuitively quantifies the similarity between new samples and previously observed samples. As in Section 2.5, we denote the $(\lceil (n+1)(1-\alpha) \rceil)$ empirical quantile of the scores $s(X^{(i)}, Y^{(i)}), \ldots, s(X^{(n)}, Y^{(n)})$ as $\hat{q}$. Let us assume that the conformity score increases with y, and let us denote this conformity score as $s(y \mid x) = s(x, y)$, with $s^{-1}(y \mid x)$ the inverse w.r.t. y. Then, an interval $\hat{R}(x) = (-\infty, s^{-1}(\hat{q} \mid x)]$ can be constructed, ensuring the conformal guarantee (3.5) at level $1 - \alpha$.

In the context of conformalizing quantiles, we adapt the conformity scores presented in Section 2.5.4 to ensure they are increasing in y. For example, a variant of CQR (Romano et al., 2019) uses $s(x, y) = y - \hat{Q}_{Y|X=x}(1 - \alpha)$, and DCP (Izbicki et al., 2020; Chernozhukov et al., 2021) uses $s(x, y) = \hat{F}_{Y|X=x}(y)$. Algorithm 6 provides a summary of the procedure to compute calibrated quantiles using SCP.

Algorithm 6 Calibrated quantiles with SCP

Input: Base predictor h_θ, $\mathcal{D}_{\text{cal}} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^{n}$, strictly increasing conformity score s, quantile level $1 - \alpha \in [0, 1]$, input x.
Compute $s^{(i)} = s(X^{(i)}, Y^{(i)}) \quad (i = 1, \ldots, n)$
Compute $\hat{q} = s_{(\lceil (n+1)(1-\alpha) \rceil)}$ where $s_{(k)}$ denotes the kth smallest value among $\{s^{(1)}, \ldots, s^{(n)}, +\infty\}$
Return: Calibrated quantile $s^{-1}(\hat{q} \mid x)$

3.5.3 Regularization-based Calibration

Regularization-based calibration methods aim to enhance model calibration by incorporating a regularization term into the training objective. Compared to classification, there are relatively fewer methods specifically designed for regression problems. In this section, we discuss two approaches: quantile regularization (Utpala and Rai, 2020) and the truncation method (Y. Chung et al., 2021). The main steps of regularization-based calibration are summarized in Algorithm 7.

[2]This is implied by the common i.i.d. assumption.

Algorithm 7 Regularization-based calibration

Input: Base predictor h_θ, $\mathcal{D}_{\text{train}} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^N$, calibration regularizer $\mathcal{R}(\theta)$ and tuning parameter $\lambda \geqslant 0$.
Define $\hat{\mathcal{R}}'(\theta; \mathcal{D}) = \frac{1}{N}\sum_{i=1}^N \mathcal{L}(h_\theta(X^{(i)}), Y^{(i)}) + \lambda \mathcal{R}(\theta; \mathcal{D})$
Compute $\hat{\theta}^* = \arg\min_{\theta \in \Theta} \hat{\mathcal{R}}'(\theta; \mathcal{D})$
Return: Regularized model $h_{\hat{\theta}*}$

Quantile Regularization

The regularizer proposed by Utpala and Rai (2020) aims to measure the deviation of the PIT variable $\hat{U}$ from a uniform distribution, which is characteristic of a probabilistically calibrated model. This regularization penalty encourages the selection of calibrated models during training.

The authors observed that the KL divergence between $\hat{U}$ and a standard uniform random variable is equivalent to the negative differential entropy of $\hat{U}$, denoted as $H(\hat{U})$. To approximate $H(\hat{U})$, they employed sample-spacing entropy estimation (Vasicek, 1976) over the set of PITs $\hat{U}_1, \ldots, \hat{U}_M$ with $\hat{U}_i = \hat{F}_{Y|X=X^{(i)}}(Y^{(i)})$. This results in the following regularizer:

$$\mathcal{R}_{\text{QREG}}(\theta; \mathcal{D}) = \frac{1}{N-k}\sum_{i=1}^{N-k} \log\left[\frac{N+1}{k}(\hat{U}_{(i+k)} - \hat{U}_{(i)})\right] \approx H(\hat{U}), \tag{3.7}$$

where k is a hyperparameter satisfying $1 \leqslant k \leqslant N$, and $\hat{U}_{(i)}$ represents the ith order statistic of $\hat{U}_1, \ldots, \hat{U}_N$.

To ensure differentiability during optimization, the authors employed a differentiable relaxation technique called NeuralSort (Grover et al., 2019), as sorting is a non-differentiable operation.

Truncation-based Calibration

The regularization approach introduced by Y. Chung et al. (2021), that we denote Trunc, involves truncating the predictive distribution based on the current level of calibration.

Given a quantile model $\hat{Q}_{Y|X}$, let $\hat{F}_{\hat{U}}(\alpha) = \frac{1}{N}\sum_{i=1}^N \mathbb{1}(Y^{(i)} \leqslant \hat{Q}_{Y|X=X^{(i)}}(\alpha))$ be the estimated PIT CDF evaluated at α and $\rho(x, y) = (y - x)\mathbb{1}(x < y)$. The regularization objective for level α is defined as follows:

$$\mathcal{R}_{\text{Trunc}}(\theta; \mathcal{D}, \alpha) \tag{3.8}$$

$$= \begin{cases} \frac{1}{N}\sum_{i=1}^N \rho(\hat{Q}_{Y|X=X^{(i)}}(\alpha), Y^{(i)}) & \text{if } \hat{F}_{\hat{U}}(\alpha) < \alpha, \\ \frac{1}{N}\sum_{i=1}^N \rho(Y^{(i)}, \hat{Q}_{Y|X=X^{(i)}}(\alpha)) & \text{otherwise.} \end{cases} \tag{3.9}$$

This regularization objective adjusts $\hat{F}_{\hat{U}}(\alpha)$ to match α by increasing it when $\hat{F}_{\hat{U}}(\alpha) < \alpha$, and vice versa. The final regularization objective is computed by averaging $\mathcal{R}_{\text{Trunc}}(\theta; \mathcal{D}, \alpha)$ over equidistant quantile levels $\alpha_1 < \cdots < \alpha_M$:

$$\mathcal{R}_{\text{Trunc}}(\theta; \mathcal{D}) = \frac{1}{M}\sum_{j=1}^M \mathcal{R}_{\text{trunc}}(\theta; \mathcal{D}, \alpha_j). \tag{3.10}$$

Y. Chung et al. (2021) combine the previous regularization objective with a sharpness objective that penalizes the width between the quantile predictions, given by $\frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N}|\hat{Q}_{Y|X=X^{(i)}}(\alpha_j) - \hat{Q}_{Y|X=X^{(i)}}(1-\alpha_j)|$. Instead, we combine it with a strictly proper scoring rule.

3.5.4 New Regularization-based Calibration Methods

Building upon the quantile calibration method discussed in Section 3.5.3, we propose two new regularization objectives which compute a differentiable PCE_p using alternative statistical distances.

The first approach, named PCE-KDE, leverages the differentiable calibration map Φ_{KDE} (3.4) based on KDE. Given a set of quantile levels $\{\alpha_j\}_{j=1}^{M}$, the regularization objective is given by

$$\mathcal{R}_{\text{PCE-KDE}}(\theta;\mathcal{D}) = \frac{1}{M}\sum_{j=1}^{M}\left|\alpha_j - \Phi_{\text{KDE}}(\alpha_j;\{\hat{U}_i\}_{i=1}^{N})\right|^p, \tag{3.11}$$

where $p > 0$. Note that $\mathcal{R}_{\text{PCE-KDE}}$ converges to PCE_p in (3.2) as τ in (3.4) goes to $+\infty$.

The second approach considers distances of the form $\int_0^1 |Q_{\hat{U}}(\alpha) - Q_U(\alpha)|^p d\alpha$, where $Q_{\hat{U}}$ and Q_U denote the quantile functions of the true and uniform distributions, respectively. When $p = 1$, this distance reduces to the 1-Wasserstein distance, equivalent to $\int_0^1 |F_{\hat{U}}(\alpha) - F_U(\alpha)|d\alpha$, which aligns with PCE (see Proposition 6 in Section C.1.1).

It is well known that the ith order statistic of N i.i.d. standard uniform samples follows a Beta distribution with mean $i/N+1$. By exploiting the fact that $\mathbb{E}\left[F_{\hat{U}}(\hat{U}_{(i)})\right] = i/N+1$, we approximate $Q_{\hat{U}}(i/N+1)$ using the i-th order statistic $\hat{U}_{(i)}$. The regularization objective is given by

$$\mathcal{R}_{\text{PCE-Sort}}(\theta;\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{U}_{(i)} - \frac{i}{N+1}\right|^p, \tag{3.12}$$

where, again, $p > 0$. Differentiable relaxations to sorting, such as those proposed by Blondel et al. (2020) and Cuturi et al. (2019), can be employed to obtain the order statistics.

3.6. A Comparative Study of Probabilistic Calibration Methods

In continuation of the empirical study described in Section 3.4, we now proceed to evaluate the performance of the probabilistic calibration methods outlined in the previous section. Specifically, we apply eight distinct calibration methods to the three neural regression models introduced in Section 3.4. These methods are divided into two categories: post-hoc methods and regularization-based methods.

To assess the effectiveness of these calibration methods, we employ four different evaluation metrics. The evaluation is conducted on a set of 57 datasets, utilizing the same experimental setup detailed in Section 3.4. To ensure a fair and consistent comparison, all the methods have been implemented within a unified codebase[3].

[3] `https://github.com/Vekteur/probabilistic-calibration-study`

3.6.1 Experimental Setup

Base probabilistic models and calibration methods. We consider the three probabilistic models presented in Section 3.4, namely `MIX-NLL`, `MIX-CRPS`, and `SQR-CRPS`. For the `MIX` models, when applying QR, we transform the CDF using the empirical CDF estimator (`QR-EMP`), the linear estimator (`QR-LIN`), or the KDE estimator (`QR-KDE`). For `SQR-CRPS`, we transform multiple quantiles using conformalized quantile regression (`CQR`). For the three models, we consider the four regularization objectives presented in Sections 3.5.3 and 3.5.4 (with $p = 1$), namely $\mathcal{R}_{\text{PCE-KDE}}$ (`PCE-KDE`), $\mathcal{R}_{\text{PCE-Sort}}$ (`PCE-Sort`), $\mathcal{R}_{\text{QREG}}$ (`QREG`), and $\mathcal{R}_{\text{Trunc}}$ (`Trunc`). `PCE-Sort` is only shown in the Appendix because it performs similarly to `PCE-KDE`. To ensure a fair comparison, methods that are not post-hoc use the calibration dataset as additional training data.

Metrics. We measure the accuracy of the probabilistic predictions using NLL and CRPS. For the SQR model, we estimate CRPS by averaging the quantile score at 64 equidistant quantile levels. Probabilistic calibration is measured using PCE, defined in (3.2). Finally, we measure sharpness using the mean standard deviation of the predictive distributions, denoted by STD.

Hyperparameters. In our experiments, `MIX-NLL` and `MIX-CRPS` output a mixture of 3 Gaussians, and `SQR-CRPS` outputs 64 quantiles. We justify the choice of these hyperparameters in Section C.3. The hyperparameter τ of `QR-KDE` and `PCE-KDE` is fixed at 100, which was found to perform well empirically. For regularization methods, an important hyperparameter is the regularization factor λ. As previously observed in classification (Karandikar et al., 2021), we found that higher values of λ tend to improve calibration but worsen NLL, CRPS, and STD. Karandikar et al. (2021) proposed to limit the loss in accuracy by a maximum of 1%. We adopt a similar strategy by selecting λ which minimizes PCE with a maximum increase in CRPS of 10% in the validation set. For each dataset, we select λ in the set $\{0, 0.01, 0.05, 0.2, 1, 5\}$, which corresponds to various degrees of calibration regularization.

Comparison of multiple models over many datasets. Since the NLL, CRPS and STD have different scales across datasets, direct comparison of their absolute values is misleading. To address this, we follow Karandikar et al. (2021) and report *Cohen's d*, a standardized effect size (over $\kappa = 5$ independent runs, in our case). For a given dataset, Cohen's d quantifies the difference between the sample mean value of a method ($\tilde{\mu}_{\text{method}}$) and a baseline ($\tilde{\mu}_{\text{baseline}}$), standardized by their pooled standard deviation. With unbiased sample variances $\tilde{\sigma}^2_{\text{method}}$ and $\tilde{\sigma}^2_{\text{baseline}}$, it is defined as

$$\frac{\tilde{\mu}_{\text{method}} - \tilde{\mu}_{\text{baseline}}}{\sqrt{\frac{\tilde{\sigma}^2_{\text{method}} + \tilde{\sigma}^2_{\text{baseline}}}{2}}}.$$

This metric quantifies the magnitude of the performance difference in units of standard deviation, making results comparable across datasets. A negative value indicates an improvement over the baseline, and in Karandikar et al. (2021) values of -0.8 and -2 are considered large and huge improvements, respectively.

Due to the heterogeneity of the datasets that we consider, the performance of our models can vary widely across datasets. To visualize the results, we show the distribution of Cohen's d using letter-value plots (Hofmann et al., 2011), which indicate the quantiles at levels $1/8$, $1/4$, $1/2$, $3/4$

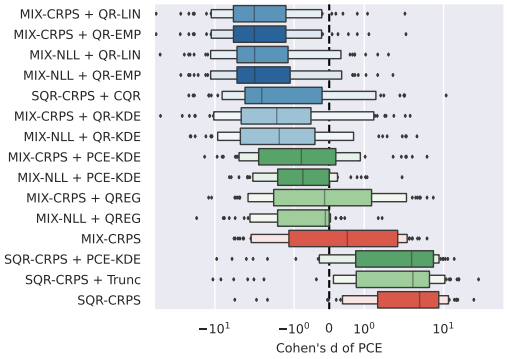
and 7/8, as well as outliers. A median value below zero indicates that the model improved the metric on more than half the datasets.

In order to assess whether significant differences exist between different methods, we follow the recommendations of Ismail Fawaz et al. (2019), which are based on Demšar (2006). First, we test for a significant difference among model performances using the Friedman test (Friedman, 1940). Then, we use the pairwise post-hoc analysis recommended by Benavoli et al. (2016), performing Wilcoxon signed-rank tests (Wilcoxon, 1945) with Holm–Bonferroni correction (Holm, 1979). The results of this procedure are shown using a *critical difference diagram* (Demšar, 2006). The lower the rank (further to the right), the better the performance. A thick horizontal line shows a group of models whose performance is not significantly different, with a significance level of 0.05.

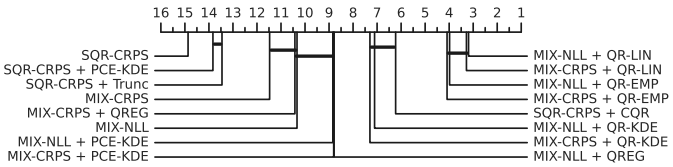
3.6.2 Results

Figure 3.3 shows the letter-values plots for the Cohen's d of PCE (left panel) as well as the associated critical diagram (right panel), for all methods and datasets. The reference model is `MIX-NLL`. The results with other models as reference are available in Section C.2.1. Blue, green, and red colors are used for the post-hoc methods, the regularization-based methods, and the base predictors, respectively. The same information is given in Figures 3.4 and 3.5 for the CRPS and the NLL, respectively.

Comparison of PCE. As expected, Figure 3.3 shows that the PCE of calibration methods is improved compared to the base predictors. Furthermore, independently of the base predictor, we can see that post-hoc methods achieve significantly better PCE than regularization methods. When comparing `PCE-KDE` with `QREG`, we can see that there is a significantly larger decrease in PCE with the `MIX-CRPS` base predictor compared to `MIX-NLL`. Finally, both `PCE-KDE` and `Trunc` decrease PCE for `SQR-CRPS`, without a significant difference between them.



(a) Cohen's d of PCE w.r.t. the `MIX-NLL` model

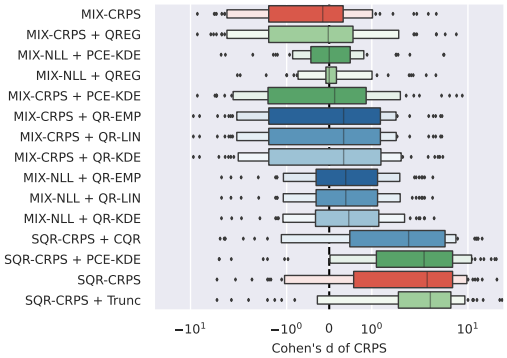


(b) Critical difference diagram

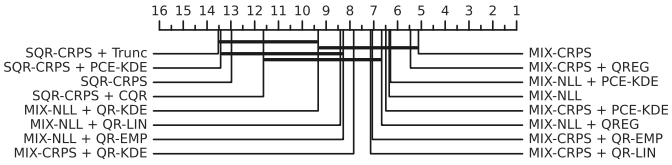
Figure 3.3: Comparison of PCE with multiple base losses and calibration methods.

Comparison of CRPS. While post-hoc methods outperform regularization methods in terms of PCE, Figure 3.4 shows they have a higher CRPS (except for the `SQR` base predictor). This

can be explained by the fact that regularization methods prevent the CRPS from increasing exceedingly due to the selection criterion for λ.



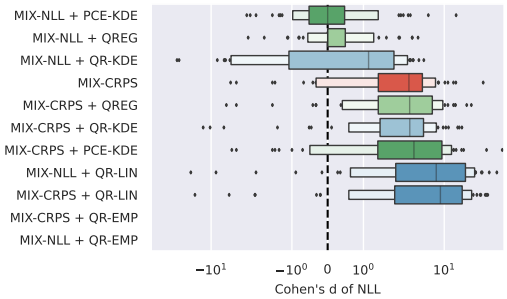
(a) Cohen's d of CRPS w.r.t. the `MIX-NLL` model



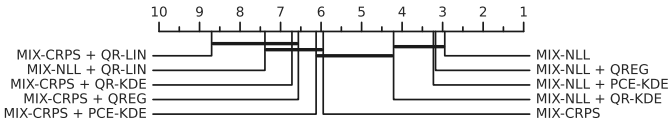
(b) Critical difference diagram

Figure 3.4: Comparison of CRPS with multiple base losses and calibration methods.

Comparison of NLL. Figure 3.5 shows the importance of the calibration map. In fact, quantile recalibration with a linear map significantly increases the NLL, while smooth interpolation decreases PCE without a large increase in NLL. Note that we only consider `MIX` models since we cannot compute the NLL for `SQR`.



(a) Cohen's d of NLL w.r.t. the `MIX-NLL` model



(b) Critical difference diagram

Figure 3.5: Comparison of NLL with multiple base losses and calibration methods.

On the choice of a calibration method. If probabilistic calibration is critical to the application, our experiments suggest that post-hoc methods such as quantile recalibration and conformal prediction should be preferred. However, when we also want to control the CRPS or the NLL, regularization methods can offer a better trade-off in terms of calibration and sharpness. In fact, as shown in Figure C.1 in Section C.2.1, when the base predictor is `MIX-NLL`, all regularization methods provide a significant improvement in probabilistic calibration without deteriorating the CRPS, NLL or STD. For the `MIX-CRPS` model, Figure C.2 shows that `QREG` has limited impact on CRPS and NLL, while providing better calibration. For

the `SQR-CRPS` base predictor, Figure C.3 shows that the `SQR-CRPS + CQR` conformal method significantly outperforms the `Trunc` and `PCE-KDE` regularization methods both in terms of PCE and CRPS. Overall, Section C.2.1 suggests that `MIX-NLL + PCE-KDE`, `MIX-CRPS + QREG` and `SQR-CRPS + CQR` are good choices for practitioners aiming to improve PCE without significantly impacting other aspects of the conditional distribution. Finally, since both regularization and post-hoc methods are able to improve calibration, we investigate whether a combination of these two methods can lead to better performance. Figure C.4 in Section C.2.2 shows that such a combination does not significantly improve probabilistic calibration, with an increase in CRPS and NLL. This indicates that practitioners should exercise caution when applying regularization to a model that is already well-calibrated.

3.6.3 Link between Quantile Recalibration and Conformal Prediction

Conformal prediction methods are well-known for their finite-sample coverage guarantee. Interestingly, a specific implementation of QR can be considered a special case of conformal prediction. This implies that QR can also provide a finite-sample coverage guarantee. This observation could potentially explain why both methods, conformal prediction and QR, are effective in improving probabilistic calibration.

Theorem 6. Quantile recalibration is equivalent to distributional conformal prediction (DCP) of left intervals at each coverage level $\alpha \in [0, 1]$. The equivalence is obtained when the estimator of the calibration map is defined by a slightly different estimator than the conventional one in (3.3), namely $\Phi_{\text{DCP}}(\alpha) = \frac{1}{n+1} \sum_{i=1}^{n} \mathbb{1}(\hat{U}_i \leqslant \alpha)$.

Proof. Given a predictive distribution with CDF $\hat{F}_{Y|X}$ learned from a training dataset $\mathcal{D} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^{N}$ where $(X^{(i)}, Y^{(i)}) \overset{\text{i.i.d.}}{\sim} P_{X,Y}$, let $\hat{U}_i = \hat{F}_{Y|X=X^{(i)}}(Y^{(i)})$ represent the PIT values computed on a separate calibration dataset $\mathcal{D}_{\text{cal}} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^{n}$, where the data points are also i.i.d. from $P_{X,Y}$.

In the DCP approach, as outlined in Algorithm 6, the conformal scores are given by the PIT values $\hat{U}_i$. DCP first computes the α empirical quantile of the scores as $\hat{q} = \hat{U}_{(\lceil (n+1)\alpha \rceil)}$, where $\hat{U}_{(k)}$ represents the kth smallest value among $\{\hat{U}_1, \ldots, \hat{U}_n, +\infty\}$. Then, the conformalized quantile is computed as $\hat{Q}'_{Y|X=x}(\alpha) = \hat{Q}_{Y|X=x}(\hat{q})$, which corresponds to conformal prediction with coverage α for the left interval $(-\infty, \hat{Q}'_{Y|X=x}(\alpha)]$.

Let us consider QR with the calibration map Φ in Algorithm 5 given by $\Phi_{\text{DCP}}(\alpha) = \frac{1}{n+1} \sum_{i=1}^{n} \mathbb{1}(\hat{U}_i \leqslant \alpha)$. It computes a recalibrated CDF $\hat{F}'_{Y|X}$ by composing the original CDF $\hat{F}_{Y|X}$ with Φ_{DCP}, yielding $\hat{F}'_{Y|X=x}(y) = \Phi_{\text{DCP}}(\hat{F}_{Y|X=x}(y))$.

We observe that Φ_{DCP} is the CDF of a discrete random variable, with $\Phi^{-1}_{\text{DCP}}(\alpha) = \hat{U}_{(\lceil (n+1)\alpha \rceil)}$ representing its empirical quantile function. Furthermore, the composition $\Phi_{\text{DCP}} \circ \hat{F}_{Y|X=x}$ acts as the inverse function of $\hat{Q}_{Y|X=x} \circ \Phi^{-1}_{\text{DCP}}$. As a result, both the DCP approach and QR yield QFs and CDFs that correspond to the same underlying distribution. □

QR with other calibration maps (e.g., Φ_{EMP}, Φ_{LIN}, or Φ_{KDE}) would correspond to DCP where the empirical quantile $\hat{q}$ is selected using other strategies, which does not provide the exact

conformal guarantee (3.5).

3.7. Conclusion

The observation that neural network classifiers tend to be miscalibrated (Guo et al., 2017) has prompted the development of various approaches for calibrating these models. In this paper, we present the largest empirical study conducted to date on the probabilistic calibration of neural regression models. Our study provides valuable insights into their performance and the selection of calibration methods. Notably, we introduce a novel differentiable calibration map based on kernel density estimation for quantile recalibration, as well as two novel regularization objectives derived from the PCE.

Our study reveals that regularization methods can provide a favorable trade-off between calibration and predictive accuracy. However, post-hoc methods demonstrate superior performance in terms of PCE. We attribute this finding to the finite-sample coverage guarantee offered by CP and demonstrate that quantile recalibration can be viewed as a specific case of CP. In Chapter 4, we design a method that achieves the superior PCE of post-hoc methods while also improving predictive accuracy.

Limitations.

A first limitation of this work is that the study is limited to tabular datasets and simple neural architectures. Future works could study probabilistic regression in the context of other data modalities such as images or text. For example, El Nahhas et al. (2024) consider a single-output regression problem where the input consists of high-resolution digitized tissue images and the output is a continuous biomarker value.

A second limitation is that the study is limited to probabilistic calibration. Future studies may explore alternative notions of calibration (Gneiting and Resin, 2023). Notably, auto-calibration is a potential direction which inspired several calibration methods (H. Song et al., 2019; Popordanoska et al., 2022; Kuleshov and Deshpande, 2022).

A third limitation is that we only consider calibration on univariate output spaces, which excludes multi-output problems. We consider multi-output approaches in Chapters 5 to 7.

CHAPTER 4

Probabilistic Calibration by Design

This chapter is based on the following paper:

> **Victor Dheur** and Souhaib Ben Taieb. Probabilistic Calibration by Design for Neural Network Regression (2024). *The 27th International Conference on Artificial Intelligence and Statistics.*

4.1. Introduction

The preceding chapter established an empirical baseline for uncertainty quantification in single-output neural network regression. Through a large-scale study, we compared the main families of methods for improving probabilistic calibration, a foundational concept introduced in Section 2.4. A key finding from our analysis in Chapter 3 was the consistent ability of quantile recalibration (QR) in achieving lower PCE compared to training-time regularization techniques. We provided a theoretical justification for this observation by demonstrating that QR is a special case of split conformal prediction (SCP), thereby benefiting from its finite-sample guarantees (Theorem 6).

However, this effectiveness comes with a conceptual drawback: the two-stage nature of the process. The base predictor is first trained to optimize a strictly proper scoring rule such as the NLL, resulting in informative but potentially miscalibrated probabilistic predictions. Then, post-hoc recalibration operates independently of the model's training. This separation is suboptimal, as the recalibration step can potentially result in lower predictive accuracy. This raises a central question: can we bridge the gap between the strong performance of post-hoc methods and the integrated nature of end-to-end learning?

This chapter, based on our paper Dheur and Ben Taieb (2024), introduces a novel neural network training procedure called *quantile recalibration training* (QRT) that seamlessly integrates post-hoc calibration into the training process, resulting in an end-to-end approach. Our method leverages the principle of minimizing the sharpness of predictions while ensuring calibration (Gneiting et al., 2007). The method uses the differentiable KDE-based calibration map developed in Section 3.5.1 to allow gradient-based optimization. By minimizing the NLL, our approach directly incentivizes

informative predictions, while simultaneously ensuring calibration at each training step, either on the minibatch or on a dedicated calibration dataset.

QRT stands apart from other regularization methods by offering improvements in both the NLL and probabilistic calibration of the final model. Our approach aligns with the recommendation made by D.-B. Wang et al. (2021) to view model training and post-hoc calibration as an integrated framework rather than treating them as separate steps. Related works in classification (Stutz et al., 2022; Einbinder et al., 2022) proposed integrating SCP into neural network training, resulting in finite-sample coverage with smaller prediction sets after post-hoc SCP. The code base has been used for the implementation of all methods to ensure a fair comparison.

We make the following main contributions:

1. We propose a novel training procedure to learn predictive distributions that are probabilistically calibrated at every training step, called QRT (see Section 4.3). We also propose an algorithm which unifies QRT with quantile recalibration, Quantile Regularization and standard NLL minimization.

2. We demonstrate the effectiveness of our method in a large-scale experiment based on 57 tabular datasets. The results show improved NLL on the test set while at the same time ensuring calibration (see Section 4.4).

3. We provide an in-depth analysis of the impact of the base predictor and different hyperparameters on predictive accuracy and calibration. We also conduct an ablation study to evaluate the significance of the different components of our proposed method (see Section 4.5).

4.2. Background

Quantile Recalibration Recall that *quantile recalibration* (QR, Kuleshov et al. (2018)) computes a probabilistically calibrated CDF $\hat{F}'_{Y|X} = F_{\hat{U}} \circ \hat{F}_{Y|X}$, where $\hat{U} = \hat{F}_{Y|X}(Y)$ and $F_{\hat{U}}$ is estimated from data.

The estimator of $F_{\hat{U}}$, called a calibration map, can be the empirical CDF Φ_{EMP} computed from the PITs $\hat{U}_i = \hat{F}_{Y|X=X^{(i)}}(Y^{(i)})$ of a separate i.i.d. calibration dataset $\mathcal{D}_{\text{cal}} = \{(X^{(i)}, Y^{(i)})\}_{i=1}^n$. Since Φ_{EMP} is not differentiable, the resulting calibrated CDF $\hat{F}'_{Y|X}$ is not differentiable either. As introduced in Chapter 3, Dheur and Ben Taieb (2023) proposed to compute a differentiable calibration map

$$\Phi_{\text{KDE}}(\alpha) = \frac{1}{n} \sum_{i=1}^{n} F_{\text{Log}}(\alpha; \hat{U}_i, b^2 n^{-2/5}), \tag{4.1}$$

based on kernel density estimation (KDE). This corresponds to a mixture of logistic CDFs F_{Log} with means $\hat{U}_1, \dots, \hat{U}_n$. Compared to Dheur and Ben Taieb (2023), to obtain more consistent hyperparameters across datasets, we compute the variance $b^2 n^{-2/5}$ following Scott's rule (Scott, 1992). The bandwidth $b > 0$ is a hyperparameter controlling the smoothness of the calibration map. Note that Φ_{KDE} converges to Φ_{EMP} as $b \to 0$.

Furthermore, Dheur and Ben Taieb (2023) showed that QR provides a finite-sample guarantee

with a specific calibration map, namely:

$$\mathbb{P}\left(\Phi_{\mathrm{DCP}}(\hat{F}_{Y|X}(Y)) \leqslant \alpha\right) = \frac{\lceil (n+1)\alpha \rceil}{n+1} \approx \alpha, \tag{4.2}$$

where $\Phi_{\mathrm{DCP}}(\alpha) = \frac{1}{n+1}\sum_{i=1}^{n} \mathbb{1}(\hat{U}_i \leqslant \alpha)$ is a calibration map derived from distributional conformal prediction (Chernozhukov et al., 2021; Izbicki et al., 2020). This property is approximately obtained by other calibration maps such as Φ_{EMP} and Φ_{KDE}. We note that the probability in (4.2) is also taken over the calibration dataset $\mathcal{D}_{\mathrm{cal}}$.

Quantile Regularization Recall that *quantile regularization* (QREG, Utpala and Rai (2020)) minimizes a loss function of the form

$$-\frac{1}{N}\sum_{i=1}^{N} \log \hat{f}_{Y|X=X^{(i)}}(Y^{(i)}) + \lambda \mathcal{R}_{\mathrm{QREG}}(\theta), \tag{4.3}$$

where the first term is the NLL and $\lambda > 0$ is a regularization hyperparameter. The regularization function $\mathcal{R}_{\mathrm{QREG}}(\theta)$ encourages calibration by minimizing the KL divergence between $\hat{U}$ and a uniformly distributed random variable U. This reduces to maximizing the differential entropy $H(\hat{U})$ of $\hat{U}$ since $D_{\mathrm{KL}}\left(\hat{U} \,\middle\|\, U\right) = -H(\hat{U})$.

We note that this approach should be distinguished from regularizers in classification (Pereyra et al., 2017) that maximize the entropy of the target Y and not the differential entropy of the PIT $\hat{U}$.

4.3. Quantile Recalibration Training

We introduce *quantile recalibration training* (QRT), a novel method for training neural network regression models. Predictive distributions are iteratively recalibrated during model training and are hence calibrated by design.

4.3.1 The QRT learning procedure

Recall that QR involves training $\hat{F}_{Y|X}$ by minimizing the NLL and then adjusting it by producing a revised predictive distribution $\hat{F}'_{Y|X} = \Phi_{\mathrm{KDE}} \circ \hat{F}_{Y|X}$. Given that the calibration map Φ_{KDE} is differentiable, our QRT procedure integrates it end-to-end into the optimization procedure. Specifically, we directly minimize the NLL of $\hat{F}'_{Y|X}$ which involves iteratively recalibrating it during training. Using the chain rule, the NLL of $\hat{F}'_{Y|X}$ can be conveniently decomposed as follows:

$$\begin{aligned}
&\textstyle\sum_{i=1}^{N} - \log \hat{f}'_{Y|X=X^{(i)}}(Y^{(i)}) && (4.4)\\
&= \textstyle\sum_{i=1}^{N} - \log \hat{f}_{Y|X=X^{(i)}}(Y^{(i)}) - \log \hat{f}_{\hat{U}}(\hat{F}_{Y|X=X^{(i)}}(Y^{(i)})) && (4.5)\\
&= \textstyle\sum_{i=1}^{N} - \log \hat{f}_{Y|X=X^{(i)}}(Y^{(i)}) + \hat{H}(\hat{U}). && (4.6)
\end{aligned}$$

The first term in (4.6) is the NLL of the base predictor $\hat{F}_{Y|X}$ and $\hat{H}(\hat{U})$ can be interpreted as the differential entropy of $\hat{U}$.

Interestingly, the second term $\hat{H}(\hat{U})$ corresponds to the opposite of the regularization term of QREG (3.7). This observation could seem counter-intuitive since it implies that, when training QRT, the PCE of $\hat{F}_{Y|X}$ will be maximized by the second term $\hat{H}(\hat{U})$ in the decomposition. However, QRT is valid since it produces $\hat{F}'_{Y|X}$ by minimizing the NLL of $\hat{F}'_{Y|X}$, which is a strictly proper scoring rule.

To compute the second term in (4.5), we estimate $\hat{f}_{\hat{U}}$ using the derivative of the calibration map Φ_{KDE}, which has a closed-form expression given by

$$\phi_{\text{KDE}}(\alpha) = \frac{\partial \Phi_{\text{KDE}}(\alpha)}{\partial \alpha} = \frac{1}{N} \sum_{i=1}^{N} f_{\text{Log}}(\alpha; \hat{U}^{(i)}, b^2 N^{-2/5}), \tag{4.7}$$

where f_{Log} is the PDF of a logistic distribution, as described in Section 4.2.

During training, ϕ_{KDE} is computed on the current batch and $\hat{F}'_{Y|X}$ is thus, by design, calibrated on the current batch. However, it does not satisfy the calibration guarantee (4.2) since the current batch has been observed during training. Hence, as a final step, we perform QR on a separate calibration dataset to obtain the calibration guarantee. We give more details in Section 4.3.3.

Finally, to account for the finite domain $[0, 1]$ of the PIT $\hat{U}$, we perform a slight adjustment to the calibration map Φ_{KDE}. The standard approach is to truncate the distribution by redistributing the density that has been estimated outside of $[0, 1]$ uniformly in $[0, 1]$. Instead, Blasiok and Nakkiran (2023) propose to redistribute the density slightly outside of $[0, 1]$ near 0 and 1, assuming that $\phi_{\text{KDE}}(x) = 0$ for $x \notin [-1, 2]$. The resulting calibration map Φ_{REFL} has the following derivative:

$$\phi_{\text{REFL}}(x) = \begin{cases} \phi_{\text{KDE}}(x) + \phi_{\text{KDE}}(-x) + \phi_{\text{KDE}}(2 - x) & \text{if } x \in [0, 1] \\ 0 & \text{if } x \notin [0, 1]. \end{cases} \tag{4.8}$$

This approach avoids an ill-defined calibration map and often leads to improved NLL. More motivation and details, including the definition of the corresponding CDF Φ_{REFL}, are given in Section D.3.

4.3.2 Illustrative example

Figure 4.1 illustrates the decomposition (4.6), where the NLL of $\hat{F}'_{Y|X}$ (first column) is equal to the sum of the NLL of the base predictor $\hat{F}_{Y|X}$ (second column) and the differential entropy of the PIT (third column). To allow a comparison between `QRT` and `BASE`, we alter the decomposition by introducing a coefficient β to the second term. When $\beta = 1$, we obtain the exact decomposition of `QRT`. When $\beta = 0$, the first and second column are equal and correspond to the loss of `BASE`. Metrics on this figure are computed on the validation dataset and metrics computed on the training dataset are available in Section D.1.8. The vertical bars correspond to the epoch selected by early stopping while the horizontal bars correspond to the value of the metric at the epoch selected by early stopping, on average over 5 runs. The stars indicate the methods `QRC` and `QRTC`, corresponding to `BASE` and `QRT`, respectively, after QR on a separate calibration dataset.

We can see that `QRT` achieves a lower validation NLL, suggesting improved probabilistic predictions, even though the NLL of $\hat{F}_{Y|X}$ is higher, which means that `QRT` relies on the calibration map

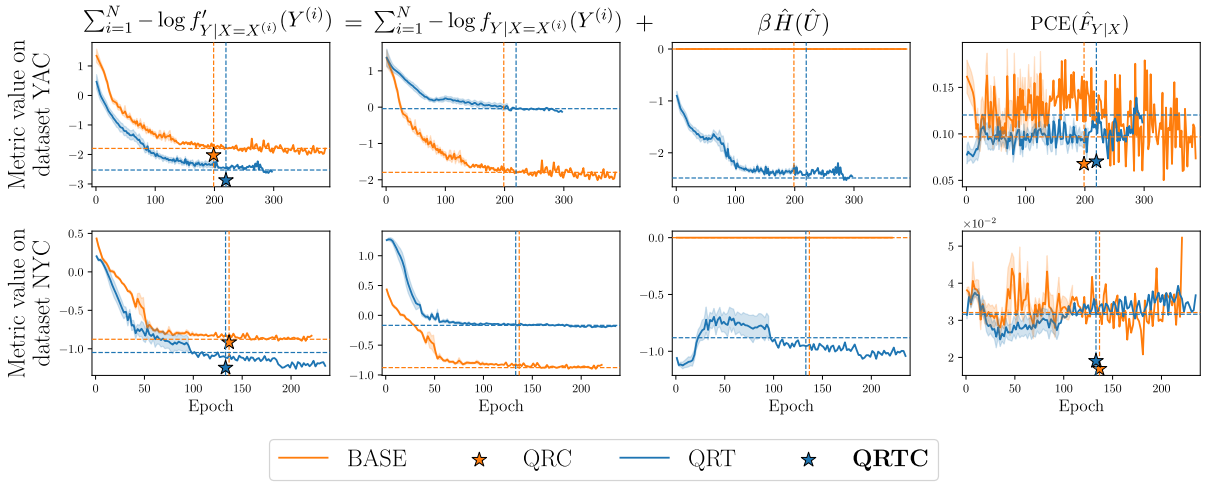


Figure 4.1: Comparison of QRT and BASE according to different metrics computed on the validation dataset. The three first columns show the decomposition of the NLL of QRT, where $\beta = 1$ for QRT and $\beta = 0$ for BASE. Each row represents one dataset and each column one metric. The training curves are averaged over 5 runs and the shaded area corresponds to one standard error. The vertical bars represent the epoch that was selected by early stopping (the one that minimizes the validation NLL), averaged over the 5 runs. The horizontal bars represent the value of the metric at the selected epoch, averaged over the 5 runs.

to achieve a lower NLL. We note that the calibration map does not introduce any additional parameters. In terms of calibration, we can see that the PCE of `BASE` has a higher variance across the epochs compared to `QRT`. The PCE of `QRT` is more stable during training and often lower. By constraining the model to be calibrated on a specific dataset at each training step, `QRT` involves a form of regularization which is fundamentally different from QREG.

The stars indicate that, after the post-hoc step, `QRTC` still benefits from improved NLL compared to `QRC`, and the PCE is improved in both cases due to the finite-sample guarantee provided by QR. These metrics reported on the 57 datasets that we consider in Section 4.4 are available in Section D.1.8, where we obtain similar observations on most datasets despite their heterogeneity.

4.3.3 A Unified Algorithm

Algorithm 8 unifies QRT, QREG and BASE, with or without QR, where the methods only differ by the hyperparameters β and C, as indicated in Table 4.1. The hyperparameter β, introduced in Section 4.3.2, is a coefficient for the second term of the decomposition (4.6). A value of $\beta = 1$ corresponds to QRT and $\beta = 0$ corresponds to NLL minimization without QRT. Tuning the hyperparameter β in order to minimize $\mathrm{PCE}(\hat{F}_{Y|X})$ corresponds to QREG with regularization strength $\lambda = -\beta$. The hyperparameter C controls whether the final model is recalibrated on a separate calibration dataset using QR.

Algorithm 8 QRT framework

Input: Predictive CDF $\hat{F}_{Y|X}$ constructed from a neural network model h_θ with parameters $\theta \in \Theta$, regularization strength $\beta \in \mathbb{R}$, boolean C, training dataset $\mathcal{D}_{\text{train}}$, calibration dataset $\mathcal{D}_{\text{cal}}$.
for each minibatch $\{(X^{(i)}, Y^{(i)})\}_{i=1}^{B} \subseteq \mathcal{D}_{\text{train}}$, until early stopping **do**
  $\hat{U}^{(i)} \leftarrow \hat{F}_{Y|X=X^{(i)}}(Y^{(i)})$ for $i = 1, \ldots, B$.
  Define ϕ_{REFL} from $\hat{U}^{(1)}, \ldots, \hat{U}^{(B)}$ using (4.8).
  $$\mathcal{L}(\theta) \;=\; -\frac{1}{B} \sum_{i=1}^{B} \underbrace{\left[\log \hat{f}_{Y|X=X^{(i)}}(Y^{(i)}) \;+\; \beta \, \log \phi_{\text{REFL}}(\hat{U}^{(i)}) \right]}_{\log \hat{f}'_{Y|X=X^{(i)}}(Y^{(i)})}.$$
  Update parameters θ using $\nabla_\theta \mathcal{L}(\theta)$.
end for
if C is True **then**
  $\hat{U}_i \leftarrow \hat{F}_{Y|X=X^{(i)}}(Y^{(i)})$ for each $(X^{(i)}, Y^{(i)}) \in \mathcal{D}_{\text{cal}}$.
  Define ϕ_{REFL} from the resulting PITs $\{\hat{U}_i\}_{i=1}^{n}$.
  return the predictive CDF $\Phi_{\text{REFL}} \circ \hat{F}_{Y|X}$.
else
  return the predictive CDF $\hat{F}_{Y|X}$.
end if

In Algorithm 8, the calibration map Φ_{KDE} is computed at each step on the current batch, allowing QRT to simultaneously use the neural network outputs to compute the first term and the second term of the decomposition (4.6). In Section D.2.2, we investigate the impact of computing the calibration map from data sampled randomly in the training dataset, which allows to compute the calibration map on a larger dataset. We observe that the approach in Algorithm 8 provides similar NLL than the approach in Section D.2.2, while being more computationally efficient. In Section D.1.3, we confirm that $\beta = 1$ provides the best NLL compared to other values of β.

Table 4.1: Summary of the compared methods, which differ only by the hyperparameters β and C in Algorithm 8.

Method	`BASE`	`QRC`	`QREG`	`QREGC`	`QRT`	`QRTC`
β	0	0	Tuned	Tuned	1	1
C	False	True	False	True	False	True

4.3.4 Time complexity

The proposed method can introduce increased computational demand due to evaluating $\log \phi_{\text{KDE}}(\hat{U}^{(i)})$, which results in $O(B^2)$ evaluations of f_{Log} per minibatch, where B is the batch size ($B = 512$ in our experiments). More precisely, $3B^2$ evaluations of f_{Log} are performed due to using the estimator Φ_{REFL} (see Section D.3). This additional computational demand does not depend on the size of the underlying neural network and hence becomes less significant when training highly computationally demanding models. In practice, we observe the time per minibatch to be nearly

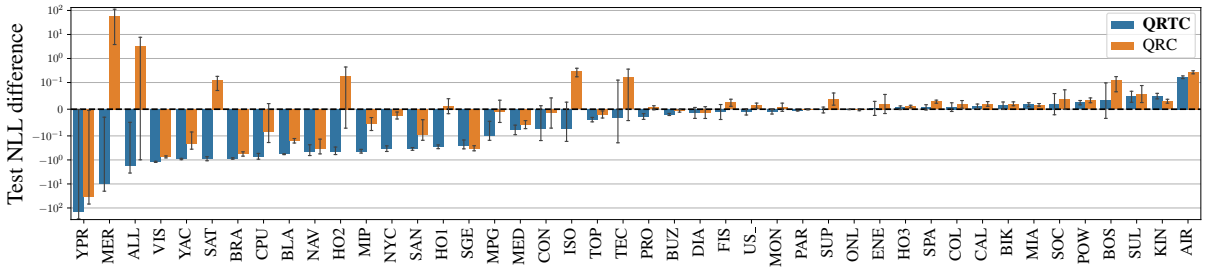


Figure 4.2: Difference in test NLL between two post-hoc methods (`QRTC` and `QRC`) and `BASE`, where negative values indicate an improvement compared to `BASE`, averaged over 5 runs with error bars corresponding to one standard error. We observe that `QRTC` achieves a lower NLL than `BASE` and `QRC` on most datasets. Note that, for `BASE`, $\hat{F}_{Y|X}$ is trained with a larger dataset that includes the calibration data of `QRTC` and `QRC`. The experimental setup is described in Section 4.4.

two-fold compared to a method without QR, as detailed in Section D.1.7.

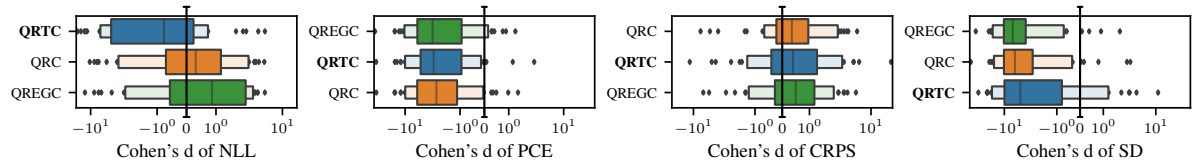
4.4. A Large-Scale Experimental Study

We compare the performance of `QRTC` (Section 4.3) against `BASE`, `QRC` and `QREG` on several metrics in a large-scale experiment. We also consider multiple ablated versions of `QRTC`. We build on the large-scale empirical study of Dheur and Ben Taieb (2023) (Chapter 3) and consider the same underlying neural network architectures, datasets and metrics. For these experiments, 81926 models were trained during a total of 180 hours on 40 CPUs.

4.4.1 Benchmark datasets

The datasets are the same as in Chapter 3, namely 27 from the recently curated benchmark by OpenML (Grinsztajn et al., 2022), 18 obtained from the AutoML Repository (Gijsbers et al., 2019), and 12 from the UCI Machine Learning Repository (Dua and Graff, 2017). We divide each dataset into four sets: training (65%), validation (10%), calibration (15%), and test (10%). To ensure robustness, we repeat this partitioning five times randomly and then average the results. During the training process, we normalize both the features, X, and the target, Y, using their respective means and standard deviations derived from the training set. After obtaining predictions, we transform them back to the original scale. For all methods, we use early stopping (with a patience of 30) to choose the epoch that gives the smallest validation NLL.

To avoid a potential bias in our analysis, we exclude certain datasets that may not suitable for regression. We identify these datasets using the proportion of targets Y that are among the 10 most frequent values in the dataset, and we call this proportion the level of discreteness. Table A.1 in the Supplementary Material shows that 13 out of 57 datasets have a level of discreteness above 0.5 and these datasets appear in all 4 benchmark suites. Section D.1.5 where full results are available, shows that `QRTC` performs better on these datasets.



(a) Letter-value plots showing Cohen's d for different metrics w.r.t. **BASE**.

(b) Critical difference diagrams for different metrics.

Figure 4.3: Comparison of **QRTC**, **QRC**, **QREGC** and **BASE**, as detailed in Section 4.4.

4.4.2 Experimental setup

Base neural network model The base predictor $\hat{F}_{Y|X}$ is a mixture density network (Section 2.2.3) with $M = 3$ Gaussians components, where the means $\hat{\mu}_m(X)$, standard deviations $\hat{\sigma}_m(X)$, and weights $\hat{\pi}_m(X)$, for each component $m = 1, ..., M$ are obtained as outputs of a hypernetwork, which is a 3-layer MLP with 128 hidden units per layer. We also consider other base predictors in Section D.1.2.

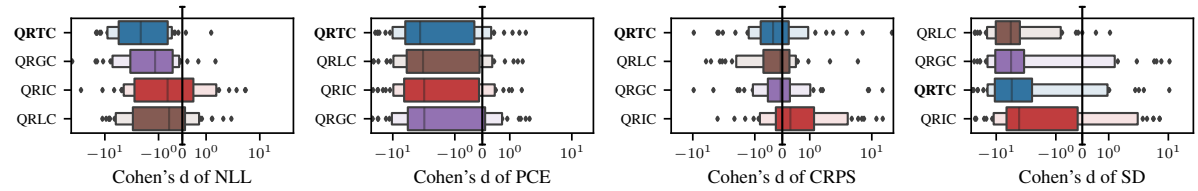
Compared methods We compare all methods in Table 4.1 where QR is applied, namely **QRC**, **QRTC** and **QREGC**. As in Chapter 3, to ensure a fair comparison, the calibration dataset is used as additional training data for the base predictor $\hat{F}_{Y|X}$ since there is no need for a calibration dataset. For **QRTC** and **QREGC**, the bandwidth b of ϕ_{KDE} is selected by minimizing the validation NLL in the set $\{0.01, 0.05, 0.1, 0.2\}$. Section D.2.1 shows that QRT does not require extensive tuning of the hyperparameter b. In fact, good results are already obtained with a default value of $b = 0.1$. For **QREGC**, we select $\lambda = -\beta$ where $\lambda \in \{0, 0.01, 0.05, 0.2, 1, 5\}$ and minimizes PCE with a maximum increase in continuous ranked probability score (CRPS) of 10% in the validation set, as in Dheur and Ben Taieb (2023). Since none of the compared methods introduce additional parameters compared to the baseline, all methods estimate parameters in the same space Θ.

Metrics We evaluate probabilistic predictions using the NLL and CRPS, which are strictly proper scoring rules. Probabilistic calibration is measured using PCE (3.1). Finally, we measure sharpness using the mean standard deviation of the predictions, denoted by SD. Similarly to Section 3.6.1 in the previous chapter, we report Cohen's d as a standardized effect size metric, and critical difference diagrams to determine if there's a significant difference in model performance compared to a baseline.

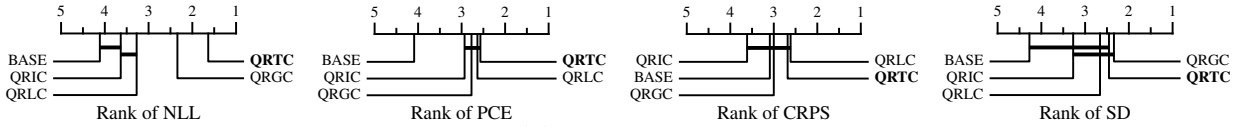
4.4.3 Results

Figure 4.2 illustrates the comparison in NLL of **QRTC** and **QRC** across various datasets, relative to **BASE**. We observe that **QRTC** consistently achieves a lower NLL on the majority of the datasets. This suggests that allowing the model to adapt to the calibration map during training improves the final predictive accuracy, without the need for extra parameters.

Figure 4.3 shows the letter-values plots for Cohen's d of different metrics (top panel) as well as



(a) Letter-value plots showing Cohen's d for different metrics w.r.t. `BASE`.



(b) CD diagrams

Figure 4.4: Comparison of `QRTC`, `QRGC`, `QRIC`, `QRLC` and `BASE` as detailed in Section 4.5.

the associated critical difference diagram (bottom panel), for all methods and datasets. The reference model is `BASE`. Figure 4.3b shows that our proposed method, `QRTC`, is able to significantly outperform the baseline and other methods in terms of test NLL, as suggested by Figure 4.2. In terms of PCE, since all considered methods except `BASE` are combined with QR, they benefit from the finite-sample guarantee (4.2) and achieve a similar PCE, outperforming `BASE`.

We also observe that there is no significant difference in terms of the CRPS of `QRTC` compared to other methods. This suggests that QRT is able to place a high density at the observed/realized test data points while the characteristics measured by CRPS, a distance-sensitive scoring rule (H. Du, 2021), are not significantly impacted. Furthermore, all methods produce sharper predictions than `BASE`, suggesting that `BASE` is underconfident, despite achieving similar NLL than `QRC`.

4.4.4 The importance of the base predictor

We note that previous studies on calibration have often focused on single Gaussian predictions with a small number of layers (Lakshminarayanan, Pritzel, et al., 2017; Utpala and Rai, 2020; S. Zhao et al., 2020). These models have been outperformed in terms of NLL and CRPS by mixture predictions (Dheur and Ben Taieb, 2023). Following Dheur and Ben Taieb (2023), we consider a 3-layer MLP that predicts a mixture of 3 Gaussians.

To further understand the role of the flexibility of the base predictor, we consider a 3-layer MLP with varying number of components in the mixture as well as a ResNet. We observe that, in all scenarios, `QRTC` outperforms `QRC` on most datasets in terms of NLL. Moreover, the enhancement is most pronounced in the case of misspecified single Gaussian mixture predictions. Detailed results are available in Section D.1.2.

4.5. An Ablation Study

In addition to the methods compared above, we provide an ablation study in order to understand the importance of the different components of QRT. We consider three ablated versions of QRT that differ from `QRTC` by one aspect each.

`QRIC`, for *QRT at initialization only*, corresponds to `QRTC` except that the calibration map $\hat{F}_{\hat{U}}$ is

computed once before the first training step and is fixed during the rest of training (except for the last post-hoc step). The goal is to show that improved initialization is not the only strength of QRT.

`QRGC`, for *QRT without gradient backpropagation*, corresponds to `QRTC` except that backpropagation does not occur on the computation graph generated by the calibration map, i.e., when computing $\hat{U}^{(i)}$ in Algorithm 8. While `QRTC` considers the calibration map as part of the model, `QRGC` considers the calibration map as an external actor that modifies the predictions at each step. The goal is to show that merely applying QR at each training step is not sufficient unless it is considered an integral part of the model.

`QRLC`, for *QRT with learned calibration map*, corresponds to `QRTC` except that the PITs $\hat{U}^{(i)}$ in Algorithm 8 are replaced by additional learned parameters initialized uniformly between 0 and 1. Thus, `QRLC` possesses B more parameters than `QRTC`. The goal is to show that the benefits of QRT are not only due to the additional flexibility provided by the calibration map.

Figure 4.4 shows a comparison of these ablated versions of `QRTC` against `QRTC`. We observe that all ablated versions result in significantly decreased NLL compared to `QRTC`, highlighting the strengths of the different components of QRT. Additionally, the CRPS and PCE show no improvement compared to `QRTC`, and all ablated versions result in slightly sharper predictions than `BASE`.

4.6. Conclusion

We introduced quantile recalibration training (QRT), a novel method that produces predictive distributions that are probabilistically calibrated by design at each training step. We demonstrated the effectiveness of this approach through a large-scale experiment and an ablation study. Our results indicate that QRT followed by a post-hoc QR step demonstrates enhanced performance in predictive accuracy (NLL) while maintaining or improving calibration compared to the baselines. This combination presents a compelling option to produce predictive distributions that are both accurate and well-calibrated. We also discussed the issue of training regression models on datasets with a discrete output variable. For future work, we suggest extending our method to encompass other calibration notions, such as distribution calibration (H. Song et al., 2019). Additionally, integrating other calibration methods, such as Conformal Quantile Regression (Romano et al., 2019), into the training procedure is an interesting direction to explore.

Limitations. While QRT has proven effective in improving both calibration and predictive accuracy, several limitations should be acknowledged. First, the methodology and theoretical guarantees are currently restricted to single-output regression under i.i.d. sampling, and do not extend to multivariate outputs or to scenarios involving distribution shift. Second, the guarantee established by QRT is marginal, and does not ensure calibration within specific subgroups or conditionally on the input features. Future work could address this by incorporating group-calibration approaches Section 2.4.5 or by learning conditional calibration maps (Dey et al., 2022), though such extensions may come at the cost of weakening the finite-sample probabilistic calibration guarantee. Third, the differentiable calibration step introduces non-negligible computational overhead during training; nevertheless, the additional cost is fixed

per training step, similar to other regularization methods, and not prohibitive. Finally, QRT requires additional design choices, such as the selection of kernel family and bandwidth b, but our experiments suggest that stable default settings are sufficient in practice.

CHAPTER 5

Multi-Output Conformal Regression

This chapter is based on the following papers:

> **Victor Dheur**, Matteo Fontana, Yorick Estievenart, Naomi Desobry, and Souhaib Ben Taieb (2025). A Unified Comparative Study with Generalized Conformity Scores for Multi-Output Conformal Regression. *The 42nd International Conference on Machine Learning.*
>
> **Victor Dheur***, Tanguy Bosser*, Rafael Izbicki, and Souhaib Ben Taieb (2024). Distribution-Free Conformal Joint Prediction Regions for Neural Marked Temporal Point Processes. *Machine Learning, Volume 113.*

For the first paper, Victor Dheur developed the methodology and conducted the experimental studies. Matteo Fontana contributed to the design of the latent-based conformity score methodology. The research was carried out under the supervision of Souhaib Ben Taieb. All authors contributed to the preparation and revision of the manuscript.

For the second paper, Victor Dheur and Tanguy Bosser jointly developed the methodology and conducted the experiments. The research was carried out under the supervision of Souhaib Ben Taieb. All authors contributed to the preparation and revision of the manuscript.

5.1. Introduction

Many real-world problems require predicting multiple, often dependent, outputs simultaneously. For instance, medical diagnostics may involve tracking several correlated health indicators, such as a patient's blood pressure and cholesterol levels, to monitor disease progression (Rajkomar et al., 2018). While modern probabilistic models can capture complex dependencies, they often produce unreliable or overly confident predictions (Nalisnick et al., 2018).

*Equal contribution

Conformal prediction (CP) offers a robust framework for improving model reliability by generating distribution-free prediction sets with a finite-sample coverage guarantee. Extending CP to multi-output settings is non-trivial because the multidimensional space $\mathbb{R}^d$ lacks the natural ordering of the real line. For example, the conformity scores of DCP and CQR in Section 2.5.4 require a CDF or QF. Other approaches achieve only marginal coverage by combining univariate regions, failing to capture dependencies between variables (Y. Zhou et al., 2024). Others, based on sampling or numerical approximations, can be computationally prohibitive (Izbicki et al., 2022; Z. Wang et al., 2023; Plassier et al., 2025a), while some may not achieve desirable conditional coverage properties (Sadinle et al., 2019). Figure 5.1 illustrates the diverse prediction sets produced by different methods on a simple bivariate problem.

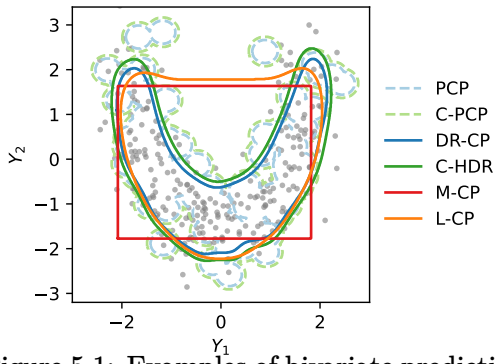


Figure 5.1: Examples of bivariate prediction sets with an 80% coverage level for a toy example.

In Dheur et al. (2025), we conduct a thorough comparative analysis of existing methods and introduce novel classes of generalized conformity scores designed to produce sharp, reliable, and computationally efficient prediction sets that adapt to the multivariate structure of the data. We make the following contributions:

- We present a unified comparative study of nine conformal methods for constructing multivariate prediction sets within the same framework. This study highlights their key properties while also exploring the connections between them. We examine different conformity scores with different multivariate base predictors, discussing prediction sets derived from the marginal distributions of individual output variables, their joint PDF, or sampling procedures (e.g., generative models).

- We introduce two novel classes of conformity scores for multi-output regression that generalize their univariate counterparts. These scores ensure asymptotic conditional coverage while maintaining exact finite-sample marginal coverage.

  The first, CDF-based scores, leverage the cumulative distribution function (CDF) of any conformity score to achieve asymptotic conditional coverage. This approach generalizes the univariate HPD-split score, based on univariate highest density region from Izbicki et al. (2020), to multivariate prediction sets derived from any conformity score. Additionally, we propose a specific instance of CDF-based scores that builds on PCP (Z. Wang et al., 2023). This method avoids the estimation of a predictive density, instead relying solely on samples from any generative model.

  The second, latent-based scores are inspired by Feldman et al. (2023) and can be interpreted as an extension of distributional conformal prediction (Chernozhukov et al., 2021) to multivariate outputs. Compared to Feldman et al. (2023), it does not require directional quantile regression, and the conformalization is performed directly in the latent space, eliminating the need to construct a grid. This enhances both computational efficiency and scalability.

- We conduct a large-scale empirical study comparing the different multi-output conformal methods across 13 tabular datasets with multivariate outputs, evaluating several performance metrics. We consider a variety of multi-output regression models, namely multivariate quantile function forecaster (MQF^2, Kan et al., 2022), Distributional random forests (DRFs, Cevid et al., 2022), and Cholesky-based mixture density networks (MDNs, Bishop, 1994; Muschinski et al., 2022).

5.2. Background

We focus on the SCP framework, which we reviewed in Section 2.5.1. Many conformal prediction methods have been proposed in the literature and implemented within the SCP framework for various base predictors and conformity scores, with a specific focus on univariate prediction problems. In this section, we survey several conformal methods for constructing multivariate prediction sets, using different multivariate base predictors and corresponding conformity scores. Specifically, we discuss density-based, and sample-based methods, which are based on their joint PDF, or a sampling procedure (e.g., a generative model), respectively. In the following, we describe the conformity scores for different methods. The methods `M-CP` and `CopulaCPTS`, which produce hyperrectangular sets, are detailed in Section E.2.

Once a conformity score is defined, SCP computes the $1-\alpha$ empirical quantile of the scores on the calibration dataset, denoted $\hat{q}$. Then, the corresponding prediction set is defined by

$$\hat{R}(x) = \{y \in \mathcal{Y} : s(x,y) \leqslant \hat{q}\}. \tag{5.1}$$

We detail the connection between the conformity score and prediction set of each consider methods in Section E.3. Furthermore, in Section 5.5, we analyze the properties and relationships between these methods and provide illustrative examples of the resulting prediction sets.

DR-CP. Given a predictive density $\hat{f}_{Y|X=x}$, a direct conformity score is the negative density:

$$s_{\text{DR-CP}}(x,y) = -\hat{f}_{Y|X=x}(y). \tag{5.2}$$

We call this baseline method DR-CP for Density-Rank conformal prediction. The corresponding prediction set is a density superlevel set, $\hat{R}_{\text{DR-CP}}(x) = \{y \in \mathcal{Y} : \hat{f}_{Y|X=x}(y) \geqslant -\hat{q}\}$. Sadinle et al. (2019) use this conformity score in the context of classification.

C-HDR. Izbicki et al. (2022) proposed the HPD-split method, which defines a conformity score based on the Highest Predictive Density (HPD):

$$\begin{aligned} \text{HPD}_{\hat{f}_{Y|X=x}}(y) &= \int_{\{y' | \hat{f}_{Y|X=x}(y') \geqslant \hat{f}_{Y|X=x}(y)\}} \hat{f}_{Y|X=x}(y')dy' && (5.3)\\ &= \mathbb{P}\left(\hat{f}_{Y|X=x}(\hat{Y}) \geqslant \hat{f}_{Y|X=x}(y) \mid X = x\right), && (5.4) \end{aligned}$$

where $\hat{Y} \sim \hat{f}_{Y|X=x}$. The corresponding prediction set is a highest density region (HDR, Hyndman, 1996) w.r.t. $\hat{f}_{Y|X=x}$ at level $\hat{q}$:

$$\begin{aligned} \hat{R}_{\text{C-HDR}}(x) &= \{y \in \mathcal{Y} : \hat{f}_{Y|X=x}(y) \geqslant t_{\hat{q}}\}, \\ &\text{where } t_{\hat{q}} = \sup\{t : \mathbb{P}(\hat{f}_{Y|X=x}(\hat{Y}) \geqslant t \mid X = x) \geqslant \hat{q}\}. \end{aligned} \tag{5.5}$$

Compared to DR-CP, where the threshold $-\hat{q}$ is independent of x, C-HDR allows the threshold $t_{\hat{q}}$ to vary with x. To compute the HPD in (5.3), Izbicki et al. (2022) use numerical integration, whereas in our experiments, we approximate (5.4) using Monte Carlo sampling, as described in (5.10). When using the Monte Carlo sampling approach, we call this method C-HDR for conformalized HDR.

In the context of classification, the Adaptive Prediction Sets (Romano et al., 2020) method follows a similar principle by constructing a "highest mass region", which corresponds to a superlevel set of the probability mass function with probability content at least $\hat{q}$.

PCP. Let $\tilde{Y}^{(1)}, \tilde{Y}^{(2)}, \ldots, \tilde{Y}^{(L)}$ denote a sample with L points from the (estimated) conditional distribution $\hat{P}_{Y|X=x}$. Probabilistic conformal prediction (PCP, Z. Wang et al. (2023)) defines a conformity score as the closest distance to y:

$$s_{\text{PCP}}(x, y) = \min_{l \in [L]} \|y - \tilde{Y}^{(l)}\|, \tag{5.6}$$

$$\text{where } \tilde{Y}^{(l)} \sim \hat{P}_{Y|X=x}, \quad l \in [L]. \tag{5.7}$$

The corresponding prediction set is a union of L balls centered at each sampled point $\tilde{Y}^{(l)}$, i.e. $\hat{R}_{\text{PCP}}(x) = \bigcup_{l \in [L]} \{y \in \mathcal{Y} : \|y - \tilde{Y}^{(l)}\| \leqslant \hat{q}\}$.

HD-PCP. When a predictive density is available alongside a sample of L points, Z. Wang et al. (2023) proposed an extension to PCP, called HD-PCP. This method uses the same conformity score as in (5.6), but only retains the $\lfloor (1-\alpha)L \rfloor$ samples with the highest density, ensuring that the prediction set is concentrated on high-density points.

ST-DQR. Motivated by the limitation that existing multivariate quantile regression methods do not allow the construction of prediction sets with arbitrary shapes, Feldman et al. (2023) proposed to construct convex regions in a latent space $\mathcal{Z}$ using directional quantile regression (Paindaveine and Šiman, 2011). These regions are then mapped to the output space $\mathcal{Y}$ using a conditional variational autoencoder (CVAE), allowing a non-linear mapping between the two spaces. Specifically, they apply a conformalization step by creating a grid of points within the region in $\mathcal{Z}$, map the points to the output space $\mathcal{Y}$, and construct d-balls around the mapped samples, similarly to PCP.

5.3. Related Work

Conformal prediction (CP), introduced by Vovk et al. (1999), forms the foundation of our work by providing prediction sets with finite-sample coverage guarantees. CP methods are well established for regression with univariate outputs (Papadopoulos et al., 2008; Lei and Wasserman, 2014; Romano et al., 2019; Sesia and Romano, 2021) and classification (Romano et al., 2020; Angelopoulos et al., 2021). In the multi-output regression setting, we need to capture dependencies between output dimensions, represent more complex prediction sets and handle a larger computational demand.

To address multivariate prediction challenges, optimal transport methods such as cyclically monotone mappings (Carlier et al., 2016) define multivariate quantile regions with desirable

properties such as existence and uniqueness of mappings. Hallin and Šiman (2017), Hallin et al. (2021), and Barrio et al. (2024) have proposed extensions of these approaches. Neural network-based techniques leverage normalizing flows (Kan et al., 2022; C.-W. Huang et al., 2021) or variational autoencoders (Feldman et al., 2023) to learn flexible quantile regions. Additionally, highest density regions (HDRs) (Hyndman, 1996) handle multimodality and have been applied in various contexts (Camehl et al., 2024; Izbicki et al., 2022; Dheur et al., 2024). Recently, Z. Wang et al. (2023) proposed constructing prediction sets as hyperballs centered on generated samples, with extensions by Plassier et al. (2025a) improving conditional coverage. Other methods use copulas (Messoudi et al., 2021; S. H. Sun and Yu, 2024) to model the dependency between variables. Section E.1 provides a more detailed discussion on related work.

Recent research has extended CP in several directions, including methods for handling distribution shift (Tibshirani et al., 2019; Gibbs and Candes, 2021), non-exchangeable data (Foygel Barber et al., 2021a; Zaffran et al., 2022), federated learning (C. Lu et al., 2023), and scenarios where multiple observations of the same input are available at prediction time (Fermanian et al., 2025).

5.4. Generalized Conformity Scores for Multi-Output Regression

In this section, we introduce two new classes of conformity scores: *CDF-based* and *latent-based* scores. These scores generalize existing conformity scores for single-output regression to accommodate any conformity score for multivariate outputs. The former generalizes HPD-Split (Izbicki et al., 2020) to any conformity score, allowing to apply this method to multivariate outputs. We further propose a specific instance that builds on PCP (Z. Wang et al., 2023). The latter is inspired by Feldman et al. (2023) and can be interpreted as an extension of distributional conformal prediction (Chernozhukov et al., 2021) for multivariate outputs. Section 5.5 will present a comparative study of the conformity scores introduced in Section 5.2 alongside those introduced in this section.

5.4.1 CDF-based conformity scores

Consider a conformity score s_W, and define the random variable $W = s_W(X, Y)$ for a random pair (X, Y). For an observation (x, y), we introduce a new conformity score based on the conditional CDF of W given $X = x$, evaluated at $s_W(x, y)$. Specifically, the score is given by

$$s_{\text{CDF}}(x, y) = \mathbb{P}(s_W(X, Y) \leqslant s_W(x, y) \mid X = x) \tag{5.8}$$

$$= F_{W|X=x}(s_W(x, y)). \tag{5.9}$$

This new conformity score measures the rank of $s_W(x, y)$ relative to the conditional distribution of W given $X = x$.

This method applies to any conformity score s_W and generalizes the (oracle) HPD-split introduced in Izbicki et al. (2020) in the context of single-output regression. Specifically, when $s_W(x, y) = s_{\text{DR-CP}}(x, y)$ is used in (5.9), we recover the `C-HDR` method. Additionally, by the probability integral transform, $s_{\text{CDF}}(X, Y) \mid X = x \sim \mathcal{U}(0, 1)$ for $x \in \mathcal{X}$, meaning that the conformity score's distribution is independent of x. This property ensures that conditional coverage is achieved as $|\mathcal{D}_{\text{cal}}| \to \infty$ (see Section E.5.2, Lemma 8). A similar observation was made by Izbicki et al. (2020) for `C-HDR`.

However, in practice, since the distribution of $Y \mid X = x$ is unknown, we approximate s_{CDF} using Monte Carlo sampling:

$$s_{\text{ECDF}}(x, y) = \frac{1}{K} \sum_{k \in [K]} \mathbb{1}\left(s_W(x, \hat{Y}^{(k)}) \leqslant s_W(x, y)\right),$$
$$\text{where } \hat{Y}^{(k)} \sim \hat{P}_{Y|X=x},\ k \in [K]. \tag{5.10}$$

Dheur et al. (2024) considered a particular case of this empirical CDF-based approach with the $s_{\text{DR-CP}}$ score for a bivariate prediction problem involving temporal point processes, where the HDR is estimated via Monte Carlo sampling.

C-PCP. We introduce a special case of our new score, called `C-PCP` (CDF-based Probabilistic Conformal Prediction), by setting $s_W(x, y) = s_{\text{PCP}}(x, y)$ in (5.10), which gives:

$$s_{\text{C-PCP}}(x, y) = \frac{1}{K} \sum_{k \in [K]} \mathbb{1}\left(\min_{l \in [L]} \|\hat{Y}^{(k)} - \tilde{Y}^{(l)}\| \leqslant \min_{l \in [L]} \|y - \tilde{Y}^{(l)}\|\right).$$

Compared to the methods in Izbicki et al. (2020) and Dheur et al. (2024), this score has the advantage of not requiring the estimation of a predictive density, relying instead on samples from the conditional distribution. Consequently, this score can be applied with any generative model that does not have an explicit density, while still retaining the desirable properties of our CDF-based score.

Interestingly, `C-PCP` shares similarities with the recently proposed CP^2-PCP method by Plassier et al. (2025a). For a given $x \in \mathcal{X}$, both methods adapt the radius of the balls based on a second sample from the conditional distribution composed of K points, requiring a total of $L + K$ samples. A detailed discussion can be found in Section E.8.

5.4.2 Latent-based conformity scores

Inspired by Feldman et al. (2023), we propose a latent-based conformity score with key distinctions. First, our method does not require the use of directional quantile regression. Additionally, the conformalization step is performed in the latent space, eliminating the need to construct a grid, which improves both computational efficiency and scalability.

Our base predictor is a conditional invertible generative model $\hat{T} : \mathcal{Z} \times \mathcal{X} \to \mathcal{Y}$, which maps a latent random variable $Z \in \mathcal{Z}$ (e.g., drawn from a standard multivariate normal distribution) to the output space $\mathcal{Y}$, conditional on $X \in \mathcal{X}$ (e.g., using normalizing flows). The model is both conditional and invertible, meaning that

$$\hat{T}(\hat{T}^{-1}(y; x); x) = y, \forall x \in \mathcal{X}, y \in \mathcal{Y}.$$

We propose the following conformity score, called `L-CP` (Latent-based conformal prediction), defined as:

$$s_{\text{L-CP}}(x, y) = \rho_{\mathcal{Z}}(\hat{T}^{-1}(y; x)), \tag{5.11}$$

where $\rho_{\mathcal{Z}} : \mathcal{Z} \to \mathbb{R}$ is a conformity function in the latent space $\mathcal{Z}$, independent of x. In our experiments, we use $Z \sim \mathcal{N}(0, I_d)$ and $\rho_{\mathcal{Z}}(z) = \|z\|$.

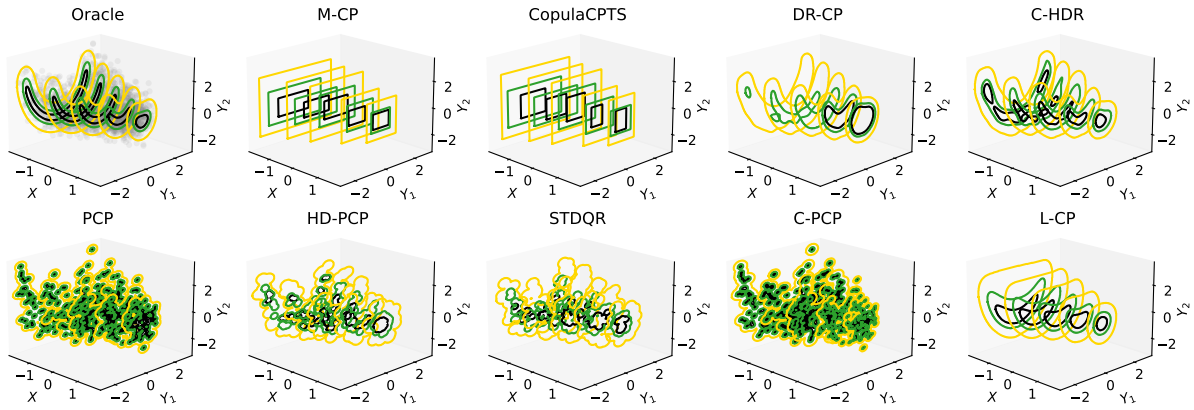


Figure 5.2: Prediction sets for a bivariate unimodal dataset, conditional on a univariate input. The black, green, and yellow contours represent regions with nominal coverage levels of 20%, 40%, and 80%, respectively.

The corresponding prediction set is obtained by mapping a region in the latent space, $R_{\mathcal{Z}}(\hat{q}) = \{z \in \mathcal{Z} : \rho_{\mathcal{Z}}(z) \leqslant \hat{q}\}$, to a region in the output space, $\hat{R}_{\text{L-CP}}(x) = \{\hat{T}(z; x) : z \in R_{\mathcal{Z}}(\hat{q})\}$.

`L-CP` generalizes distributional conformal prediction (Chernozhukov et al., 2021), which is a special case when Y is univariate ($d = 1$), $Z \sim \mathcal{U}(0, 1)$, $\rho_{\mathcal{Z}}(z) = |z - \frac{1}{2}|$, and $\hat{T}(\cdot; x)$ is the quantile function of Y given x.

Concurrent work by Fang et al. (2025) introduces CONTRA, sharing the same algorithm as our latent-based methods. While related, the papers diverge in their primary focus. Fang et al. (2025) emphasizes the smaller prediction sets achieved by CONTRA, whereas our work concentrates on the computational complexity and conditional coverage guarantees of the latent-based methods while obtaining set sizes that are small but not smaller than density-based methods.

5.5. Comparison of Multi-Output Conformal Methods

In this section, we present a unified comparison of the conformity scores introduced in Section 5.2 and the generalized scores proposed in Section 5.4.

5.5.1 Illustrative examples

We provide illustrative examples of bivariate prediction sets for different conformal methods on simulated data, covering both unimodal (Figure 5.2) and bimodal distributions (Figure E.4 in Section E.4.2). The data-generating processes are given in Section E.4.2. Additionally, we present bivariate prediction sets for a real-world application, predicting a taxi passenger's drop-off location based on the passenger's information (Figures E.2 and E.3 in Section E.4.1).

In both Figures 5.2 and E.4, the black, green, and yellow contours represent prediction sets with nominal coverage levels of 20%, 40%, and 80%, respectively. The top-left panel illustrates the density level sets of the oracle distribution $P_{Y|X}$. The remaining panels display the prediction sets generated by various conformal methods, all utilizing the MQF^2 base predictor, as explained in Section E.6.2.

Table 5.1: Properties of different multivariate conformal methods. (*) `M-CP` achieves ACC under certain assumptions (Section E.5.2). (**) `STDQR` and `L-CP` require a conditional invertible generative model $\hat{T} : \mathcal{Z} \times \mathcal{X} \to \mathcal{Y}$. (†) `CopulaCPTS` has a pre-training cost of $O(t_C)$.

Method	Type of region	Asymptotic conditional coverage	Computational complexity	Predictive density not required	Sampling procedure not required
`M-CP`	Hyperrectangle	✗(*)	$O(dt_M)$	✓	✓
`CopulaCPTS`	Hyperrectangle	✗	$O(dt_M)^\dagger$	✓	✓
`DR-CP`	Density superlevel set	✗	$O(t_D)$	✗	✓
`C-HDR`	Density superlevel set	$K \to \infty$	$O(K(t_D + t_S))$	✗	✗
`PCP`	Union of d-balls	✗	$O(Lt_S)$	✓	✗
`HD-PCP`	Union of d-balls	✗	$O(L(t_D + t_S))$	✗	✗
`STDQR`	Union of d-balls	✗	$O(Lt_S)$	✓(**)	✗
`C-PCP`	Union of d-balls	$K \to \infty$	$O((K + L)t_S)$	✓	✗
`L-CP`	Quantile region	✓	$O(t_Q)$	✓(**)	✗

We observe the following for the unimodal case in Figure 5.2. `M-CP` and `CopulaCPTS` capture heteroscedasticity but produce rectangular prediction sets, which do not align with the circular level sets of the oracle conditional distribution, resulting in a lack of sharpness. `DR-CP` fails to maintain conditional coverage, and for $X = 1$, the absence of black and green contours indicates that the predictive density does not reach the threshold $-\hat{q}$ defined in (5.2) for coverage levels of 0.2 and 0.4. `C-HDR` generates prediction sets that closely resemble the oracle level sets. `PCP` generates highly discontinuous regions, especially at lower coverage levels, where the regions appear as balls centered on individual samples. In contrast, `HD-PCP` and `STDQR` yield smoother, more continuous regions but require the estimation of a predictive PDF or the identification of a map from the latent space to the output space, respectively.

For our methods, unlike `PCP`, `C-PCP` adjusts the radius of the prediction sets to improve conditional coverage. This is evident in the example, where the radius of the balls for $X = -1$ is smaller than for $X = 1$, as indicated by the tighter regions around the samples. `L-CP` generates prediction sets that closely align with the oracle level sets, demonstrating good conditional coverage.

For the bimodal distribution in Figure E.4 (Section E.4.2), the prediction sets generated by `M-CP` and `L-CP` are connected, failing to capture the bimodal nature of the distribution. For the real-world application, Figures E.2 and E.3 (Section E.4.1) illustrate predictions under low and high uncertainty, respectively. Our methods, `L-CP` and `C-PCP`, alongside `M-CP` and `C-HDR`, demonstrate the best adaptability to outputs with varying levels of uncertainty.

5.5.2 Properties

In this section, we compare conformal methods based on several key properties. In the following, we use $\overset{\text{d.}}{=}$ to denote equality in distribution and $\overset{\text{a.s.}}{=}$ to denote almost sure equality.

Marginal coverage. All the conformal methods presented achieve the classical finite-sample *marginal coverage*. But, as noted by Z. Wang et al. (2023) (Theorem 1), the marginal coverage of methods such as `C-HDR`, `PCP`, `HD-PCP`, and `C-PCP` also depends on the randomness of the generated samples. In Section E.5.1, we demonstrate that the marginal coverage, conditional on

the calibration dataset $\mathcal{D}_{\text{cal}}$ and the samples drawn from it, follows a beta distribution, using standard arguments. `CopulaCPTS` is the only method that does not enter into the standard split conformal algorithm and who does not satisfy the above property.

Asymptotic conditional coverage (ACC). We examine the *asymptotic conditional coverage* property, which corresponds to conditional coverage as defined in (2.59) under the assumptions that $|\mathcal{D}_{\text{cal}}| \to \infty$ and the base predictor corresponds to the oracle distribution $P_{Y|X}$.

While the assumption of oracle base predictor is strong, it is essential to demonstrate that the conformal procedure preserves the performance of the base predictor. Specifically, given $x \in \mathcal{X}$, for `M-CP` and `CopulaCPTS`, we assume $\hat{l}_i(x) = Q_{Y_i|X=x}(\alpha_l)$ and $\hat{u}_i(x) = Q_{Y_i|X=x}(\alpha_u)$ with $i = 1, \ldots, d$; for `DR-CP`, `C-HDR`, and `HD-PCP`, $\hat{f}_{Y|X=x} = f_{Y|X=x}$; for `L-CP`, $\hat{T}(Z; x) \overset{\text{d.}}{=} Y|(X = x)$; and for `PCP` and `C-PCP`, $\hat{P}_{Y|X=x} = P_{Y|X=x}$.

Our empirical results (Section 5.6) demonstrate that methods achieving ACC under these assumptions also exhibit superior approximate conditional coverage across diverse datasets and base predictors. `L-CP` is the only method that achieves ACC without additional assumptions. `C-HDR` and `C-PCP` achieve ACC with $K \to \infty$. Finally, `M-CP` achieves ACC under specific assumptions. Assuming that $Y_1, \ldots, Y_d$ are conditionally independent given X, `M-CP` achieves ACC if $\alpha_u - \alpha_l = \sqrt[d]{1-\alpha}$. Furthermore, under the unrealistic assumption that $Y_1 \mid X \overset{\text{a.s.}}{=} \ldots \overset{\text{a.s.}}{=} Y_d \mid X$, `M-CP` achieves ACC if $\alpha_u - \alpha_l = 1 - \alpha$. The true dependence typically lies between these two extremes. We provide detailed proofs of these statements in Section E.5.2.

As discussed in Section 5.5.1, `DR-CP` fails to achieve ACC. Likewise, `PCP`, `HD-PCP` and `STDQR` do not achieve ACC, as they are constrained to producing sets with upper bounded volume for any $x \in \mathcal{X}$. Assuming each ball has a volume of V, `PCP` generates sets with a total volume of at most LV. For a given instance $x \in \mathcal{X}$ with high uncertainty, it may be impossible to capture sufficient probability mass to achieve conditional coverage.

Region size. Among the methods that achieve ACC, `C-HDR` is expected to perform best, as it converges to the highest density regions, which correspond to the smallest volume regions (Hyndman, 1996). Prediction sets from `C-PCP` are expected to have a larger volume since they are constrained to a union of L d-balls. Similarly, prediction sets from `L-CP` are less flexible than those from `C-HDR`, as they are connected when the region $R_{\mathcal{Z}}(\lambda)$ in the latent space is connected for all $\lambda \in \mathbb{R}$ and $\hat{T}$ is continuous. This constraint may be desirable when more interpretable prediction sets are preferred (Sesia and Romano, 2021).

Among the remaining methods, `DR-CP` minimizes the mean prediction set size $\mathbb{E}\left[|\hat{R}(X)|\right]$ under the oracle PDF as $|\mathcal{D}_{\text{cal}}| \to \infty$, as shown in Theorem 1 by Sadinle et al. (2019). In contrast, `M-CP` and `CopulaCPTS` are expected to yield larger prediction sets, as they do not explicitly account for dependencies between outputs. While `PCP`, `HD-PCP`, and `C-PCP` can capture multimodality, they are susceptible to the randomness of the sampling procedure, as evidenced by the shape of the sets in Figure 5.2. Furthermore, since they rely on a finite union of L d-balls, they are subject to the curse of dimensionality in high-dimensional spaces, where data sparsity necessitates larger balls to maintain marginal coverage.

A potential weakness of the mean prediction set size is that it can be disproportionately skewed by inputs with high uncertainty. To mitigate this sensitivity, we also report the median prediction

set size as a more robust alternative.

Computational complexity. Table 5.1 reports the computational complexity of each conformity score. For `M-CP` and `CopulaCPTS`, let t_M represent the compute time of the univariate conformity score for a single dimension and t_C the optimization time for CopulaCPTS. Let t_D, t_S, and t_Q denote the time required for density evaluation, sampling, and calculating the inverse of the quantile function $\hat{T}^{-1}$, respectively. In many cases, t_M and t_C are relatively low, while t_D, t_S, and t_Q are comparable. `C-HDR`, `PCP`, `HD-PCP`, `STDQR` and `C-PCP` are significantly slower than `M-CP`, `L-CP`, and `DR-CP` since they need to generate a large number of samples to compute the conformity score (we used $K = L = 100$ in our experiments).

Base predictor. Some conformal methods stand out because they do not need to evaluate the predictive density $\hat{f}_{Y|X}$ or generate samples. `M-CP` and `CopulaCPTS` only require a univariate model for each dimension, without needing a model for the joint distribution of Y. `DR-CP` does not require sampling from the model, which is beneficial when using normalizing flows that are slower to invert (e.g., masked autoregressive flows (MAF, Papamakarios, Pavlakou, et al., 2017) or convex potential flows (C.-W. Huang et al., 2021)). `PCP` and `C-PCP` do not require evaluating the predictive density $\hat{f}_{Y|X}$, making them compatible with any generative model, including diffusion models and GANs. `L-CP` and `STDQR` do not require predictive density evaluation but require the model to be invertible. We summarize the different properties in Table 5.1.

5.5.3 Connection between sample-based and density-based methods

Interestingly, the sample-based methods (`PCP`, `HD-PCP`, `C-PCP`) can be viewed as special cases of density-based methods (`DR-CP`, `C-HDR`). Let us assume a common predictive PDF $\hat{f}_{Y|X}$ is used for the base predictor of these conformal methods. Let $\tilde{Y}^{(l)} \sim \hat{P}_{Y|X=x}$ for $l \in [L]$, and $f_{\mathbb{S}}(\cdot; \tilde{Y}^{(l)})$ be a PDF with spherical level sets, centered at $\tilde{Y}^{(l)}$, such as a standard multivariate Gaussian $\mathcal{N}(\cdot; \tilde{Y}^{(l)}, I_d)$. For $x \in \mathcal{X}$, we define a new PDF $\hat{f}^{\max}_{Y|X=x}(y) = \max_{l \in [L]} f_{\mathbb{S}}(y; \tilde{Y}^{(l)})/C$, where C is a normalizing constant ensuring that $\hat{f}^{\max}_{Y|X=x}(\cdot)$ integrates to 1. The following proposition establishes the relationship between these methods.

Proposition 3. `PCP` is equivalent to `DR-CP` with $\hat{f}_{Y|X=x} = \hat{f}^{\max}_{Y|X=x}$. Similarly, `HD-PCP` is equivalent to `DR-CP` with $\hat{f}_{Y|X=x} = \hat{f}^{\max}_{Y|X=x}$ where only $\lfloor (1-\alpha)L \rfloor$ samples with the highest density among $\{\tilde{Y}^{(l)}\}_{l \in [L]}$ are kept. Finally, `C-PCP` is equivalent to `C-HDR` with $\hat{f}_{Y|X=x} = \hat{f}^{\max}_{Y|X=x}$.

We provide a proof in Section E.5.3. Although these sample-based methods are special cases of density-based approaches, the key advantage of `PCP` and `C-PCP` is that they rely solely on a sampling procedure, without requiring a predictive density $\hat{f}_{Y|X}$ as a base predictor. Figure 5.3 summarizes the connections between the main conformal methods.

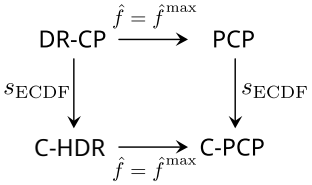


Figure 5.3: Connections between different methods.

An interesting practical takeaway is that `DR-CP` and `C-HDR` are linked in the same way as `PCP` and `C-PCP`. Since `DR-CP` under the oracle has the smallest mean set size while `C-HDR` empirically has a smaller median set size, similar observations are expected for `PCP` and `C-PCP`. This is verified empirically: `PCP` has a smaller mean set size across all base predictors, while `C-PCP` has a smaller median set size.

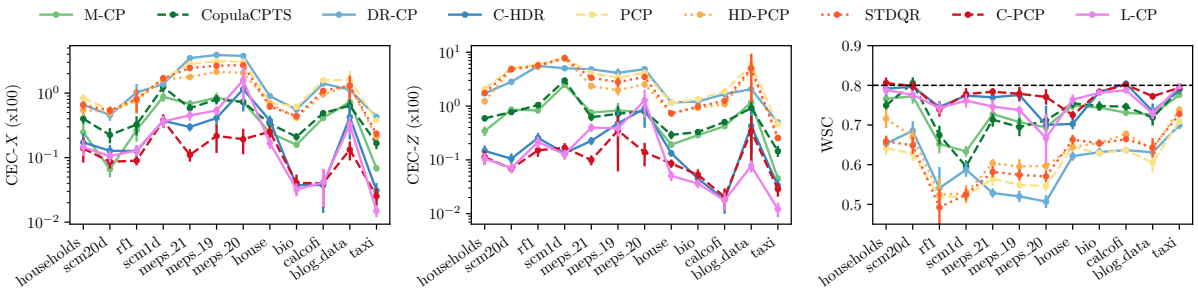


Figure 5.4: Conditional coverage metrics across datasets sorted by size. CEC-X and CEC-Z should be minimized while WSC should approach $1 - \alpha$.

5.6. A Large-Scale Study of Multi-Output Conformal Methods

In this section, we present a large-scale study of multi-output conformal methods using 13 tabular datasets from previous studies (Tsoumakas et al., 2011; Feldman et al., 2023; Z. Wang et al., 2023; Barrio et al., 2024; Camehl et al., 2024). To ensure sufficient data for training, calibration, and testing, we include only datasets with at least 2,000 instances. The selected datasets contain between 7,207 and 50,000 data points, with the number of input features p ranging from 1 to 279 and the number of output variables d ranging from 2 to 16.

We consider three base predictors: the Multivariate Quantile Function Forecaster (MQF^2), a normalizing flow (Kan et al., 2022), Distributional random forests (Cevid et al., 2022), and a multivariate Gaussian mixture model (Bishop, 1994). We present results for MQF^2 in the main text, while similar results for the other models are provided in Section E.7. We compare the methods using several metrics, including conditional coverage (WSC, CEC-X, and CEC-V), marginal coverage (MC), set size, and computational time. A detailed description of the experimental setup is provided in Section E.6.

Conditional coverage. Figure 5.4 presents the results for all datasets, ordered by increasing dataset size. On most datasets, `C-PCP`, `L-CP`, and `C-HDR` obtain the best conditional coverage. In contrast, `HD-PCP`, `STDQR`, `PCP`, and `DR-CP` are the least conditionally calibrated. Finally, `M-CP` and `CopulaCPTS` attain intermediate conditional coverage, with `M-CP` performing slightly better. These results align with our analysis in Section 5.5.2, where we showed that `C-PCP`, `L-CP`, and `C-HDR` achieve ACC, while `HD-PCP`, `STDQR`, `PCP`, and `DR-CP` do not, and `M-CP` achieves it only under specific conditions. Finally, Figure E.6 shows that all methods achieve marginal coverage, as expected.

Region size. Figure 5.5 presents a critical difference (CD) diagram (Demšar, 2006) comparing the median set size of all methods across datasets. Higher-ranked methods (further right) perform better. Thick horizontal lines indicate models with no statistically significant difference at the 0.05 level (see Section E.6.5 for details).

Among the methods that achieve ACC, `C-HDR` yields the smallest median set size, as expected, since its regions converge to the highest density regions (Izbicki et al., 2022). `C-PCP` and `L-CP` produce slightly larger regions, though the difference is not significant for these datasets. Among the remaining methods, `DR-CP` yields the smallest median region. In contrast, `M-CP` and

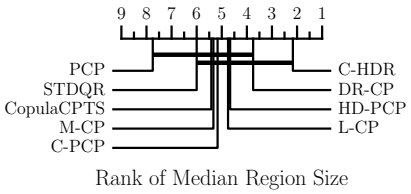


Figure 5.5: CD diagrams with the base predictor MQF^2 based on 10 runs per dataset and method.

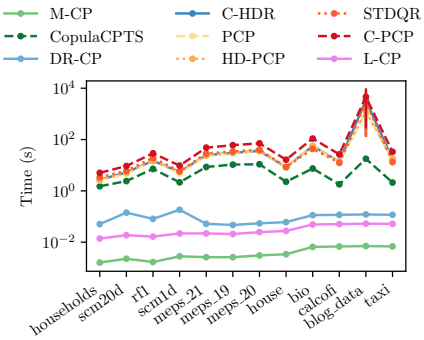


Figure 5.6: Total time in seconds for calibration and test.

`CopulaCPTS` generate larger regions, which is expected given their less flexible hyperrectangular shape. `PCP` tends to obtain the largest set sizes as it includes samples from low-density areas, whereas `STQDR` and `HD-PCP` mitigate this by removing samples from low-density areas, resulting in more compact regions. Finally, Figure E.7 in Section E.7 provides results for the mean set size, where `DR-CP` consistently performs best, under the oracle setting, it minimizes the expected set size, as explained in Section 5.5.2.

Computation time. Figure 5.6 shows the total computation time for each method. `M-CP` and `CopulaCPTS` have the shortest computation times, as they do not require learning a complex model for the output joint distribution. `L-CP` and `DR-CP` follow, benefiting from the absence of per-instance sampling. In contrast, sampling-based methods typically require 100 to 200 times more computation time.

5.7. Application to Continuous-Time Event Data

So far, this chapter has focused on constructing multi-output prediction sets for standard regression tasks on tabular data. We now turn to another application. Continuous-time event data, comprising sequences of events occurring at irregular intervals, is ubiquitous across fields such as healthcare (Enguehard et al., 2020), finance (Bacry and Muzy, 2014), and social media (Farajtabar et al., 2017). Temporal Point Processes (TPPs) provide a principled mathematical framework for modeling these sequences. In this section, we apply multi-output conformal prediction methods to TPPs to predict future events. This task provides a compelling use case for multi-output conformal regression, as it involves generating a joint prediction set for a bivariate output composed of a continuous variable (the event's arrival time) and a categorical one (the event's mark). We consider either combining individual predictions regions or directly producing highest density regions, with both heuristic and conformal methods. This section is based on Dheur et al. (2024), where more details and related works on TPPs are provided.

5.7.1 Problem Formulation

A marked temporal point process describes a sequence of events $\{(t_j, k_j)\}_{j=1}^{m}$, where $t_j \in \mathbb{R}^+$ is the arrival time and $k_j \in [K]$ is the associated mark. The task is to predict the next event given the history of past events $\mathcal{H}_t = \{(t_i, k_i) \mid t_i < t\}$.

We frame this as a supervised learning problem. The input X is a fixed-dimensional embedding $\boldsymbol{h}$ of the history $\mathcal{H}_t$, generated by a neural encoder. The output Y is the next event, represented as a pair (τ, k), where $\tau \in \mathbb{R}^+$ is the inter-arrival time (time since the last event) and $k \in [K]$ is the mark. The output space is thus $\mathcal{Y} = \mathbb{R}^+ \times [K]$.

Our base predictor is a neural TPP model, specifically the Conditional LogNormMix (CLNM) model (Shchur et al., 2020; Bosser and Ben Taieb, 2023), which learns an estimate of the conditional joint PDF $\hat{f}(\tau, k \mid \boldsymbol{h}) = \hat{f}_{Y|X=x}(y)$, with $y = (\tau, k)$ and $x = \boldsymbol{h}$. This density provides the foundation for constructing conformal prediction sets.

5.7.2 Individual Prediction Sets for Arrival Times and Marks

Before constructing joint prediction sets, we first define the methods for creating marginal prediction sets for the arrival time, $\hat{R}_\tau(\boldsymbol{h})$, and the mark, $\hat{R}_k(\boldsymbol{h})$. These serve as building blocks for one of the joint prediction strategies.

Arrival Time. The inter-arrival time τ is a positive, right-skewed variable. A suitable heuristic approach, *Heuristic QRL (H-QRL)*, constructs an asymmetric interval directly from the estimated quantile function for a given miscoverage level α:

$$\hat{R}_{\tau,\text{H-QRL}}(\boldsymbol{h}) = [0, \hat{Q}_{\tau|X=\boldsymbol{h}}(1-\alpha)]. \tag{5.12}$$

To ensure valid coverage, we use its conformal counterpart, *Conformalized Quantile Regression Left (C-QRL)*. This method adapts the CQR framework (see Section 2.5.4) using the conformity score:

$$s_{\text{C-QRL}}(\boldsymbol{h}, \tau) = \tau - \hat{Q}_{\tau|X=\boldsymbol{h}}(1-\alpha). \tag{5.13}$$

Mark. For the categorical mark k, the heuristic method *Heuristic RAPS (H-RAPS)* forms a set of likely marks by taking all marks whose RAPS score is below the target level:

$$\hat{R}_{k,\text{H-RAPS}}(\boldsymbol{h}) = \{k' \in [K] : s_{\text{RAPS}}(\boldsymbol{h}, k') \leqslant 1-\alpha\}. \tag{5.14}$$

Its conformal version, *Regularized Adaptive Prediction Sets (C-RAPS)* (Angelopoulos et al., 2021), applies the split conformal algorithm to the same score. The conformity score sorts marks by their predicted probabilities $\hat{p}_{k|X=\boldsymbol{h}}(k)$ and includes regularization terms to produce smaller sets:

$$s_{\text{RAPS}}(\boldsymbol{h}, k) = \sum_{k':\hat{p}_{k'|X=\boldsymbol{h}}(k) \geqslant \hat{p}_{k|X=\boldsymbol{h}}(k)} \hat{p}_{k|X=\boldsymbol{h}}(k') + U \cdot \hat{p}_{k|X=\boldsymbol{h}}(k) + \gamma \left(o(k) - k_{\text{reg}}\right)^+, \tag{5.15}$$

where $U \sim \mathcal{U}(0,1)$ handles ties, $o(k)$ is the rank of mark k, and $\gamma, k_{\text{reg}} \geqslant 0$ are regularization hyperparameters that penalize large sets.

5.7.3 Joint Prediction Sets for Arrival Time and Mark

Our primary objective is to construct a joint prediction set $\hat{R}(\boldsymbol{h}) \subseteq \mathcal{Y}$ for the pair (τ, k) with a marginal coverage guarantee of at least $1-\alpha$. We explore two main strategies to achieve this.

Combining Individual Regions (Baseline). A simple and valid approach is to combine the individual conformal prediction sets using a Bonferroni correction. We construct a C-QRL region for the time and a C-RAPS region for the mark, each with a miscoverage level of $\alpha/2$. The joint region is their Cartesian product:

$$\hat{R}(\boldsymbol{h}) = \hat{R}_{\tau,\text{C-QRL}}(\boldsymbol{h}) \times \hat{R}_{k,\text{C-RAPS}}(\boldsymbol{h}). \tag{5.16}$$

This method is guaranteed to achieve the desired joint coverage. However, as shown in Figure 5.7a, it produces a rigid rectangular region that ignores any correlation between the time and the mark, which can be inefficient. The prediction interval for the time is forced to be identical for every mark included in the prediction set.

Highest Density Regions. To create more adaptive and potentially tighter regions, we can work directly with the joint density $\hat{f}(\tau, k|\boldsymbol{h})$. The heuristic approach, *Heuristic HDR (H-HDR)*, constructs a Highest Density Region by finding a threshold $t_{1-\alpha}$ on the density values:

$$\hat{R}_{\text{H-HDR}}(\boldsymbol{h}) = \{(\tau, k) \mid \hat{f}(\tau, k \mid \boldsymbol{h}) \geqslant t_{1-\alpha}\}, \tag{5.17}$$

where $t_{1-\alpha}$ is chosen such that the region's probability mass under $\hat{f}$ is $1-\alpha$. To obtain a finite-sample guarantee, we use the *Conformal Highest Density Region (C-HDR)* method from Section 5.2, which calibrates this threshold. It uses the conformity score:

$$s_{\text{C-HDR}}(\boldsymbol{h}, (\tau, k)) = \text{HPD}_{\hat{f}_{Y|X=\boldsymbol{h}}((\tau,k))} = \mathbb{P}(\hat{f}(\hat{\tau}, \hat{k} \mid \boldsymbol{h}) \geqslant \hat{f}(\tau, k \mid \boldsymbol{h})), \tag{5.18}$$

where $(\hat{\tau}, \hat{k}) \sim \hat{f}(\cdot, \cdot \mid \boldsymbol{h})$. As illustrated in Figure 5.7b, this method produces a prediction set based on the joint density, allowing the prediction interval for time to vary for each mark and excluding unlikely combinations altogether.

5.7.4 Experiments

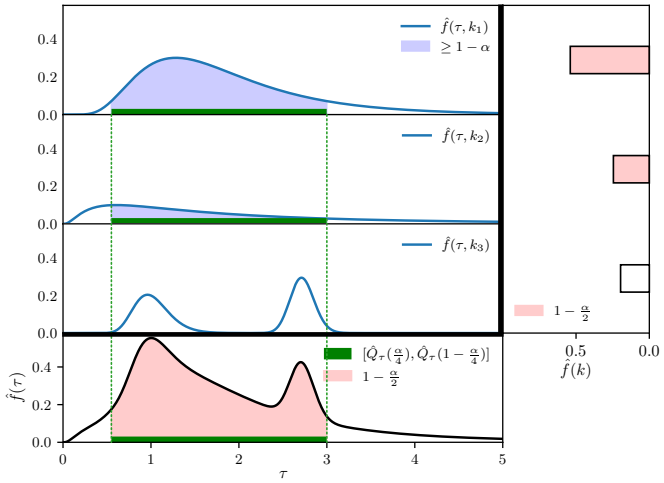
We now empirically evaluate the performance of the joint prediction methods on continuous-time event data. Our goal is to assess both the coverage and efficiency (producing small regions) of the different approaches.

We use several real-world event sequence datasets, including *LastFM*, *MOOC*, *Reddit*, *Retweets*, and *Stack Overflow*, along with synthetic data generated from a Hawkes process. A detailed summary of these datasets, along with a full description of the experimental setup and evaluation metrics, is provided in the appendix. While the appendix also presents a thorough analysis of marginal prediction sets for arrival times and marks separately, here we focus on the primary goal: constructing and evaluating joint prediction sets.

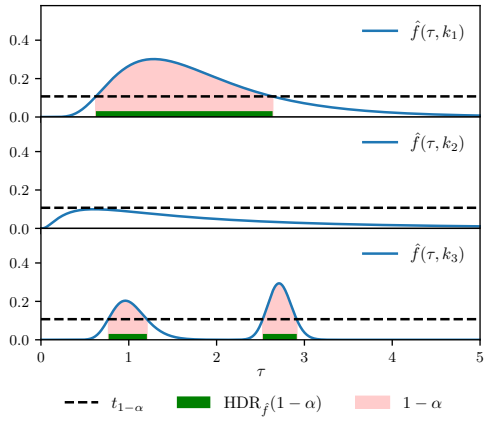
Joint Prediction Sets for Arrival Time and Mark

Figure 5.8 presents the results for methods generating bivariate prediction sets for the arrival time and mark, with a target coverage of 80% ($\alpha = 0.2$).

As expected, the first row shows that all conformal methods (C-QRL-RAPS and C-HDR) successfully achieve the target marginal coverage (MC). In contrast, their heuristic counterparts (H-QRL-RAPS and H-HDR) consistently undercover, highlighting the necessity of the conformalization step to ensure coverage.



(a) The joint region from combining marginal regions is rectangular. Its coverage is at least $1-\alpha$ if each marginal region has $1-\alpha/2$ coverage.



(b) The joint HDR adapts to the joint density, producing variable-length intervals for each included mark (e.g., k_1, k_3) and excluding unlikely marks (e.g., k_2).

Figure 5.7: Example of joint bivariate prediction sets with $\alpha = 0.4$ on a synthetic example with $\tau \in \mathbb{R}^+$ and marks $[K] = \{k_1, k_2, k_3\}$.

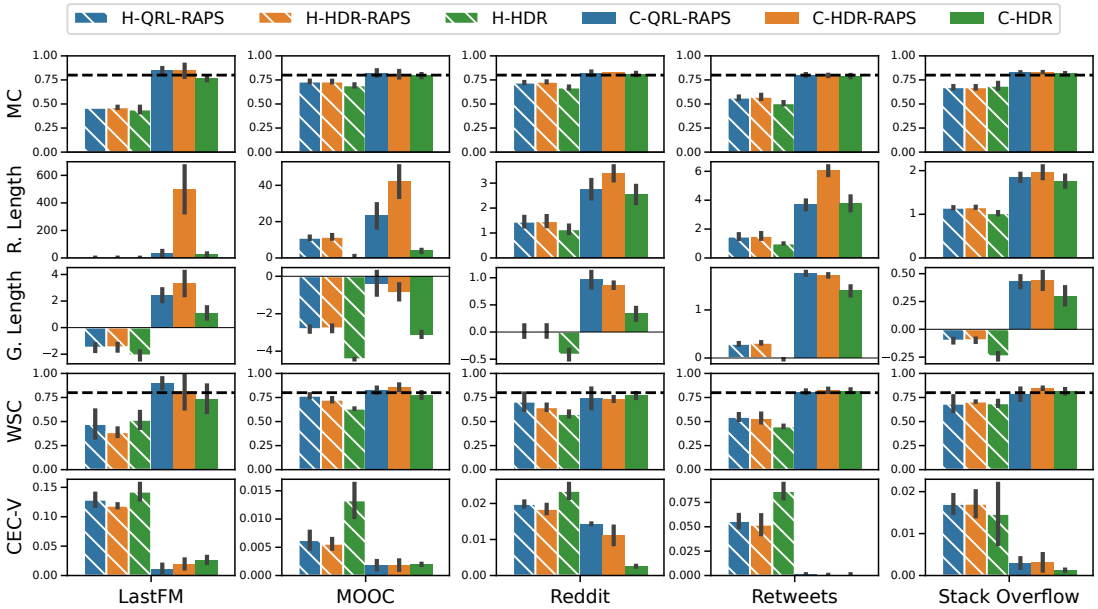


Figure 5.8: Performance of different methods producing a joint region for the time and mark on real-world datasets using the CLNM model. Heuristic methods are hatched.

The second and third rows evaluate the prediction set length (or volume). For the second row, we define the average prediction set length of a method m over the test dataset as $\bar{L}_m$. Then, for a better comparison, when comparing a set of M methods with average lengths $L_1, \ldots, L_M$, we

report the *relative length* of the ith method as

$$\text{R. Length} = \frac{L_i}{\min_{j \in \{1,\dots,M\}} L_j}. \tag{5.19}$$

For the third row, we consider the *geometric mean* of the lengths computed on $\mathcal{D}_{\text{test}}$ as:

$$\text{G. Length} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{i=1}^{|\mathcal{D}_{\text{test}}|} \log(|\hat{R}_{\boldsymbol{y}}(\boldsymbol{h}_i)| + \epsilon), \tag{5.20}$$

where ϵ is an offset that we fix at $\epsilon = 0.01$ to avoid values of $-\infty$ when $|\hat{R}_{\boldsymbol{y}}(\boldsymbol{h}_i)| = 0$.

While the simple average length (second row) can be sensitive to outliers with highly skewed distributions, the geometric mean length (third row) provides a more robust comparison. On this metric, C-HDR produces the smallest regions across all datasets. This result empirically validates the benefit of modeling the joint dependency, as C-HDR is less conservative than the C-QRL-RAPS baseline, which combines marginals.

Finally, the last two rows assess approximate conditional coverage using either WSC or CEC-V, which are described in Section E.6.6. C-HDR again performs competitively, often achieving the best conditional calibration among the valid methods, particularly on the Reddit dataset.

To provide a qualitative illustration, Figure 5.9 shows example prediction sets for a test instance from the LastFM dataset. The figure clearly shows the structural difference between the approaches. C-QRL-RAPS produces a rigid rectangular region, with a constant-width time interval for all selected marks. In contrast, C-HDR creates a more nuanced region that adapts to the joint density, yielding variable-length intervals for each mark and capturing the underlying dependency structure.

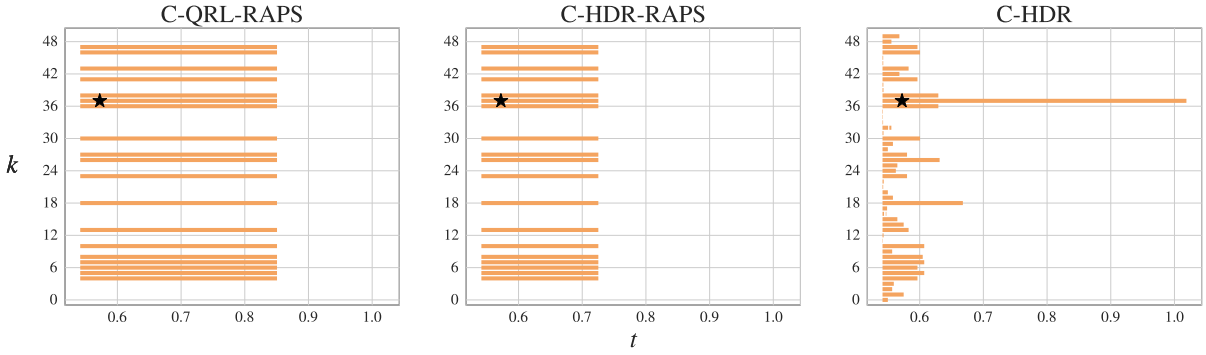


Figure 5.9: Examples of prediction sets generated by CLNM using the C-QRL-RAPS and C-HDR methods for the last event of a test sequence of the LastFM dataset. The black star corresponds to the actual event that materializes.

Empirical Coverage for Different Coverage Levels

To ensure our findings are not specific to a single miscoverage level, we also evaluate the empirical marginal coverage across a range of target coverage levels from 10% to 90%. Figure 5.10 shows these results for the joint prediction methods.

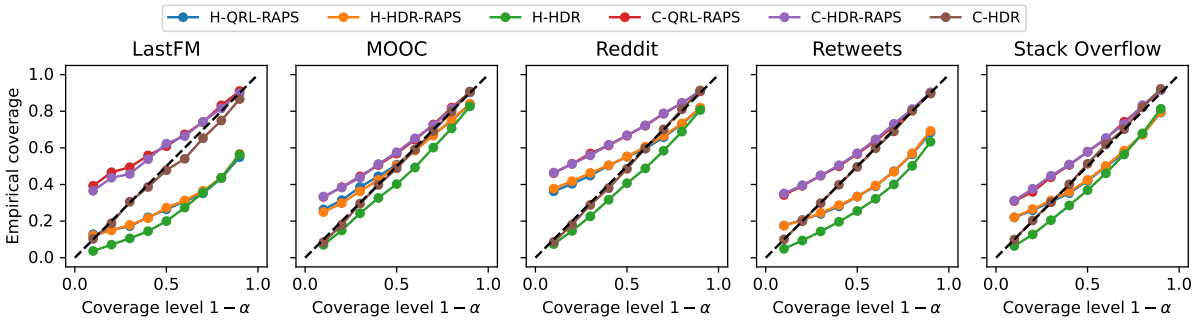


Figure 5.10: Empirical marginal coverage for different coverage levels with the CLNM model. All conformal methods achieve marginal coverage, but the naive method (C-QRL-RAPS) tends to overcover. The heuristic methods do not achieve coverage in most cases.

The plots confirm that the heuristic methods consistently fail to achieve the nominal coverage, with the gap widening as the target coverage increases. All conformal methods satisfy the coverage guarantee, as their empirical coverage lies at or above the diagonal identity line. However, the plot also reveals the conservativeness of the Bonferroni-based C-QRL-RAPS method, which tends to significantly overcover, especially at lower coverage levels. In contrast, C-HDR provides much tighter coverage, staying closer to the nominal level across the entire spectrum. This indicates that it is not only valid but also more statistically efficient than the baseline approach.

5.7.5 Discussion

This application highlights two key findings. First, conformalization is essential for reliable uncertainty quantification with neural marked temporal point processes; heuristic methods consistently undercovered, while their conformal counterparts provided valid coverage guarantees. Second, modeling dependencies is essential for statistical efficiency. By leveraging the joint density, the C-HDR method produced significantly tighter and more adaptive prediction sets than the conservative baseline that combines marginals, confirming the clear advantage of a joint prediction strategy for mixed-type outputs.

5.8. Application to Image Data

To better understand the behavior of prediction sets in high-dimensional spaces, we apply conformal methods to the CIFAR-10 dataset (Krizhevsky et al., 2014), which consists of 32x32 RGB images, each labeled with one of 10 possible classes. We train a generative model conditioned on the image label, where $\mathcal{Y} = [0,1]^{3\times32\times32}$ ($d = 3072$) represents the image space, and $\mathcal{X} = \{0,\dots,9\}$ ($p = 1$) represents the labels. The training, calibration, and test datasets contain 50,000, 1,500, and 1,500 images, respectively. As noted in Angelopoulos and Bates (2023), this calibration dataset size is sufficient to ensure good marginal coverage.

Our generative model is a conditional Glow model (Kingma and Dhariwal, 2018) based on the implementation from Stimper et al. (2022) using a 3-level multi-scale architecture with 32 blocks per level. Like MQF^2 (Section E.6.2), this generative model is a normalizing flow and directly

compatible with all methods presented, except `M-CP`. For a direct comparison with `M-CP`, we compute quantiles based on samples from the generative model as in Section E.6.2.

The latent space of the conditional Glow model, due to its multi-scale architecture, consists of three subspaces: $\mathcal{Z} = \mathcal{Z}_1 \times \mathcal{Z}_2 \times \mathcal{Z}_3$, where $\mathcal{Z}_1 = \mathbb{R}^{48\times4\times4}$, $\mathcal{Z}_2 = \mathbb{R}^{12\times8\times8}$, and $\mathcal{Z}_3 = \mathbb{R}^{6\times16\times16}$. As the distance function $\rho_{\mathcal{Z}}$ in the latent space, we use the maximum norm across the three spaces to penalize high norms in any of them: $\rho_{\mathcal{Z}}(z) = \max\{\|z_1\|, \|z_2\|, \|z_3\|\}$, where $z = z_1 \times z_2 \times z_3$.

Table 5.2 presents the coverage metrics introduced in Section E.6.4. All methods achieve marginal coverage despite the high dimensionality of $\mathcal{Y}$, which is expected as the marginal coverage distribution conditional on the calibration dataset is independent of d (Section E.5.1). Regarding conditional coverage, as in other experiments, `L-CP`, `C-HDR`, `C-PCP`, and `M-CP` exhibit the smallest CEC-X and CEC-Z values, indicating superior conditional coverage. The `WSC` metric supports similar conclusions, with `DR-CP` and `PCP` being the least calibrated.

Table 5.3 presents other metrics. The median of the logarithm of the set size is the smallest for `C-HDR` and `DR-CP`, which matches results on tabular datasets. The mean set size is not reported because it sometimes is infinite with machine precision. Instead, we report the mean of the logarithm of the set size, with similar conclusions to the median set size. M-CP is the fastest, followed by DR-CP and L-CP.

Table 5.2: Results obtained with a conditional Glow model on CIFAR-10 with $1 - \alpha = 0.9$.

Method	MC	CEC-X ($\times100$)	CEC-Z ($\times100$)	WSC
M-CP	$\mathbf{0.900_{0.0035}}$	$0.111_{0.028}$	$0.201_{0.033}$	$0.855_{0.012}$
DR-CP	$\mathbf{0.903_{0.0042}}$	$0.152_{0.030}$	$0.325_{0.033}$	$0.861_{0.0064}$
C-HDR	$\mathbf{0.902_{0.0041}}$	$\mathbf{0.0533_{0.010}}$	$\mathbf{0.0629_{0.020}}$	$\mathbf{0.903_{0.0030}}$
PCP	$\mathbf{0.899_{0.0038}}$	$0.342_{0.070}$	$0.195_{0.030}$	$0.825_{0.0098}$
HD-PCP	$\mathbf{0.899_{0.0038}}$	$0.359_{0.075}$	$0.198_{0.030}$	$0.819_{0.0098}$
STDQR	$\mathbf{0.898_{0.0046}}$	$0.357_{0.071}$	$0.205_{0.033}$	$0.828_{0.015}$
C-PCP	$\mathbf{0.900_{0.0036}}$	$0.118_{0.021}$	$\mathbf{0.0877_{0.023}}$	$0.880_{0.0067}$
L-CP	$\mathbf{0.900_{0.0033}}$	$\mathbf{0.0668_{0.0086}}$	$0.190_{0.027}$	$0.877_{0.011}$

Table 5.3: Results obtained with a conditional Glow model on CIFAR-10 with $1 - \alpha = 0.9$.

Method	Median Log Size	Mean Log Size	Time (s)
M-CP	$-7.10\text{e}{+03}_{5.7}$	$-7.05\text{e}{+03}_{1.3\text{e}+01}$	$\mathbf{0.465_{0.16}}$
DR-CP	$\mathbf{-8.30\text{e}{+03}_{1.4\text{e}+01}}$	$-8.33\text{e}{+03}_{1.4\text{e}+01}$	$47.0_{1.7\text{e}+01}$
C-HDR	$\mathbf{-8.33\text{e}{+03}_{1.9\text{e}+01}}$	$\mathbf{-8.40\text{e}{+03}_{1.9\text{e}+01}}$	$453_{8.7\text{e}+01}$
PCP	$-7.11\text{e}{+03}_{4.5}$	$-7.06\text{e}{+03}_{1.3\text{e}+01}$	$203_{3.5\text{e}+01}$
HD-PCP	$-7.12\text{e}{+03}_{4.8}$	$-7.06\text{e}{+03}_{1.2\text{e}+01}$	$406_{6.9\text{e}+01}$
STDQR	$-7.11\text{e}{+03}_{4.6}$	$-7.06\text{e}{+03}_{1.2\text{e}+01}$	$204_{3.5\text{e}+01}$
C-PCP	$-7.08\text{e}{+03}_{4.9}$	$-7.04\text{e}{+03}_{1.1\text{e}+01}$	$408_{6.9\text{e}+01}$
L-CP	$-7.19\text{e}{+03}_{7.0}$	$-7.15\text{e}{+03}_{1.1\text{e}+01}$	$47.4_{1.8\text{e}+01}$

5.9. Conclusion

We investigated the construction of conformal prediction sets for multi-output regression, a topic that remains relatively underexplored compared to the single-output setting. In particular, we presented a unified comparative study of several conformal methods along with their associated conformity scores, highlighting their properties and interconnections. In addition, we introduced two new classes of conformity scores: CDF-based scores, including a variant compatible with generative models, and latent-based methods, which exploit invertible generative models for improved computational efficiency. Both classes generalize existing conformity scores from the single-output setting.

The choice of conformity score directly influences the geometry and flexibility of the resulting prediction sets. In the single-output setting, the most flexible regions are typically unions of intervals. In contrast, the multivariate case allows for a wider variety of geometries, ranging from hyperrectangles and ellipsoids to highly flexible, nonconvex regions that can be disconnected and capture distributional bimodality. A simple and computationally efficient approach is to construct separate univariate prediction sets for each output dimension and apply a correction for joint coverage. However, these methods do not capture dependencies between output dimensions and typically result in rigid, (unions of) hyperrectangular regions with limited flexibility. In contrast, more flexible methods account for correlations and dependencies across outputs by incorporating the covariance structure, modeling the joint density, or leveraging generative models. These approaches produce more expressive prediction sets but are generally more computationally demanding.

While conformal prediction (CP) always guarantees marginal coverage, conformity scores whose thresholds do not vary instance-wise fail to achieve the desirable property of asymptotic conditional coverage (ACC). In contrast, our proposed scores enable ACC but require estimating the conditional distribution of the conformity score—an inherently challenging task in low-data regimes. Similarly, CP methods based on generative models introduce additional sampling variability. Finally, our large-scale empirical study systematically compares these conformal methods across multiple multi-output regression datasets, using various evaluation metrics, including conditional coverage and prediction set volume.

Limitations. First, the proposed conformity scores have distinct limitations. CDF-based methods require sampling, which is computationally demanding; moreover, PCP becomes less efficient as the output dimension d grows due to the curse of dimensionality, and C-PCP inherits this weakness. Latent-based methods assume access to an invertible conditional generative model, which restricts modeling choices.

Second, ACC is an important but not sufficient criterion. It is asymptotic and distribution-level, and does not by itself guarantee small region volume or finite-sample conditional coverage. Nonetheless, because several methods in our comparison are not ACC, this notion remains a relevant point of reference.

Third, our empirical evaluation emphasizes tabular data and temporal point processes, and only briefly touches image data. Our approach could be extended to other semi-structured and unstructured modalities, including text, graphs, and multi-modal data.

Fourth, conformal prediction yields prediction sets, which may be insufficient for tasks that require full predictive distributions. In Chapter 7, we complement region construction with a recalibrated multivariate distribution equipped with an explicit density function and conformal coverage guarantees.

CHAPTER 6

Rectifying Conformity Scores

This chapter is based on the following paper:

> Vincent Plassier*, Alexander Fishkov*, **Victor Dheur***, Mohsen Guizani, Souhaib Ben Taieb, Maxim Panov, and Eric Moulines (2025). Rectifying Conformity Scores for Better Conditional Coverage. *The 42nd International Conference on Machine Learning.*

Vincent Plassier developed the theoretical framework and established the associated guarantees, while Alexander Fishkov and Victor Dheur carried out the experimental studies. The research was conducted under the supervision of Eric Moulines, Maxim Panov, and Souhaib Ben Taieb. All authors contributed to the preparation of the manuscript.

6.1. Introduction

In the previous chapter, we conducted a comprehensive study of multi-output conformal regression, introducing conformity scores designed to produce reliable and sharp prediction sets, either without assumptions on the underlying generative model, or with low computational cost. Our analysis in Section 5.5.2 and the empirical results in Section 5.6 showed the importance of achieving asymptotic conditional coverage under the oracle distribution. Methods that satisfied this property, such as C-HDR and our proposed L-CP and C-PCP scores, consistently produced regions with better empirical conditional coverage.

However, a first limitation of these methods is their reliance on either a full predictive density (like C-HDR) or an invertible generative model (like L-CP), which is not always available. Moreover, estimating a full conditional distribution, especially for high-dimensional outputs, can be difficult, especially with little data or complex output distributions. This observation motivates a fundamental question: can we improve conditional coverage without bearing the full cost of conditional density estimation?

*Equal contribution

This chapter introduces a novel method, rectified conformal prediction (RCP), designed precisely to address this challenge. Instead of modeling the entire conditional distribution of the output Y, RCP refines the conformity scores themselves by learning to adjust them based on input features. This extends normalized nonconformity scores; see, e.g., Papadopoulos et al. (2008) and Papadopoulos and Haralambous (2011). By focusing on estimating only the conditional quantile of a univariate conformity score, a much simpler task than full density estimation, RCP produces prediction sets that are adaptive to local data structures while preserving the essential marginal coverage guarantee of conformal prediction.

The main contributions of this work can be summarized as follows.

- We introduce rectified conformal prediction (RCP), a new conformal method designed to enhance conditional validity by refining conformity scores (see Sections 6.3 and 6.4). The proposed method avoids the need to estimate the full conditional distribution of a multivariate response, relying instead on estimating only the conditional quantile of a univariate conformity score.
- We provide a theoretical lower bound on the conditional coverage of the prediction sets generated by RCP (see Section 6.6). This conditional coverage is explicitly governed by the approximation error in estimating the conditional quantile of the conformity score distribution.
- We evaluate our method on several benchmark datasets and compare it against state-of-the-art alternatives[1] (see Section 6.7). Our results demonstrate improved performance, particularly in terms of conditional coverage metrics such as worst slab coverage (Cauchois et al., 2021) and conditional coverage error (Dheur et al., 2024).

6.2. Background

Construction of prediction sets for regression problems is often based on distributional regression that focuses on fully characterizing the conditional distribution of a response variable given a covariate (N. Klein, 2024). This approach improves uncertainty quantification and decision-making (Berger and Smith, 2019). From the conditional predictive distribution, prediction sets can be derived to capture values likely to occur with a given probability. However, these regions rely heavily on the predictive model's quality, and poorly estimated models can result in unreliable predictions.

Towards conditional validity of CP methods. In many applications, conditional validity is a natural requirement, i.e., for all $x \in \mathcal{X}$,

$$\mathbb{P}(Y \in \hat{R}(X) \mid X = x) \geqslant 1 - \alpha. \tag{6.1}$$

Conditional coverage (6.1) is stronger and implies marginal coverage. While classical conformal methods provide marginal validity, they do not ensure conditional validity.

Let us denote the conditional distribution $\mathrm{P}_{\mathbf{s}|X=x}$ with $\mathbf{s}$ being a shorthand for $s(X, Y)$. The

[1] `https://github.com/stat-ml/rcp`

following oracle prediction set

$$\hat{R}(x) = \left\{y \in \mathcal{Y} \colon s(x,y) \leqslant Q_{\mathbf{s}|X=x}(1-\alpha)\right\} \tag{6.2}$$

satisfies conditional coverage (6.1) by the definition of conditional quantile $Q_{\mathbf{s}|X=x}(1-\alpha)$ (2.8). However, exact conditional validity is not achievable within the conformal prediction framework (Vovk, 2012; Lei and Wasserman, 2014; Foygel Barber et al., 2021b). In what follows we will present a new conformal prediction method that will achieve *approximate* conditional validity while satisfying exact marginal guarantees.

6.3. Rectified Conformal Prediction

The primary objective of our *rectified conformal prediction* (RCP) method is to enhance the conditional coverage of any given conformity score while maintaining their exact marginal validity. Equation (6.2) suggests that one could approximate the $(1-\alpha)$-quantile of the conditional distribution of the scores to construct the prediction set:

$$\hat{R}(x) = \{y \in \mathcal{Y} \colon s(x,y) \leqslant \hat{Q}_{\mathbf{s}|X=x}(1-\alpha)\}.$$

This prediction set provides approximate conditional coverage that depends on the accuracy of the quantile estimator. However, it fails to ensure exact marginal coverage which is an essential property for conformal prediction methods.

A motivation for RCP. Our RCP method is specifically designed to achieve both exact conformal marginal validity and approximate conditional coverage. To achieve this, RCP first constructs specially transformed (rectified) scores to enhance conditional coverage. To construct the rectified scores, it builds on the key observation that *marginal* and *conditional* coverage coincide precisely when the conditional $(1-\alpha)$-quantile of the conformity score is independent of the covariates. RCP then applies the SCP procedure to these rectified scores, ensuring the classical exact conformal marginal validity.

For any given score $s(x,y)$, referred to as the basic score, RCP computes a rectified score $\tilde{s}(x,y)$, which is a transformation of the basic score that satisfies, for P_X-a.e. $x \in \mathcal{X}$,

$$Q_{\tilde{\mathbf{s}}}(1-\alpha) = Q_{\tilde{\mathbf{s}}|X=x}(1-\alpha). \tag{6.3}$$

Below we present two examples that show how one can construct the rectified scores satisfying (6.3).

Example 4. Consider the rectified score $\tilde{s}(x,y) = s(x,y)/Q_{\mathbf{s}|X=x}(1-\alpha)$, with the assumption that $Q_{\mathbf{s}|X=x}(1-\alpha) > 0$ for any $x \in \mathcal{X}$. We can define the following prediction set, equivalent to (6.2): $\hat{R}(x) = \{y \in \mathcal{Y} \colon \tilde{s}(x,y) \leqslant 1\}$. This prediction set satisfies conditional coverage. Furthermore, in Appendix G.3.1, we prove that this rectified score satisfies the equality in (6.3).

Example 5. Consider the rectified score $\tilde{s}(x,y) = s(x,y) - Q_{\mathbf{s}|X=x}(1-\alpha)$. The corresponding prediction set, also equivalent to (6.2), is: $\hat{R}(x) = \{y \in \mathcal{Y} \colon \tilde{s}(x,y) \leqslant 0\}$, and it satisfies conditional coverage. Furthermore, in Appendix G.3.2, we prove that this rectified score satisfies the equality in (6.3).

In the following, we generalize over these two basic examples and present a rich family of general score transformations that allow for score rectification.

RCP with general transformations. Recall that starting from a basic score function $s(x, y)$, we develop a transformed score $\tilde{s}(x, y)$ to achieve conditional validity at a given confidence level α. To do so, we introduce a transformation to rectify the basic conformity score s.

Consider a parametric family $\{f_t\}_{t\in\mathbb{T}}$ with $(t, v) \in \mathbb{T} \times \mathbb{R} \mapsto f_t(v) \in \mathbb{R}$ and $\mathbb{T} \subseteq \mathbb{R}$. For convenience, we define $\tilde{f}_v(t) = f_t(v)$ and proceed under the following assumption.

***H*2.** The function $v \in \mathbb{R} \cup \{\infty\} \mapsto f_t(v)$ is increasing for any $t \in \mathbb{T}$. There exists $\varphi \in \mathbb{R}$ such that $\tilde{f}_\varphi^{-1}$ is continuous, increasing, and surjective on $\mathbb{R}$.

Under **H**2, we denote by $\tilde{f}_\varphi^{-1}$ the inverse of the function $\tilde{f}_\varphi$, i.e., $\tilde{f}_\varphi^{-1} \circ \tilde{f}_\varphi(t) = t$, for all $t \in \mathbb{T}$. Let $\varphi \in \mathbb{R}$ be such that $\tilde{f}_\varphi$ is invertible (see **H**2). Set

$$s_\varphi(x, y) = \tilde{f}_\varphi^{-1}\big(s(x, y)\big) \tag{6.4}$$

and denote $\mathbf{s} = s(X, Y)$, and $\mathbf{s}_\varphi = s_\varphi(X, Y)$. We now define the following prediction set

$$\hat{R}(x) = \big\{y \in \mathcal{Y} \colon s(x, y) \leqslant f_{\tau_\star(x)}(\varphi)\big\}, \tag{6.5}$$

where

$$\tau_\star(x) = Q_{\mathbf{s}_\varphi|X=x}(1-\alpha) = \tilde{f}_\varphi^{-1}\big(Q_{\mathbf{s}|X=x}(1-\alpha)\big), \tag{6.6}$$

i.e., the $(1-\alpha)$ conditional quantile of the transformed score $\mathbf{s}_\varphi$ given $X = x$. We retrieve Example 4 with $f_t(v) = vt$, $\varphi = 1$. In this case $\tilde{f}_1^{-1}(t) = t$ and $s_{\varphi=1}(x, y) = s(x, y)$. Similarly, for Example 5, $f_t(v) = v + t$, $\varphi = 0$. In such a case, $\tilde{f}_0^{-1}(t) = t$ and $s_{\varphi=0}(x, y) = s(x, y)$.

In the following, we show that the prediction set in (6.5) satisfies the conditional validity guarantee in (6.1) and, subsequently, the marginal coverage guarantee in (2.55). In fact, we can write

$$\begin{aligned}
\mathbb{P}(Y \in \hat{R}(X) \mid X = x) &= \mathbb{P}(\mathbf{s} \leqslant f_{\tau_\star(X)}(\varphi) \mid X = x) \\
&\overset{(a)}{=} \mathbb{P}(\mathbf{s} \leqslant \tilde{f}_\varphi(\tau_\star(X)) \mid X = x) \\
&\overset{(b)}{=} \mathbb{P}(\mathbf{s}_\varphi \leqslant \tau_\star(X) \mid X = x) \overset{(c)}{\geqslant} 1 - \alpha,
\end{aligned}$$

where we have used in (a) that $\tilde{f}_v(t) = f_t(v)$, in (b) that $\tilde{f}_\varphi$ is invertible and the definition of $\mathbf{s}_\varphi$, and in (c) the definition of $\tau_\star(x)$. We may rewrite the prediction set (6.5) in terms of the rectified score $\tilde{s}_\star(x, y) = f_{\tau_\star(x)}^{-1}\big(s(x, y)\big)$:

$$\hat{R}(x) = \big\{y \in \mathcal{Y} \colon \tilde{s}_\star(x, y) \leqslant \varphi\big\}. \tag{6.7}$$

In Section G.3.3, we establish that the rectified score satisfies (6.3), more precisely, setting $\tilde{\mathbf{s}}_\star = \tilde{s}_\star(X, Y)$, for all $x \in \mathcal{X}$,

$$\varphi = Q_{\tilde{\mathbf{s}}_\star|X=x}(1-\alpha) = Q_{\tilde{\mathbf{s}}_\star}(1-\alpha). \tag{6.8}$$

With the rectified score, conditional and unconditional coverage coincide. However, while the oracle prediction set in (6.5) provides both conditional and marginal validity, it requires the precise knowledge of the pointwise quantile function $\tau_\star(x)$. In practice, $\tau_\star(x)$ is not known and one must construct an estimate $\hat{\tau}(x)$ using some hold out dataset. Below we discuss the resulting data-driven procedure.

Algorithm 9 The RCP algorithm

Input: Calibration dataset $\mathcal{D}_{\text{cal}}$, miscoverage level α, conformity score function s, transformation function f_t, test input x.
▷ **Calibration Stage**
Split $\mathcal{D}_{\text{cal}}$ into $\mathcal{D}_{cp} = \{(X^{(k)}, Y^{(k)})\}_{k=1}^{n}$ and $\mathcal{D}_{\tau} = \{(X'^{(k)}, s(X'^{(k)}, Y'^{(k)}))\}_{k=1}^{m}$.
$\hat{Q}_{\mathbf{s}|X}(1-\alpha) \leftarrow$ conditional quantile estimate on $\mathcal{D}_{\tau}$.
Define the estimator $\hat{\tau}(x) = \tilde{f}_{\varphi}^{-1}\big(\hat{Q}_{\mathbf{s}|X=x}(1-\alpha)\big)$.
for $k = 1$ **to** n **do**
  $\tilde{\mathbf{s}}^{(k)} \leftarrow f_{\hat{\tau}(X^{(k)})}^{-1}\big(s(X^{(k)}, Y^{(k)})\big)$.
end for
$k_\alpha \leftarrow \lceil (1-\alpha)(n+1) \rceil$.
$\hat{q} \leftarrow k_\alpha$-th smallest value in $\{\tilde{\mathbf{s}}^{(k)}\}_{k\in[n]} \cup \{+\infty\}$.
▷ **Test Stage**
  $\hat{R}(x) \leftarrow \big\{y \in \mathcal{Y} \colon f_{\hat{\tau}(x)}^{-1}\big(s(x,y)\big) \leqslant \hat{q}\big\}$.
Output: $\hat{R}(x)$.

6.4. Implementation of RCP

The RCP algorithm. The RCP approach, as discussed above, requires a basic conformity score function s, a transformation function f_t, and a calibration dataset of $n = n + m$ points. A critical step in the RCP algorithm is estimating the conditional quantile $\hat{\tau}(x) \approx Q_{\mathbf{s}_\varphi|X=x}(1-\alpha)$, which we discuss in detail below. $\hat{\tau}$ is learned on a separate part of calibration dataset composed of m data points $\{(X'^{(k)}, Y'^{(k)}) \colon k = 1, \ldots, m\}$. Subsequently, RCP uses SCP with the rectified scores $\tilde{s}(x,y) := f_{\hat{\tau}(x)}^{-1}\big(s(x,y)\big)$ instead of the basic scores $s(x,y)$. SCP is applied to the rectified scores computed on the second part of the calibration dataset: $\tilde{\mathbf{s}}^{(k)} = \tilde{s}(X^{(k)}, Y^{(k)}), k = 1, \ldots, n$.

Finally, for a given test input x and miscoverage level α, RCP computes the prediction set as

$$\hat{R}(x) = \big\{y \in \mathcal{Y} \colon \tilde{s}(x,y) \leqslant \hat{q}\big\}. \tag{6.9}$$

The resulting RCP procedure is summarized in Algorithm 9. We show exact marginal validity of RCP and give a bound on its approximate conditional coverage in Section 6.6 below.

Estimation of $\tau_\star(x)$. We present below several methods for estimating $\tau_\star(x)$. Interestingly, even coarse approximations of this conditional quantile can significantly improve conditional coverage; see the discussion in Section 6.7.

Quantile regression. For any $x \in \mathbb{R}^d$, the conditional quantile, denoted by $\tau_\star(x)$, is a minimizer of the expected risk with the check function:

$$\tau_\star(x) \in \arg\min_{\tau} \ \mathbb{E}\left[\rho_{1-\alpha}(s_\varphi(X,Y) - \tau(X))\right], \tag{6.10}$$

where the minimum is taken over the function $\tau \colon \mathcal{X} \to \mathbb{R}$ and $\rho_{1-\alpha}$ is the check function (Koenker and Bassett Jr, 1978; Koenker and Hallock, 2001): $\rho_\alpha(u) = u \cdot (\alpha - \mathbb{1}(u < 0))$. In practice, the

empirical quantile function $\hat{\tau}$ is obtained by minimizing the empirical risk:

$$\hat{\tau} \in \arg\min_{\tau \in \mathcal{C}} \frac{1}{m} \sum_{k=1}^{m} \rho_{1-\alpha}\big(s_\varphi(X'^{(k)}, Y'^{(k)}) - \tau(X'^{(k)})\big) + \lambda g(\tau), \tag{6.11}$$

where g is a penalty function and $\mathcal{C}$ is a class of functions. When $\tau(x) = \theta^\top \Phi(x)$ where Φ is a feature map, and g is convex, the optimization problem in (6.11) becomes convex. Theoretical guarantees in this setting, are given, e.g., in C. Chen and Wei (2005) and Koenker (2005).

Non-parametric methods have also been extensively explored, see, e.g., Chernozhukov and Hansen (2005) and Chernozhukov et al. (2022). Takeuchi et al. (2006) introduced the kernel quantile regression (KQR) framework, formulating quantile regression as minimizing the check function loss in an RKHS with Tikhonov (squared-norm) regularization. It established some of the first theoretical guarantees for RKHS-based quantile models, deriving finite-sample generalization error bounds using Rademacher complexity; these results were later improved in Li et al. (2007).

Local quantile regression. The local quantile can be obtained by minimizing the empirical weighted expected value of the check function $\rho_{1-\alpha}$, defined as follows:

$$\hat{\tau}(x) \in \arg\min_{t \in \mathbb{R}} \left\{ \sum_{k=1}^{m} w_k(x) \rho_{1-\alpha}(s_\varphi(X'^{(k)}, Y'^{(k)}) - t) \right\}, \tag{6.12}$$

where $\{w_k(x)\}_{k=1}^m$ are positive weights; see Bhattacharya and Gangopadhyay (1990).

For instance, we can set $w_k(x) = m^{-1} K_{h_X}(\|x - X'^{(k)}\|)$, where for $h > 0$, $K_h(\cdot) = h^{-1} K_1(h^{-1}\cdot)$ is a kernel function satisfying $\int K_1(x)\mathrm{d}x = 1$, $\int x K_1(x)\mathrm{d}x = 0$ and $\int x^2 K_1(x)\mathrm{d}x < \infty$; h_X, the kernel bandwidth is tuned to balance bias and variance. With appropriate adaptive choice of $h(x)$, this approach can be shown to be asymptotically minimax over Hölder balls; see Bhattacharya and Gangopadhyay (1990), Spokoiny et al. (2013), and Reiß et al. (2009). More recently, G. Shen et al. (2024) introduced a penalized non-parametric approach to estimating the quantile regression process (QRP) using deep neural networks with rectifier quadratic unit (ReQU) activations. G. Shen et al. (2024) derives upper bounds on the mean-squared error for quantile regression using deep ReQU networks, depending only on the approximation error and network. The bounds are shown to be tight for broad function classes (e.g., Hölder compositions, Besov spaces), implying that ReQU neural networks achieve minimax-optimal convergence rates for conditional quantile estimation. Notably, the theory requires minimal assumptions and holds even for heavy-tailed error distributions.

6.5. Related Work

It is well known that obtaining exact conditional coverage for all possible inputs within the conformal prediction framework is impossible without making distributional assumptions (Foygel Barber et al., 2021b). However, the literature has proposed various relaxations of exact conditional coverage, focusing on different notions of approximate conditional coverage.

A first class of methods involves group-conditional guarantees (Jung et al., 2023; Ding et al., 2024), which provide coverage guarantees for a predefined set of groups. Another class partitions

the covariate space into multiple regions and applies classical conformal prediction within each region (LeRoy and D. Zhao, 2021; Alaa et al., 2023; Kiyani et al., 2024). The significant limitation of these methods lies in the need to specify the groups or regions in advance.

In the context of multivariate prediction sets, given a predictor $\hat{\mu}(\cdot)$, a natural choice for the conformity score is $s_\infty(x, y) = \|y - \hat{\mu}(x)\|_\infty$, where $\|u\|_\infty = \max_{1 \leqslant i \leqslant d}(|u_i|)$ (Diquigiovanni et al., 2021b). This conformity score measures the prediction error associated with the predictor $\hat{\mu}$ (Nouretdinov et al., 2001; Vovk et al., 2005; Vovk et al., 2009). Setting $f_t(v) = tv$ and $\varphi = 1$, the rectified conformity scores are given by $\tilde{s}_\infty(x, y) = s_\infty(x, y)/\hat{Q}_{\mathbf{s}_\infty|X=x}(1-\alpha)$ where $\hat{Q}_{\mathbf{s}_\infty|X=x}(1-\alpha)$ is an estimate of the conditional quantile $Q_{\mathbf{s}_\infty|X=x}(1-\alpha)$, with $\mathbf{s}_\infty = s_\infty(X, Y)$. Thus, rectified conformal prediction is similar to the approach proposed in Lei et al. (2018), but with a different choice of scaling function.

Methods utilizing conditional density estimation have been proposed to produce conformal prediction intervals that adapt to skewed data (Sesia and Romano, 2021), to minimize the average volume (Sadinle et al., 2019, denoted `DR-CP`) or to define more flexible highest density regions (Izbicki et al., 2022; Plassier et al., 2025a). Probabilistic conformal prediction (PCP; Z. Wang et al., 2023) bypasses density estimation by constructing prediction sets as unions of balls centered on samples from a generative model. All these methods are either tailored to handle the scalar response ($d = 1$) or require an accurate conditional distribution estimate which might be hard to obtain in practical scenarios.

Guan (2023) introduces a localized conformal prediction framework that adapts to data heterogeneity by weighting calibration points based on their similarity to the test sample. To do so, kernel-based localizers assign greater importance to nearby points, tailoring prediction intervals to local data patterns. Amoukou and Brunel (2023) extend Guan's approach by replacing kernels with quantile regression forest estimators for improved performance. Although effective, these methods face challenges in high-dimensional or mixed-variable settings.

Several methods aim to transform conformity scores to improve approximate conditional coverage. For example, Johansson et al. (2021), following earlier works by Papadopoulos and Haralambous (2011), Johansson et al. (2014), and Lei et al. (2018), investigate *normalized conformity scores* (NCF), which enhance standard conformal prediction by adjusting prediction sets according to instance difficulty. NCF can be represented within our framework through a specific choice of the function $f_t(v) = v/(t + \beta)$, where β-values will put a greater emphasis on the difficulty estimation. Notably, the estimation approach employed in these papers uses least-squares regression on residuals, in contrast to the quantile regression approach adopted in rectified conformal prediction, which is essential to satisfy (6.3). Han et al. (2022) presents an approach that uses kernel density estimation to approximate the conditional distribution. Similarly, Deutschmann et al. (2023) rescales the conformity scores based on an estimate of the local score distribution using the jackknife+ technique. However, these methods generally rely on estimating the conditional distribution of conformity scores, which is challenging in practice. Dewolf et al. (2025) studies conditional validity of normalized conformal predictors in oracle setting, i.e., when the optimal normalization is known.

Recent work by Colombo (2024) suggests transforming the conformity score employing a normalizing flow: $\tilde{s}(x, y) = b(s(x, y), x)$. The normalizing flow is trained to map the joint distribution $P_{\mathbf{s},X}$ of the conformity score and attributes into a product distribution, $P_{\tilde{\mathbf{s}}} \otimes P_X$, where $P_{\tilde{\mathbf{s}}}$ is an

arbitrary univariate distribution. Notably, this condition is stricter than the conditional coverage criterion (6.3), as it enforces $P_{\tilde{\mathbf{s}}|X=x} = P_{\tilde{\mathbf{s}}}$ for almost every x under P_X. Consequently, learning such a transformation typically necessitates a larger sample size; see Section 6.7.

One method (Xie et al., 2024) proposes to use a cross-validated boosting procedure to learn a new score function to be used in split conformal prediction. The authors consider a specific family of possible score functions and corresponding loss functions tailored either to deviation from conditional coverage or interval length. This method has several limitations compared to our approach: limited set of score functions, tailored to one-dimensional targets, requires access to the train set, and high computation cost.

The rectified conformal prediction method shares some similarities with that of Gibbs et al. (2025), which also performs a quantile regression of the conformity score w.r.t. the attribute X. There are two essential differences: firstly, Gibbs et al. (2025) work directly with the conformity score s, whereas we regress on a transformed score s_φ. Secondly, the manner in which the quantile regression result is used differs significantly. Rectified conformal prediction uses the quantile estimator to define the rectified score $\tilde{s}$, to which the standard CP procedure is applied, while Gibbs et al. (2025) propose a considerably more complex procedure; see Section 6.7.

6.6. Theoretical Guarantees

In this section, we study the marginal and conditional validity of the prediction sets $\hat{R}(x)$ defined in (6.9). Due to space constraints, we present simplified versions of the results. Full statements and proofs can be found in Plassier et al. (2025b). Many of the results hold independently of the specific method used to construct the conditional quantile estimator $\hat{\tau}(x)$. We impose the following minimal assumption.

***H*3.** For any $x \in \mathcal{X}$, we have $\hat{\tau}(x) \in \mathbb{T}$.

The following theorem establishes the standard conformal guarantee. We stress that for this statement, the definition of $\hat{\tau}(x)$ is not essential. The result is valid for any function $\tau(x)$, and the proof follows from Lemma 7.

Theorem 7. Assume **H**2-**H**3 hold and suppose the rectified conformity scores $\{\tilde{\mathbf{s}}^{(1)}, \dots, \tilde{\mathbf{s}}^{(n)}, \tilde{s}(X,Y)\}$ are almost surely distinct. Then, for any $\alpha \in (0,1)$, it follows

$$1 - \alpha \leqslant \mathbb{P}(Y \in \hat{R}(X)) < 1 - \alpha + \frac{1}{n+1}.$$

We will now examine the conditional validity of the prediction set. To do so, we will explore the relationship between the conditional coverage of $\hat{R}(x)$ and the accuracy of the conditional quantile estimator $\hat{\tau}(x)$. To simplify the statements, we assume that the distribution of $P_{\mathbf{s}_\varphi|X=x}$, where $\mathbf{s}_\varphi = s_\varphi(X,Y)$, is continuous. Define

$$\epsilon_\tau(x) = \mathbb{P}(s_\varphi(X,Y) \leqslant \tau(x) \mid X = x) - 1 + \alpha. \tag{6.13}$$

The function ϵ_τ represents the deviation between the current confidence level w.r.t. τ and the desired level $1-\alpha$. Define the conditional risk

$$\mathcal{R}_x(\tau) = \mathbb{E}[\rho_{1-\alpha}(s_\varphi(X,Y) - \tau(X)) \mid X = x]\,. \tag{6.14}$$

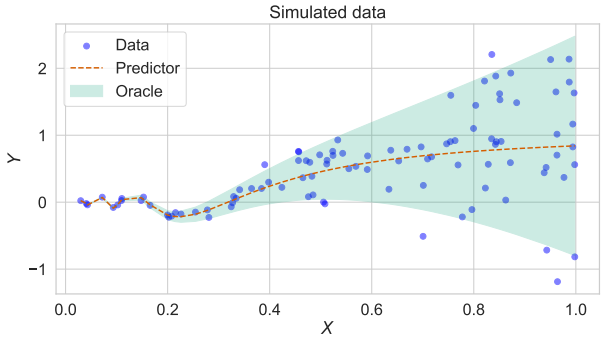


Figure 6.1: Oracle data distribution, sample data and predictor for the toy dataset.

It is shown in Plassier et al. (2025b) that, under weak technical conditions, ϵ_τ satisfies the following property: for all $x \in \mathcal{X}$,

$$|\epsilon_\tau(x)| \leqslant \sqrt{2 \times \{\mathcal{R}_x(\tau(x)) - \mathcal{R}_x(\tau_\star(x))\}},$$

where $\tau_\star(x)$ is defined in (6.6). If $\tau(x)$ is close to the minimizer of the conditional risk $\mathcal{R}_x$ (as defined in (6.14)), then $\epsilon_\tau(x)$ is expected to approach zero.

The CDF of the rectified conformity score is defined as $F_{\tilde{\mathbf{s}}} = \mathbb{P}(\tilde{\mathbf{s}} \leqslant \cdot)$. We denote its conditional version by $F_{\tilde{\mathbf{s}}|X=x} = \mathbb{P}(\tilde{s}(x, Y) \leqslant \cdot \mid X = x)$.

Theorem 8. Assume that **H2**-**H3** and $F_{\tilde{\mathbf{s}}}$ is continuous and that, for any $x \in \mathcal{X}$, $F_{\tilde{\mathbf{s}}|X=x} \circ F_{\tilde{\mathbf{s}}}^{-1}$ is L-Lipschitz. Then, for any $\alpha \in [\{n+1\}^{-1}, 1)$ it holds

$$\mathbb{P}(Y \in \hat{R}(X) \mid X = x) \geqslant 1 - \alpha + \epsilon_{\hat{\tau}}(x) - \alpha \mathrm{L} \times (F_{\tilde{\mathbf{s}}}^{+}(\varphi))^{n+1}. \tag{6.15}$$

The proof is provided in Plassier et al. (2025b). According to Theorem 8, the conditional validity of the prediction set $\hat{R}(x)$ directly depends on the accuracy of the quantile estimator $\hat{\tau}(x)$. If $\hat{\tau}(x)$ closely approximates the conditional quantile $Q_{\mathbf{s}_\varphi|X=x}(1-\alpha)$, then (6.15) ensures that conditional coverage is approximately achieved.

6.7. Experiments

6.7.1 Toy example

Let us consider the following data-generating process:

$$X \sim \mathrm{Beta}(1.2, 0.8), \quad Y \mid X = x \sim \mathcal{N}(\mu(x), x^4).$$

where $\mu(x) = x \sin(x)$. Figure 6.1 shows a realization with $n = 100$ data points. Our goal is to investigate the influence of the quality of the $(1-\alpha)$-quantile estimate $\hat{\tau}$ on performance.

We set $\alpha = 0.1$ and consider the conformity score $s(x, y) = |y - \mu(x)|$. In this case, the $(1-\alpha)$-quantile of $s(x, Y) \mid X = x$ is known and we denote it by $Q_{\mathbf{s}|X=x}(1-\alpha)$. Given $\omega \in [0, 1]$, we set $\hat{\tau}(x) = (1-\omega)Q_{\mathbf{s}|X=x}(1-\alpha) + \omega\epsilon(x)$, where we consider $\epsilon(x) \sim \mathcal{N}(0, x^4)$. We perform 1000

Table 6.1: Local coverage on the adversarially selected 10% of the data, ω corresponds to the level of contamination of the score quantile estimate.

ω	0	1/3	2/3	1
COVERAGE	90 ± 01	84 ± 01	75 ± 03	59 ± 07

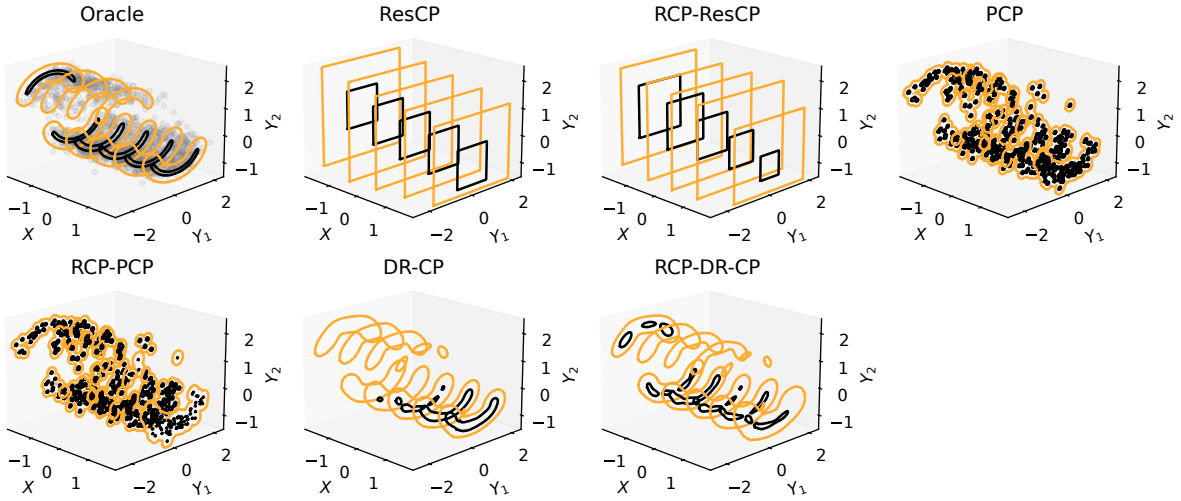


Figure 6.2: Examples of prediction sets on a synthetic dataset where the output has a bivariate and bimodal distribution.

experiments and report the 10% lower value of $x \in [0,1] \mapsto \mathbb{P}(Y \in \hat{R}(x) \mid X = x)$; the results can be found in Table 6.1. If $\omega = 0$, $\hat{\tau}(x)$ corresponds to the true $(1-\alpha)$-quantile. In this case, our method is conditionally valid, as Theorem 8 shows. However, while all settings of ω yield marginally valid prediction sets, the conditional coverage decreases as the quantile estimate $\hat{\tau}(x)$ deteriorates.

6.7.2 Real-world experiment

As in Chapter 5, we use publicly available multi-output regression datasets which are also considered in Tsoumakas et al. (2011), Feldman et al. (2023), and Z. Wang et al. (2023) and only keep datasets with at least 2000 total instances. The characteristics of the datasets are provided in Section A.2.

Base predictors. We consider two base predictors, both parameterized by a fully connected neural network with three layers of 100 units and ReLU activations. The *mean predictor* estimates the mean $\hat{\mu}_i(x)$ of the distribution for each dimension $i \in [d]$ given $x \in \mathcal{X}$. Since it only provides a point estimate, it does not capture uncertainty. The *mixture predictor* is described in Section 2.2.3.

Methods. We compare RCP with four split conformal prediction methods from the literature: `ResCP` (Diquigiovanni et al., 2021b), `PCP` (Z. Wang et al., 2023), `DR-CP` (Sadinle et al., 2019), and

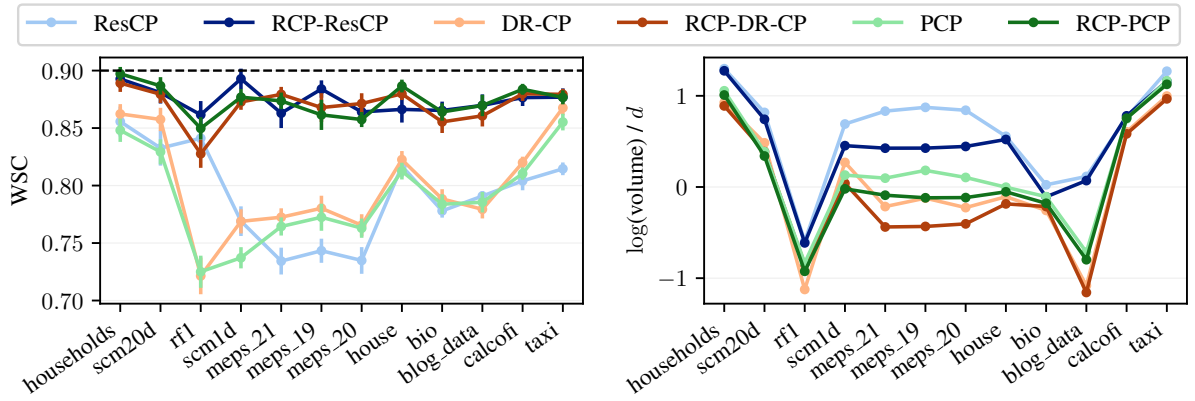


Figure 6.3: Worst-slab coverage and volume for three conformal methods and their RCP counterparts, on datasets sorted by total size.

`SLCP` (Han et al., 2022). `ResCP` uses residuals as conformity scores. To handle multi-dimensional outputs, we follow Diquigiovanni et al. (2021b) and define the conformity score as the l^∞ norm of the residuals across dimensions, i.e., $s(x, y) = \max_{i \in [d]} |\hat{\mu}_i(x) - y_i|$. `PCP` constructs the prediction set as a union of balls, while `DR-CP` defines the prediction set by thresholding the density. `ResCP` is compatible with the *mean predictor*, whereas `PCP` and `DR-CP` are compatible with the *mixture predictor*. Finally, `SLCP`, like RCP, is compatible with any conformity score and base predictor. For RCP, we compute an estimate $\hat{\tau}(x)$ (see Section 6.4) using quantile regression with a fully connected neural network composed of 3 layers with 100 units.

Visualization on a synthetic dataset. Figure 6.2 illustrates example prediction sets for different methods. The orange and black contour lines represent confidence levels of $\alpha = 0.1$ and $\alpha = 0.8$, respectively. The first panel shows the highest density regions of the oracle distribution, while the subsequent panels display prediction sets obtained by different methods, both before and after applying RCP. We can see that combining RCP with `ResCP`, `PCP`, or `DR-CP` results in prediction sets that more closely align with those of the oracle distribution.

Experimental setup. We reserve 2048 points for calibration. The remaining data is split between 70% for training and 30% for testing. The base predictor is trained on the training set, while the baseline conformal methods use the full calibration set to construct prediction sets for the test points. In RCP, the calibration set is further divided into two parts: one for estimating $\hat{\tau}(x)$ and the other as the proper calibration set for obtaining intervals. This ensures that all methods use the same number of points for uncertainty estimation. When not specified, we used the adjustment $f_t(v) = t + v$. Additional details on implementation and hyperparameter tuning are provided in Section G.1.

Evaluation metrics. To evaluate conditional coverage, we use *worst-slab coverage* (WSC, Cauchois et al., 2021; Romano et al., 2019) with $\delta = 0.2$ and the *conditional coverage error*, computed over a partition of $\mathcal{X}$, following Dheur et al. (2025) (Section E.6.6). To evaluate sharpness, we also report the median of the logarithm of the prediction set volume, scaled by the

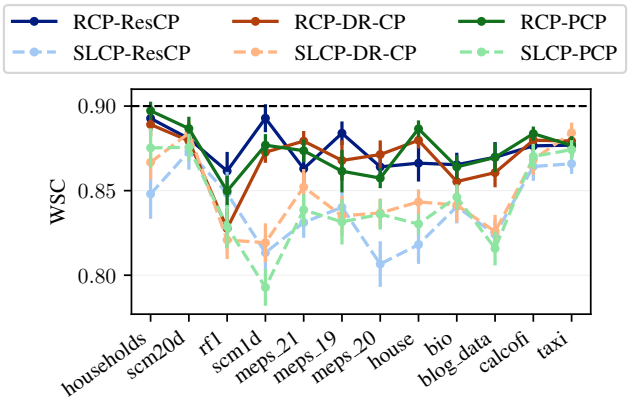


Figure 6.4: Worst-slab coverage for RCP and `SLCP` in combination with different conformity scores, on datasets sorted by total size.

dimension d.

Main results. Figure 6.3 presents the worst-slab coverage and volume for different conformity scores, both with and without RCP. Similarly, Figure 6.4 compares worst-slab coverage between `SLCP` and RCP. Additional results, including conditional coverage error and marginal coverage, are provided in Section G.2.1.

In the left panel of Figure 6.3, we observe that `ResCP`, `PCP`, and `DR-CP` fail to reach the nominal level of conditional coverage for most datasets. In contrast, all variants of RCP significantly improve coverage across all datasets. Similarly, Figure 6.4 shows that RCP often achieves better conditional coverage than `SLCP`, particularly on larger datasets. Figure G.1 in Section G.2.1 confirms these findings with the conditional coverage error. Finally, as expected, all methods achieve marginal coverage.

In the right panel of Figure 6.3, we observe that RCP improves the median prediction set volume compared to non-RCP variants in addition to improving conditional coverage.

Finally, Section G.2.6 compares average and median volumes of prediction sets produced by direct conformal methods and their RCP counterparts. Direct methods obtain a smaller average volume while RCP obtains a smaller median volume.

Additional experiments. We complement main results with multiple experiments aiming at studying variations of the proposed method and comparing it with some additional competitors.

Section G.2.2 discusses the estimation of $\hat{\tau}(x)$ using either a neural network or local quantile regression for which we have bounds on the conditional coverage. On most datasets, the neural network slightly outperforms local quantile regression, which is expected due to its flexibility. Sections G.2.3 and G.2.4 discuss the choice of adjustment function. For certain adjustment functions, the domain of the scores $v = s(x, y)$ must be restricted to a subset of $\mathbb{R}$ to satisfy **H3**. Notably, $f_t(v) = tv$ requires $v > 0$ and $f_t(v) = \exp(tv)$ requires $v > 1$.

Section G.2.5 directly compares the proposed method with CQR, showing that CQR already

obtains a competitive conditional coverage but is outperformed by RCP-DR-CP in average volume. Section G.2.7 considers an approach to improve data efficiency. Instead of dividing the calibration dataset $\mathcal{D}_{\text{cal}}$ into two parts to estimate $\hat{\tau}$, we compute out-of-sample conformity scores on the training dataset $\mathcal{D}_{\text{train}}$ using K-fold cross-validation. This results in improved conditional coverage at the cost of training K additional models.

Section G.2.8 provides an additional comparison with Conditional Prediction with Conditional Guarantees (CPCG; Gibbs et al. (2025)). CPCG obtains a competitive conditional coverage but is 200-100000 times slower than RCP overall, limiting its applicability.

6.8. Conclusion

In this chapter, we introduced rectified conformal prediction (RCP), a novel framework that improves the reliability of prediction sets by making them sensitive to local data characteristics. The primary goal of RCP is to enhance conditional coverage, ensuring predictions are accurate for specific inputs, while strictly preserving the overall marginal coverage guarantee that is central to conformal prediction. Our approach achieves this by learning to rectify conformity scores based on input features, a task significantly simpler than estimating the entire conditional distribution of the output.

Our theoretical analysis confirms that RCP provides valid marginal coverage, with its conditional performance directly linked to the accuracy of the learned score adjustment. Across a range of experiments, RCP consistently improved conditional coverage metrics compared to several state-of-the-art methods. This improvement was often accompanied by sharper, more informative prediction sets. In essence, RCP offers a powerful, theoretically sound, and practical method for making conformal predictions more adaptive and trustworthy.

Limitations Despite its strong performance, this work has some limitations. First, the method's ability to improve conditional coverage depends on accurately learning how to adjust the scores, which can be challenging when the input is high-dimensional or with limited data. Second, our standard implementation requires splitting the calibration set, which reduces data efficiency. While we explored cross-validation to mitigate this, it comes at a higher computational cost. Third, the choice of the score transformation function was manual; future research could explore data-driven methods for selecting the optimal transformation, potentially leading to further performance gains. Finally, we did not compare RCP to recent baselines that were proposed in Dheur et al. (2025).

CHAPTER 7

Latent Recalibration

This chapter is based on the following paper:

> **Victor Dheur**, Souhaib Ben Taieb (2025). Multivariate Latent Recalibration for Conditional Normalizing Flows. *The 39th Annual Conference on Neural Information Processing Systems.*

7.1. Introduction

Multi-output regression arises in many practical applications, including weather forecasting (Setiawan et al., 2024), energy consumption prediction (Makaremi, 2025), and healthcare resource utilization (Cui et al., 2018). Despite this, research on calibration has primarily focused on the univariate case (Gneiting et al., 2007; H. Song et al., 2019; Sahoo et al., 2021; Kuleshov and Deshpande, 2022; Dewolf et al., 2022; Fakoor et al., 2023; Marx et al., 2023; Y. Chung et al., 2023; Gneiting and Resin, 2023).

Motivated by the lack of methods to both *assess* and *recalibrate* multi-output models, Y. Chung et al. (2024) introduced HDR calibration and the sampling-based HDR recalibration (`HDR-R`) approach. However, `HDR-R` does not produce an explicit probability density function (PDF) and depends on computationally expensive sampling and binning at test time. Other recalibration methods are restricted to single-output settings (Kuleshov et al., 2018; Vovk et al., 2020; Marx et al., 2022). Meanwhile, recent multi-output conformal prediction (CP) approaches provide calibrated prediction sets (Z. Wang et al., 2023; Feldman et al., 2023; Fang et al., 2025; Dheur et al., 2025), but they do not yield a fully calibrated predictive distribution.

In previous chapters, we developed methods for reliable probabilistic prediction in the form of univariate predictive distributions (Chapters 3 and 4) and multivariate prediction sets (Chapters 5 and 6). However, we have not addressed calibration techniques that produce multivariate predictive distributions.

To fill this gap, this final chapter introduces **latent calibration** and the corresponding **latent recalibration** (`LR`) method, which recalibrates invertible generative models (e.g., normalizing

flows (NFs) or flow matching (FM)) directly in their latent space. The key idea is to learn a latent-space transformation such that the model satisfies finite-sample guarantees on latent calibration. Unlike CP-based set methods or sampling-based recalibration such as `HDR-R`, `LR` outputs a fully recalibrated generative model with an explicit multivariate PDF. This enables efficient density evaluation and sampling, while providing finite-sample calibration guarantees. These properties are essential for downstream applications and improved decision-making (N. Klein, 2024).

Our contributions are as follows:

- We introduce **latent calibration**, a new notion of calibration defined within the latent space of invertible generative models, based on the distribution of latent norms.
- We propose `LR`, a recalibration method that produces multivariate predictive distributions with an explicit PDF, finite-sample latent calibration guarantees, and strong computational efficiency.
- We provide empirical evidence on 29 multi-dimensional tabular datasets and one high-dimensional image dataset, showing that `LR` consistently improves latent calibration and reduces negative log-likelihood (NLL). To support adoption and reproducibility, we release an open-source implementation[1].

7.2. Background

Most of the necessary background has been presented in Chapter 2.

Calibration, including probabilistic and HDR calibration, is discussed in Section 2.4.1. Then, their respective recalibration methods, quantile recalibration (`QR`) and HDR recalibration (`HDR-R`), are discussed in Section 2.4.3. We emphasize several limitations related to `HDR-R`: (1) it does not produce an explicit recalibrated PDF $\hat{f}'_{Y|X}$; (2) it generates duplicate samples; (3) it is subject to discretization errors when estimating $F_{G|x}$; and (4) it is computationally expensive, as it requires generating K initial samples for every recalibrated output.

Conformal prediction is discussed in Section 2.5. In this section, we emphasize that specific choices of conformity scores correspond to recalibration statistics: distributional conformal prediction (DCP) (Chernozhukov et al., 2021) uses $s_{\text{DCP}}(x, y) = \hat{F}_{Y|X=x}(y)$, while HPD-split (Izbicki et al., 2022) uses $s_{\text{HPD-split}}(x, y) = \text{HPD}_{\hat{f}_{Y|X=x}}(y)$; these match the transformations used in `QR` and `HDR-R`, respectively. In Section 2.4.3, we also considered pre-rank recalibration as a direct generalization of `HDR-R`. This highlights a unified framework connecting conformal prediction and recalibration, summarized in Table 7.1.

Invertible generative models that we consider in this chapter are reviewed in Section 2.2.3. These include normalizing flows (NFs) and flow matching (FM). Various NF architectures exist, including RealNVP (Dinh et al., 2017), MAF (Papamakarios, Pavlakou, et al., 2017), Glow (Kingma and Dhariwal, 2018), spline flows (Durkan et al., 2019), convex potential flows (C.-W. Huang et al., 2021) and transformer flows (Zhai et al., 2025). NFs are well-suited for `LR` due to their invertible mapping and explicit density.

[1] `https://github.com/Vekteur/latent-recalibration`

Table 7.1: Comparison of calibration notions, the associated random variable S (uniform under calibration), recalibration methods, and related conformal conformity scores.

Calibration notion	**Random variable**	**Recalibration method**	**Conformal method**
Probabilistic ($d = 1$)	$\hat{F}_{Y\|X}(Y)$	Quantile recalibration (`QR`)	DCP
HDR ($d \geqslant 1$)	$\text{HPD}_{\hat{f}_{Y\|X}}(Y)$	HDR recalibration (`HDR-R`)	HPD-split
Pre-rank ($d \geqslant 1$)	$F_{\hat{G}\|X}(G)$	Pre-rank recalibration	/
Latent ($d \geqslant 1$)	$F_{\rho_{\mathcal{Z}}(Z)}(\ell_{\hat{T}}(Y;X))$	**Latent recalibration** (`LR`)	CONTRA/L-CP

7.3. Related Work

Our work builds upon and contributes to generative modeling, calibration, conformal prediction, and methods that combine these concepts in the context of multi-dimensional regression.

Various notions of calibration have been studied, including probabilistic (Gneiting et al., 2007), marginal (Gneiting et al., 2007) and HDR (Y. Chung et al., 2024) calibration. Ziegel and Gneiting (2014) and Allen et al. (2024) also proposed multivariate notions of calibration but, to our knowledge, no calibration methods for these notions have been proposed.

While traditional CP focuses on univariate intervals (Romano et al., 2019; Sesia and Romano, 2021), recent multivariate CP methods create flexible regions. HPD-split (Izbicki et al., 2022) uses HPD values as scores. PCP (Z. Wang et al., 2023) uses balls around samples. ST-DQR (Feldman et al., 2023) selects samples based on a region in a latent space and creates balls around these samples. CONTRA (Fang et al., 2025) and L-CP (Dheur et al., 2025) operate in the latent space of NFs.

Certain methods explicitly merge CP and recalibration. Vovk et al. (2020) and Vovk et al. (2019) developed conformal predictive systems for calibrated univariate distributions. Marx et al. (2022) unified univariate recalibration methods under a CP lens. Our work extends this direction to multivariate outputs via a transformation in the latent space.

7.4. A New Latent Recalibration Method for Normalizing Flows

We propose a new recalibration method, called *latent recalibration* (`LR`), for conditional NFs. `LR` operates in the latent space and is specifically designed to achieve our newly introduced notion of multivariate *latent calibration*.

7.4.1 A New Notion of Multivariate Latent Calibration

Recall that, given a latent variable $Z \in \mathcal{Z}$ with a known distribution and an input $x \in \mathcal{X}$, conditional NFs estimate the conditional distribution of Y, $F_{Y|X=x}$, by learning a conditional bijective transformation $\hat{T} : \mathcal{Z} \to \mathcal{Y}$ such that the PDF $\hat{f}_{Y|X}$ of the transformed variable $\hat{T}(Z;x)$ approximates $f_{Y|X=x}$. However, model misspecification or significant estimation errors in the learned transformation $\hat{T}$ can lead to poor calibration of the induced distribution of $\hat{T}(Z;x)$.

We propose to leverage the simple structure of the latent space $\mathcal{Z}$ and assess calibration directly

in this space, a notion we refer to as *latent calibration*. By definition, if the NF is well-specified for $F_{Y|X=x}$, then the inverse transformation $\hat{T}^{-1}$ satisfies $\hat{T}^{-1}(Y;X) \overset{\text{d.}}{\approx} Z$, where $\overset{\text{d.}}{\approx}$ denotes approximate equality in distribution. Building on this observation, we define a norm $\rho_{\mathcal{Z}} : \mathcal{Z} \to \mathbb{R}_+$ over $\mathcal{Z}$ (e.g., $\rho_{\mathcal{Z}}(z) := \|z\|$). The goal is to test whether $\rho_{\mathcal{Z}}(\hat{T}^{-1}(Y;X)) \overset{\text{d.}}{\approx} \rho_{\mathcal{Z}}(Z)$.

Since the distribution of Z is known and standard (e.g., standard Gaussian), the distribution of $\rho_{\mathcal{Z}}(Z)$ is often known in closed-form. For instance, if $Z \sim \mathcal{N}(0, I_d)$ and $\rho_{\mathcal{Z}}(z) = \|z\|$, then $\rho_{\mathcal{Z}}(Z)$ follows a Chi distribution with d degrees of freedom (χ_d), whose PDF, CDF, and quantile function can be computed efficiently. As another example, if $Z \sim \mathcal{U}(B_d)$ is uniformly distributed over the unit hyperball and $\rho_{\mathcal{Z}}(z) = \|z\|$, then $\rho_{\mathcal{Z}}(Z)$ follows a $\text{Beta}(d, 1)$ distribution.

Definition 12. Consider a NF defined by a latent variable Z and a bijective transformation $\hat{T}$. For a pair (X, Y), define the *latent norm* w.r.t. $\hat{T}$ as

$$\hat{L} = \ell_{\hat{T}}(Y;X) = \rho_{\mathcal{Z}}(\hat{T}^{-1}(Y;X)). \tag{7.1}$$

The NF is said to be *latent calibrated* w.r.t. Z and the norm $\rho_{\mathcal{Z}}$ if the PIT of the latent norm follows a standard uniform distribution, i.e.,

$$\hat{U} = F_{\rho_{\mathcal{Z}}(Z)}(\hat{L}) \sim \mathcal{U}(0,1). \tag{7.2}$$

To assess whether a model is latent calibrated, we define the *latent expected calibration error* (L-ECE) as the L^1 distance between the CDF of the PIT variable $\hat{U}$ and the CDF of the uniform distribution:

$$\text{L-ECE}(\hat{T}) = \int_0^1 \left| F_{\hat{U}}(\alpha) - \alpha \right| d\alpha, \tag{7.3}$$

The L-ECE is minimized at 0 when $\hat{T}$ is perfectly latent calibrated, and has a maximum value of 0.5.

7.4.2 Multivariate Latent Recalibration

We propose a multivariate latent recalibration method, called `LR`, which performs a post-hoc adjustment of the latent space of a NF to ensure that the resulting model is latent calibrated. Key advantages of `LR` are that it yields a recalibrated distribution with an explicit PDF, remains computationally efficient, and has finite-sample guarantees on latent calibration (see Section 7.4.3).

Latent space transformation. `LR` uses the CDF $F_{\hat{L}}$ as its calibration map. We define a scalar strictly increasing transformation $r : \mathbb{R}_+ \to \mathbb{R}_+$ using the quantile function $F_{\hat{L}}^{-1}$, which maps the original latent norms $l \in \mathbb{R}_+$ to recalibrated norms as follows:

$$r(l) = F_{\hat{L}}^{-1}(F_{\rho_{\mathcal{Z}}(Z)}(l)). \tag{7.4}$$

We also define a vector-valued transformation $R : \mathcal{Z} \to \mathcal{Z}$ based on the scalar transformation r, which maps latent vectors z such that $\rho_{\mathcal{Z}}(R(z)) = r(\rho_{\mathcal{Z}}(z))$. When using the Euclidean norm $\rho_{\mathcal{Z}}(z) = \|z\|$, R is a radial transformation:

$$R(z) = \frac{r(\|z\|)}{\|z\|} \cdot z \quad (\text{with } R(0) = 0). \tag{7.5}$$

The transformation R rescales each vector z along its original direction by replacing its norm $\|z\|$ with $r(\|z\|)$. This procedure defines a new latent variable $Z' = R(Z)$, and the associated recalibrated NF $\hat{T}(Z'; X)$.

Proposition 4. The recalibrated NF $\hat{T}(Z'; X)$ defined with the new latent variable $Z' = R(Z)$ is *latent calibrated*, i.e. $\hat{U}' = F_{\rho_{\mathcal{Z}}(Z')}(\hat{L}) \sim \mathcal{U}(0, 1)$.

Proof. Consider the inverse transformation $r^{-1}(l) = F^{-1}_{\rho_{\mathcal{Z}}(Z)}\left(F_{\hat{L}}(l)\right)$ for a latent norm $l \in \mathbb{R}_+$. Then, using $\rho_{\mathcal{Z}}(R(z)) = r(\rho_{\mathcal{Z}}(z))$ the following identity holds:

$$F_{\rho_{\mathcal{Z}}(Z')}(l) = F_{r(\rho_{\mathcal{Z}}(Z))}(l) = F_{\rho_{\mathcal{Z}}(Z)}\left(r^{-1}(l)\right) = F_{\hat{L}}(l), \quad \forall l \in \mathbb{R}_+. \tag{7.6}$$

Then, it follows that

$$\mathbb{P}(\hat{U}^{'} \leqslant \alpha) = \mathbb{P}(F_{\rho_{\mathcal{Z}}(Z')}(\hat{L}) \leqslant \alpha) = \mathbb{P}(F_{\hat{L}}(\hat{L}) \leqslant \alpha) = \alpha. \tag{7.7}$$

□

Recalibrated predictive density. A distinctive feature of our LR recalibration procedure is that it produces a recalibrated distribution with an explicit multivariate PDF.

Note that the recalibrated NF can be interpreted as a composite transformation $\hat{T}' = \hat{T} \circ R$, applied to the original latent variable Z with density f_Z, typically a standard multivariate Gaussian. Given $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, assuming the transformation R is differentiable, the recalibrated predictive density $\hat{f}'_{Y|X=x}(y)$ can be computed using the change of variables formula. Let $z' = \hat{T}^{-1}(y; x)$ and $z = R^{-1}(z')$. Then, we have

$$\hat{f}'_{Y|X=x}(y) = f_Z\left(z\right) \left|\det\left(\nabla_z R(z)\right)\right|^{-1} \left|\det\left(\nabla_y \hat{T}^{-1}(y; x)\right)\right|. \tag{7.8}$$

Let us consider the case where $\rho_{\mathcal{Z}}(z) = \|z\|$. The inverse transformation takes the form $R^{-1}(z') = \frac{r^{-1}(\|z'\|)}{\|z'\|} \cdot z'$ and the Jacobian determinant of R can be computed efficiently as:

$$\left|\det\left(\nabla_z R(z)\right)\right| = \left(\frac{r(l)}{l}\right)^{d-1} \cdot \frac{\partial r(l)}{\partial l}, \quad \text{with } l = \|z\|. \tag{7.9}$$

A detailed proof is provided in Section H.1.1. The term $\frac{\partial r(l)}{\partial l}$ in (7.9) is computed using the chain rule as:

$$\frac{\partial r(l)}{\partial l} = \frac{\partial F^{-1}_{\hat{L}}(u)}{\partial u} \cdot \frac{\partial F_{\rho_{\mathcal{Z}}(Z)}(l)}{\partial l}, \quad \text{where } u = F_{\rho_{\mathcal{Z}}(Z)}(l). \tag{7.10}$$

To compute $\partial F_{\rho_{\mathcal{Z}}(Z)}(l)/\partial l$, we leverage the fact that $\rho_{\mathcal{Z}}(Z) \sim \chi_d$, whose PDF is available in closed-form and can be evaluated efficiently.

In practice, $F_{\hat{L}}$ is estimated by computing latent norms $\hat{L}_i = \ell_{\hat{T}}(X^{(i)}, Y^{(i)})$ using samples $(X^{(i)}, Y^{(i)})$ from the calibration set $\mathcal{D}_{\text{cal}}$. Section H.2 details how this can be achieved using kernel density estimation or monotonic splines, resulting in a differentiable estimate $\hat{F}_{\hat{L}}$ of $F_{\hat{L}}$. All operations are carried out in log-space to ensure numerical stability. Figure 7.1 illustrates `LR`, with the recalibrated predictive density $\hat{f}'_{Y|X}$ shown in the second column of the second row.

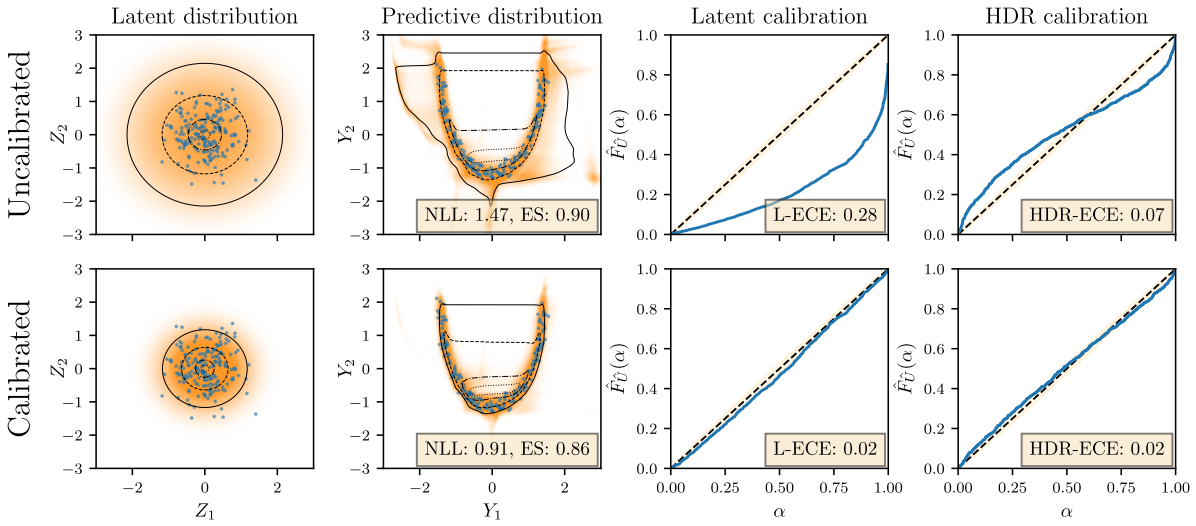


Figure 7.1: Illustration of LR for a bivariate output. The first column shows the latent distribution, the second column displays the predictive PDF, and the third and fourth columns show reliability diagrams for latent and HDR calibration, respectively. The first row corresponds to an uncalibrated NF, and the second row is the same model after LR. Calibration points and their projections in the latent space are shown in blue. The PDF for both the latent distribution and the predictive distribution is shown in orange. Level sets of the PIT of the latent norm at levels 0.01, 0.1, 0.5, and 0.9 are indicated with black contours in the second column, and their corresponding preimages are shown in the first column. LR improves both latent calibration (third column) and HDR calibration (fourth column). Additional prediction examples on real-world datasets are presented in Section H.4.3.

7.4.3 Useful Properties of Multivariate Latent Recalibration

We present finite-sample coverage guarantees for LR and highlight its connections to conformal prediction methods. We assume that R depends on an estimate $\hat{F}_{\hat{L}}$ of $F_{\hat{L}}$ based on latent norms $\hat{L}_1, \dots, \hat{L}_n$.

Finite-sample coverage guarantees for recalibrated latent norms. Let us assume that the estimated calibration map $\hat{F}_{\hat{L}}$ maps the i-th order statistic $\hat{L}_{(i)}$ of $\hat{L}_1, \dots, \hat{L}_n$ within a margin $\lambda/(n+1) \geqslant 0$ of the target quantile $i/(n+1)$, that is,

$$\hat{F}_{\hat{L}}(\hat{L}_{(i)}) \in \left[\frac{i-\lambda}{n+1}, \frac{i+\lambda}{n+1}\right]. \tag{7.11}$$

Then, letting $\epsilon = \frac{1+\lambda}{n+1}$, Theorem 1 of Marx et al. (2022) yields the following finite-sample coverage guarantee for the recalibrated latent norms:

$$\mathbb{P}\left(F_{\rho_{\mathcal{Z}}(Z')}\left(\ell_{\hat{T}}(Y;X)\right) \leqslant \alpha\right) = \mathbb{P}\left(\hat{F}_{\hat{L}}(\hat{L}) \leqslant \alpha\right) \in [\alpha - \epsilon, \alpha + \epsilon], \tag{7.12}$$

where we used (7.6) for the first equality and the probabilities are taken over X, Y, and the recalibrated latent norms $\hat{L}_1, \dots, \hat{L}_n$.

Equivalence with conformal prediction sets. We observe that the prediction sets derived from the recalibrated predictive density of `LR` coincide exactly with those obtained by the multivariate conformal methods CONTRA (Fang et al., 2025) and L-CP (Dheur et al., 2025). Specifically, this equivalence holds when `LR` uses the empirical CDF of the calibration scores $\mathcal{L} = \{\hat{L}_1, \dots, \hat{L}_n, +\infty\}$ as its calibration map, i.e., $\hat{F}_{\hat{L}}(l) = \frac{1}{n+1} \sum_{i=1}^{n} \mathbb{1}(\hat{L}_i \leqslant l)$.

CONTRA and L-CP are conformal methods that construct prediction sets using the conformity score $s_{\text{CONTRA}}(x, y) = s_{\text{L-CP}}(x, y) = \ell_{\hat{T}}(y; x)$. Under this choice of calibration map, for any $x \in \mathcal{X}$ and $\alpha \in (0, 1)$, we have

$$\{y \in \mathcal{Y} : F_{\rho_{\mathcal{Z}}(Z')}(\ell_{\hat{T}}(y; x)) \leqslant \alpha\} = \{y \in \mathcal{Y} : \hat{F}_{\hat{L}}(\ell_{\hat{T}}(y; x)) \leqslant \alpha\} \tag{7.13}$$

$$= \{y \in \mathcal{Y} : s_{\text{CONTRA}}(x, y) \leqslant \hat{F}^{-1}_{\hat{L}}(\alpha)\}, \tag{7.14}$$

where $\hat{F}^{-1}_{\hat{L}}(\alpha) = \hat{L}_{(\lceil \alpha(n+1) \rceil)}$ denotes the $(1 - \alpha)$ right empirical quantile of the calibration scores $\mathcal{L}$. This shows that the α-sublevel sets of the PIT of the latent norm of `LR` (7.13) correspond exactly to the conformal prediction sets produced by CONTRA and L-CP at coverage α (7.14). While this equivalence is notable, it is important to point out that the chosen calibration map $\hat{F}_{\hat{L}}$, being non-differentiable, does not yield a well-defined recalibrated predictive density function $\hat{f}'_{Y|X}$. This equivalence is summarized in the last row of Table 7.1.

Equivalence of LR and QR in the single-output setting. `QR` is a special case of `LR` where $d = 1$, $\mathcal{Z} = [0, 1]$, $Z \sim \mathcal{U}(0, 1)$, $\hat{T}^{-1}(y; x) = \hat{F}_{Y|X=x}(y)$ and $\rho_{\mathcal{Z}}(z) = z$. In this case, $R = \hat{F}^{-1}_{\hat{L}}$ and thus $\hat{T}'^{-1}(\cdot; x) = R^{-1} \circ \hat{T}^{-1}(\cdot; x) = \hat{F}_{\hat{L}} \circ \hat{F}_{Y|X=x} = \hat{F}'_{Y|X=x}$, showing that both methods perform exactly the same transformation.

7.5. Experiments

We present an extensive experimental study using 29 tabular datasets widely used in prior research (Tsoumakas et al., 2011; Cevid et al., 2022; Y. Chung et al., 2024; Feldman et al., 2023; Z. Wang et al., 2023; Barrio et al., 2024; Camehl et al., 2024). Furthermore, while recent work on model recalibration (Y. Chung et al., 2024; Fang et al., 2025) has primarily focused on data modalities with relatively low output dimensionality, we also include a high-dimensional output setting with an image dataset with a larger output dimension (Choi et al., 2020).

7.5.1 Datasets

Tabular datasets. The tabular datasets range in size from 103 to 50,000 data points, with the number of input features (p) varying from 1 to 368, and the number of output variables (d) ranging from 2 to 16. A detailed summary of these datasets is provided in Table A.2. Following the protocol of Y. Chung et al. (2024), we use a 65/20/15 split for training, validation, and testing. All input features and output targets are normalized to have zero mean and unit variance on the training set. Experiments are repeated 10 times with a different random splitting. For each run, we compare the same base predictor with or without recalibration.

Image dataset. We use the AFHQ dataset (Choi et al., 2020), which consists of high-resolution animal face images. The input $x \in \mathcal{X} = \{0, 1, 2\}$ indicates one of three classes (cat, dog, or wild animal), and the output is a 256×256 RGB image $y \in \mathcal{Y} = [-1, 1]^{3 \times 256 \times 256}$, resulting in an output dimension of $d = 196{,}608$. We follow the standard split with 14,630 training instances and 1,500 test instances. To improve sample quality, Zhai et al. (2025) add Gaussian noise $\epsilon \sim \mathcal{N}(0, 0.07^2)$ to each image y during training.

7.5.2 Experimental Setup

(Non-recalibrated) base predictor. For the tabular datasets, we consider convex potential flows (C.-W. Huang et al., 2021), MAFs (Papamakarios, Pavlakou, et al., 2017) and FM (Lipman et al., 2022). Results for the latter are deferred to Sections H.4.6 and H.4.7. As is standard, these NFs use a latent variable $Z \sim \mathcal{N}(0, I)$. Details on hyperparameter tuning are provided in Section H.3. A key difference between our setup and that of Y. Chung et al. (2024) is that their predictive distributions are restricted to multivariate Gaussians with diagonal covariance, whereas NFs can model dependencies between output dimensions. For the image dataset, we use the TarFlow model (Zhai et al., 2025), a transformer-based conditional NF pre-trained on AFHQ, which achieves state-of-the-art likelihood performance. In the following, we denote the non-recalibrated base predictor as `BASE`.

Compared methods. For our latent recalibration method, `LR`, we use the Euclidean norm $\rho_{\mathcal{Z}}(z) = \|z\|$ and estimate F_L using kernel density estimation with a Gamma kernel; details are provided in Section H.2. For both tabular and image datasets, we compare `LR` with the base predictor `BASE`. Additionally, we include `HDR-R` for tabular datasets only, as it becomes computationally prohibitive for TarFlow. For tabular datasets, following Y. Chung et al. (2024), the recalibration map $\hat{F}_U$ is learned on the validation set. This avoids using additional data for calibration and ensures a fair comparison with `BASE`, but sacrifices finite-sample guarantees. For the image dataset, since no separate calibration set is available, calibration is performed on the training data. This also sacrifices finite-sample guarantees, but we will show below that it still leads to substantial improvements in calibration.

Error metrics. We consider several error metrics to compare the different methods. For the tabular datasets, we evaluate model calibration using the latent expected calibration error (L-ECE) and the HDR expected calibration error (HDR-ECE). Both metrics range from 0 (best) to 0.5 (worst). Predictive accuracy is assessed using two strictly proper scoring rules: the NLL and the energy score (ES). Notably, `LR` yields a recalibrated density with a closed-form PDF, enabling direct computation of the NLL, which is not possible with `HDR-R`. Since the scales of NLL and ES vary across datasets, we report relative values, defined as the difference with the score achieved by `BASE`. All metrics are negatively oriented. Exact definitions are provided in Section H.3.1. For the image dataset, we report L-ECE and the bits per dimension (BPD), following Zhai et al. (2025). BPD corresponds to a rescaled version of the NLL (details in Section H.3.1).

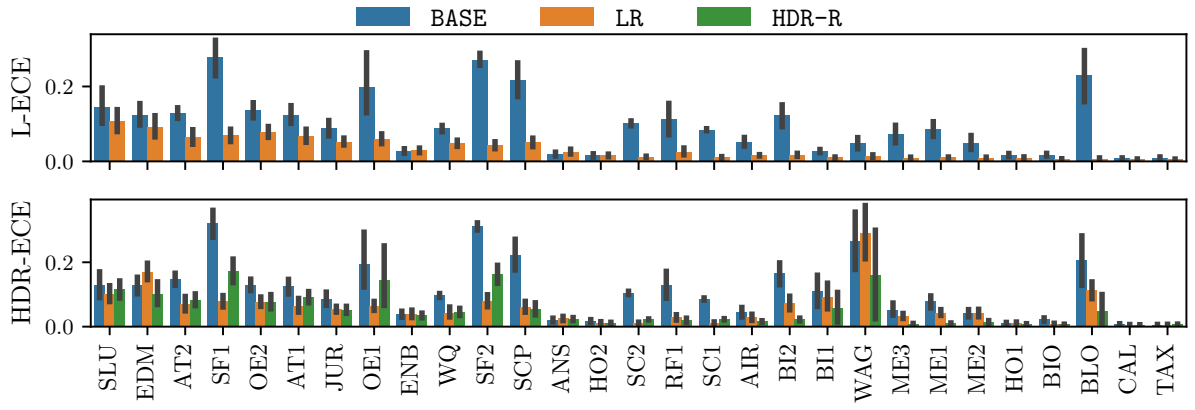


Figure 7.3: L-ECE and HDR-ECE on datasets sorted by size for a convex potential flow.

7.5.3 Results

Tabular datasets. Figure 7.2 presents the normalized difference relative to `BASE` for NLL and ES. We observe that `LR` reduces the NLL on the majority of datasets. Since NLL is a strictly proper scoring rule, this indicates that the recalibrated density $\hat{f}'_{Y|X}$ produced by `LR` generally provides a better fit to the true data distribution than the original model $\hat{f}_{Y|X}$. Both `LR` and `HDR-R` achieve an energy score close to that of `BASE`, with no significant differences. As noted by Alexander et al. (2022), the energy score has weaker discriminative ability compared to the NLL.

Figure 7.3 also shows the L-ECE (as a measure of latent calibration) and HDR-ECE (as a measure of HDR calibration), respectively. We see that `BASE` exhibits significant latent miscalibration across many datasets, with L-ECE values reaching up to 0.3 out of a maximum of 0.5. In contrast, `LR` consistently and substantially reduces L-ECE, demonstrating its effectiveness in achieving the desired latent calibration. Moreover, L-ECE tends to decrease as dataset size increases, which aligns with the finite-sample guarantees discussed in Section 7.4.3. Reliability diagrams in Section H.4.2 further confirm this improvement. Additional experiments on a misspecified model are provided in Section H.4.8, with significant improvements given by `LR` across all metrics including ES.

In Figure 7.3, while `LR` does not explicitly target HDR calibration, we observe that it significantly improves HDR-ECE compared to `BASE` on most datasets, often performing on par with `HDR-R`. As expected, `HDR-R` achieves low HDR-ECE values by design. These results suggest that improving latent calibration also enhances the calibration of HDRs.

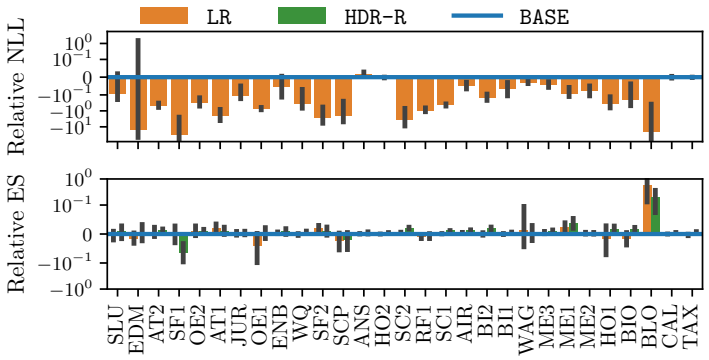


Figure 7.2: Relative NLL and ES on datasets sorted by size for a convex potential flow.

Access to a full, calibrated PDF is essential for any task requiring estimation of the probability mass within

Table 7.2: Performance of LR compared to BASE on the AFHQ dataset with TarFlow (standard errors across 20 evaluations).

L-ECE (×100)		BPD	
BASE	LR	BASE	LR
$47.43_{0.06253}$	$0.8954_{0.1609}$	$5.477_{3.523\text{e-}05}$	$5.465_{1.772\text{e-}05}$

an *arbitrary, non-standard region* of the output space, a capability that set-based methods (like CP) or pure sampling-based methods (like HDR-R) do not provide. A direct application is anomaly detection, where low-density points are classified as anomaly (Rozner et al., 2024; Perini et al., 2024). Other examples include risk assessment in engineering, targeted material design, or optimal control. To make the benefits of a full PDF concrete, we provide an experiment on a decision-making task in Section H.4.1.

Further detailed results with the same convex potential flow base predictor are provided in Section H.4.4. Section H.4.4 shows that the primary NLL gain obtained by LR is due to finding more "plausible" latent codes for the observed data under the base latent distribution. Section H.4.4 shows that LR significantly improves the time to compute the calibration map compared to HDR-R. Section H.4.4 hypothesizes through theoretical and empirical considerations that the reason the ES remains largely unchanged after LR is due to its relative insensitivity to misspecifications in variance and correlation.

Image dataset. The goal of our image data experiment is to understand the behaviour of latent calibration and recalibration in a very-high dimensional setting. We consider both noiseless and noisy settings, where the noisy test images include the same noise ϵ used during training.

Table 7.2 shows that BASE suffers from severe latent miscalibration, with L-ECE values approaching the maximum of 0.5. LR dramatically improves latent calibration, reducing L-ECE to below 0.01. We also report the bits-per-dimension (BPD), a scaled version of the NLL. Notably, LR does not degrade the original NLL; in fact, it slightly reduces it. LR preserves the visual quality of the samples from the base predictor, with no perceptually visible changes, which aligns with the very small change we observed in NLL.

7.6. Conclusion

We introduced latent recalibration (LR), a novel post-hoc method for calibrating conditional normalizing flows in multi-output regression. By transforming the latent space based on calibration scores derived from latent distances, LR achieves latent calibration, ensuring that prediction sets defined in the latent space have correct coverage. Unlike many conformal prediction methods that only output sets, and unlike sampling-based recalibration methods, LR yields a fully specified, recalibrated PDF. This offers significant advantages in terms of computational efficiency and applicability to tasks requiring density estimates. Our extensive experiments on tabular and high-dimensional image data demonstrate that LR consistently improves NLL, latent calibration, and HDR calibration.

Limitations. We identify the main limitations of `LR` as follows. Firstly, `LR` intentionally adjusts only the magnitude of latent vectors, not their direction, and thus cannot fix miscalibration arising from errors in the orientation of the learned latent manifold. While `LR` can only perform simple adjustments, this allows simplifying the difficult multivariate calibration problem into a tractable univariate one (calibrating norms). This enables connections with conformal prediction and recalibration methods, and has good empirical performance. Secondly, `LR` requires the norm of the latent distribution to follow a simple distribution. This is usually the case, as normalizing flows predominantly use a standard Gaussian latent variable. Thirdly, `LR` requires an invertible transformation between the response and latent spaces, and a latent random variable with a known, tractable density, which makes it incompatible with models such as variational auto-encoders (Kingma and Welling, 2014) or denoising diffusion probabilistic models (Ho et al., 2020). Instead, NF and FM models are natural fits for `LR`. Despite these considerations, `LR` provides a practical and effective tool for obtaining reliable, calibrated multivariate predictive distributions from generative models.

CHAPTER 8

Conclusion

Uncertainty quantification (UQ) is essential to making machine learning systems reliable and useful in practice. Beyond point accuracy, probabilistic machine learning provides a more complete view by producing probabilistic predictions such as predictive distributions and sets. In this thesis, we focused on single and multi-output regression problems and developed distribution-free algorithms that produce sharp, calibrated, and computationally efficient probabilistic predictions. In this concluding chapter, we summarize these contributions and discuss limitations and directions for future research.

8.1. Summary of Contributions

Post-hoc recalibration and recalibration training. Quantile recalibration (QR) is a state-of-the-art method for probabilistic calibration, but it is known to worsen negative log-likelihood (NLL). In Chapter 3, we designed smooth calibration maps based on kernel density estimation (KDE) for QR. This yields near state-of-the-art probabilistic calibration while substantially improving NLL relative to standard QR (Kuleshov et al., 2018). Moreover, the calibration map can be injected as a training-time regularizer, offering a practical trade-off between probabilistic calibration and predictive accuracy, as measured by the NLL and CRPS. In Chapter 4, we observed that separating the training of the base predictor and the post-hoc recalibration step is suboptimal, explaining the lower predictive accuracy of QR compared to standard training. To address this, we introduced quantile recalibration training (QRT), an end-to-end approach that integrates the effect of QR directly into training. This approach follows the principle of minimizing sharpness subject to calibration (Gneiting et al., 2007), improving predictive accuracy while maintaining state-of-the-art calibration after post-hoc recalibration.

A bridge between recalibration and conformal prediction. Recalibration and conformal prediction (CP) have largely been studied in parallel, yet they share deep structural similarities. In Chapter 3, we established an equivalence between QR and conformalizing quantiles via distributional conformal prediction (DCP, Chernozhukov et al., 2021). This connection permits

the transfer of conformal marginal guarantees between the two domains and explains the strong calibration performance of QR. In Chapter 7, we made a similar connection between latent recalibration (LR) and existing CP methods (Fang et al., 2025; Dheur et al., 2025).

A unified study of multi-output conformal regression. Many real-world problems require predicting multiple, often dependent, variables. However, the extension of UQ methods to this setting is relatively underexplored. In Chapter 5, we provided a unified comparative study of conformal methods for constructing multivariate prediction regions. This work systematically organized and analyzed nine distinct approaches, clarifying their properties and interconnections. To address existing limitations, we introduced two novel classes of conformal methods, CDF-based and latent-based, which generalize their univariate counterparts. These scores ensure the desirable property of asymptotic conditional coverage while maintaining exact finite-sample marginal coverage, offering new trade-offs between computational efficiency, sharpness, and compatibility with different types of generative models. This unified framework and the new scores contribute to the methodology for multi-output UQ.

Learning to improve conditional coverage. Although CP ensures coverage, the guarantee is marginal, i.e., averaged over all inputs, and may not hold for specific subgroups. To address this, in Chapter 6 we introduced rectified conformal prediction (RCP), a method designed to enhance conditional coverage while preserving the exact marginal guarantee. RCP refines conformity scores by learning their conditional quantiles, thereby avoiding the difficult and computationally intensive task of estimating their full conditional distribution. This allows the prediction sets to adapt to local data structure and heteroscedasticity. We provided a theoretical lower bound on the conditional coverage of RCP, explicitly linking its performance to the accuracy of the conditional quantile estimate and demonstrating its ability to improve the reliability of predictions for specific inputs.

Multivariate latent recalibration. For many downstream tasks, a prediction set is insufficient; a full, calibrated probability density function (PDF) is required for optimal decision-making. CP, however, does not provide a PDF, while existing recalibration methods are either limited to single-output settings or are computationally intensive and do not yield an explicit density. To address this, in Chapter 7, we introduce latent recalibration (LR), a recalibration method operating within the latent space of invertible generative models such as conditional normalizing flows. It is designed to improve latent calibration, ensuring that norms in the latent space of a model are statistically aligned with the ground truth. By leveraging KDE-based calibration maps as in Chapter 3, LR produces explicit and efficiently evaluable multivariate PDFs. Our empirical results showed that LR provides consistent improvements in latent calibration, HDR calibration, and NLL.

Comprehensive empirical studies. The evaluation of recalibration and conformal methods is inherently multi-faceted, yet prior empirical studies often relied on limited datasets, metrics, and on predictive distributions that were too restrictive (e.g., single Gaussians), potentially inflating the apparent gains of calibration methods (Kuleshov et al., 2018; Utpala and Rai, 2020; Marx et al., 2023; Y. Chung et al., 2024). In Chapters 3 and 5, we addressed this by conducting large-scale studies for both probabilistic calibration and multi-output CP under

flexible base predictors, such as mixture density networks and normalizing flows. This setup enabled a fairer assessment of methods across strictly proper scoring rules and calibration metrics, and provided a better picture of when and why methods succeed or fail. Other chapters also relied on comprehensive experiments to ensure the statistical significance of the results.

8.2. Limitations and Future Work

The methods developed in this thesis contribute to advancing the state of UQ in regression, yet they also highlight the field's remaining challenges. Recalibration and conformal prediction (CP) methods offer valuable guarantees and lend themselves to clear interpretations. For instance, probabilistic calibration ensures that all predictive intervals achieve their nominal coverage; HDR calibration aligns with highest-density region (HDR) coverage; and latent calibration corresponds to coverage within latent regions. However, these guarantees alone are not sufficient. Strictly proper scoring rules provide a principled framework for assessing predictive accuracy, yet they do not offer finite-sample guarantees. In the following, we suggest several concrete directions for future research in this area.

First, our empirical studies could be expanded along several important axes. Our work has primarily focused on capturing aleatoric uncertainty. A promising direction is to leverage epistemic UQ methods. For instance, Cabezas et al. (2025) leverage Bayesian models to predict the distribution of conformity scores, yielding more conservative prediction sets in regions with sparse data. Furthermore, our evaluations in Chapters 3 and 7 centered on the most common scoring rules, namely NLL, CRPS, and ES. A more complete picture would emerge from assessing performance under other strictly proper scoring rules, such as those presented in Section 2.3. When dealing with multivariate Gaussian mixtures, we provide closed-form expressions in Section B.1.

A valuable contribution to the community would be a systematic review and comparison of the rapidly growing number of proposed methods. While we compared a large set of baselines in Chapters 5 and 6, some methods were only included in one of the two studies. Given the growing number of methods in this area (Thurin et al., 2025; M. Klein et al., 2025; Braun et al., 2025), a comprehensive benchmark would help clarify the relative strengths and weaknesses of existing approaches and guide future development.

The contributions of this thesis operate within the standard in-distribution learning framework. Extending our methods to non-exchangeable data (Vovk, 2025), online (Gibbs and Candès, 2024) or federated settings (C. Lu et al., 2023) is a promising avenue, with initial ideas already available. In temporal settings such as TPPs, conformal methods that adapt to dependencies by iteratively modifying either the targeted coverage level (Gibbs and Candes, 2021) or the targeted quantile of the scores (Angelopoulos et al., 2023), are especially promising. However, they lose the marginal coverage guarantee, which is expected in the presence of distribution shift.

While this thesis focused on improving conditional reliability, further work is needed to develop methods with stronger and more practical conditional guarantees. As discussed in Section 2.4.5, applying calibration methods to pre-defined groups can yield significant gains in conditional coverage and predictive accuracy but at the cost of weaker marginal guarantees. An alternative approach involves local weighting schemes that can make more efficient use of limited calibration

data (Luo et al., 2022). Combining the training procedures developed in this thesis, such as QRT and LR, with locally weighted calibration maps is a compelling direction for developing methods that achieve approximate conditional calibration without requiring pre-defined groups.

The literature on calibration for regression, including this thesis, primarily centers on probabilistic calibration, which is an essentially unconditional notion (Gneiting and Resin, 2023). A more ambitious goal is to pursue stronger, conditional notions like auto-calibration (Section 2.4.4), which has been well-studied in classification (Kull et al., 2019; Popordanoska and Blaschko, 2025). Differentiable estimators have shown promise for achieving this in classification (Popordanoska et al., 2022), but adapting these ideas to the regression setting, particularly for multi-output problems with flexible generative models, remains a challenging open problem. Nevertheless, ideas based on low-dimensional density estimation may offer a path forward (Kuleshov and Deshpande, 2022).

Our work in Chapter 5 highlighted the complementary strengths of different multi-output conformal methods: density-based methods such as C-HDR yield sharp prediction sets but can be computationally intensive, while latent-based methods are computationally fast but may produce larger sets. A potential research direction is to design methods combining these strengths. One could, for example, enforce a monotonic relationship between the latent and output-space densities using a regularization term based on monotonic neural networks (H. Kim and Lee, 2023). The central challenge would be to design a latent representation that is expressive enough for accurate modeling while strictly preserving this monotonicity, thereby ensuring that highest density regions in latent space map to highest density regions in the output space.

A direct extension of our work is to develop a training-time procedure for latent recalibration, analogous to the quantile recalibration training (QRT) we introduced. Such a method, which could be termed latent recalibration training (LRT), would integrate the latent recalibration step from Chapter 7 directly into the end-to-end training process. This can be achieved using the readily available differentiable quantile functions for chi-squared or gamma distributions in TensorFlow Probability (Dillon et al., 2017), or by developing a custom implementation for PyTorch. This would enable the training of calibrated invertible generative models that could potentially further improve predictive accuracy while maintaining finite-sample calibration properties.

More broadly, the tools developed in this thesis can be applied to control risks beyond simple miscoverage. As outlined in Section 2.5.5, frameworks like conformal risk control (Angelopoulos et al., 2024b) and learn then test (Angelopoulos et al., 2025) can be used to provide finite-sample guarantees for task-specific losses, such as the false negative rate in image segmentation. The conformity scores proposed in Chapters 5 and 6 are fully compatible with this framework, opening the door to controlling custom risks while benefiting from the adaptive nature of our methods.

The last point connects to the goal of more reliable decision-making. We share the view of Y. Chung (2025) that an important future direction lies in elucidating connections between calibration notions and their implications for downstream decision-making, a task only attempted by a few studies. In classification, models satisfying decision calibration (S. Zhao et al., 2021) accurately estimate the utility associated with a decision-making task. In regression, Sahoo et al. (2021) established an analogous result for threshold-calibrated models, for which we present a direct generalization to pre-rank calibration in Section 2.6. Nonetheless, neither of these works

claim that calibration necessarily leads to higher utility. In a different approach, Y. Chung (2025) (Chapter 5) propose constructing tailored proper scoring rules to improve utility in specific decision-making tasks. Their method, however, is limited to classification settings with a finite number of actions. Thus, a central challenge remains to navigate the trade-offs among calibration, sharpness, and computational efficiency to enhance downstream decision-making.

While gradient-boosted decision trees such as XGBoost (Tianqi Chen and Guestrin, 2016) have remained strong performers for tabular data, the design of competitive neural architectures remains an active area of research. Efforts have explored diverse approaches, from novel architectures such as the attention-based TabNet (Arik and Pfister, 2021) to retrieval-augmented models (Gorishniy et al., 2024). Alongside these, another promising direction focuses on refining the foundational Multi-Layer Perceptron (MLP). Research has shown that well-tuned MLPs can be highly competitive (Kadra et al., 2021), and their performance can be further enhanced with techniques such as ensembling. A notable recent example is TabM (Gorishniy et al., 2025), which trains a parameter-efficient ensemble of MLPs by sharing the vast majority of parameters between its submodels, achieving state-of-the-art performance and efficiency on public benchmarks. While mixture density networks or normalizing flows in our study are based on standard MLPs, extensions involving these approaches could be considered, potentially yielding a more informative uncertainty quantification.

Finally, our empirical analyses primarily relied on tabular benchmarks, whose datasets typically contain at most 50,000 instances. While recent benchmarks aim to identify the hardest tabular datasets (McElfresh et al., 2023), the scale of tabular data is still far behind vision and language, where training on billions of images (Schuhmann et al., 2022) or trillions of tokens (DeepSeek-AI et al., 2024) is typical, reflecting a broader trend in tabular machine learning where progress has been perceived to lag behind other modalities (Borisov et al., 2024; Gorishniy et al., 2021; Shwartz-Ziv and Armon, 2022; Grinsztajn et al., 2022). Van Breugel and Van Der Schaar (2024) argue that this may be partly due to the limited scale of existing benchmarks. They push for the development of Large Tabular Models (LTMs), tabular models trained on a large collection of data, with recent developments in this direction (Hollmann et al., 2022; M. J. Kim et al., 2024). Studying the calibration of such models is timely, as they are often evaluated under distribution shift. Other emerging application frontiers include calibrating the uncertainty of large language models (Band et al., 2024; Cherian et al., 2024; Xiao et al., 2025).

Taken together, the methods and comparative studies in this thesis contribute foundational components for reliable UQ in regression. We hope these contributions will support the development of UQ methods that are well calibrated, sharp, and computationally efficient.

Bibliography

Abdar, Moloud et al. (2021). "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". *An international journal on information fusion* 76, pp. 243–297.

Abu-Mostafa, Yaser S, Malik Magdon-Ismail, and Hsuan-Tien Lin (2012). *Learning From Data*. AMLBook.

Alaa, Ahmed M, Zeshan Hussain, and David Sontag (2023). "Conformalized unconditional quantile regression". *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 10690–10702.

Alexander, C et al. (2022). "Evaluating the discrimination ability of proper multi-variate scoring rules". *Annals of operations research* 334 (1), pp. 857–883.

Allen, Sam, Johanna Ziegel, and David Ginsbourger (2024). "Assessing the calibration of multivariate probabilistic forecasts". *Quarterly journal of the Royal Meteorological Society. Royal Meteorological Society (Great Britain)* 150 (760), pp. 1315–1335.

Allen, Sam et al. (2023). "Weighted verification tools to evaluate univariate and multivariate probabilistic forecasts for high-impact weather events". *Weather and forecasting* 38 (3), pp. 499–516.

Allen, Sam et al. (2025). "In-sample calibration yields conformal calibration guarantees". arXiv: `2503.03841 [stat.ME]`.

Amini, Alexander et al. (2020). "Deep Evidential Regression". *Advances in Neural Information Processing Systems* 33, pp. 14927–14937.

Amos, Brandon, Lei Xu, and J Zico Kolter (2017). "Input Convex Neural Networks". *The 34th International Conference on Machine Learning*. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 146–155.

Amoukou, Salim I and Nicolas JB Brunel (2023). "Adaptive conformal prediction by reweighting nonconformity score". *arXiv:2303.12695*.

Angelopoulos, Anastasios N and Stephen Bates (2021). "A gentle introduction to conformal prediction and distribution-free uncertainty quantification". arXiv: `2107.07511 [cs.LG]`.

– (2023). "Conformal prediction: A gentle introduction". *Foundations and Trends® in Machine Learning* 16 (4), pp. 494–591.

Angelopoulos, Anastasios N, Emmanuel Candès, and Ryan J Tibshirani (2023). "Conformal PID Control for Time Series Prediction". *Neural Information Processing Systems* abs/2307.16895, pp. 23047–23074.

Angelopoulos, Anastasios N, Rina Foygel Barber, and Stephen Bates (2024a). "Theoretical foundations of conformal prediction". arXiv: 2411.11824 [math.ST].

Angelopoulos, Anastasios N et al. (2021). "Uncertainty Sets for Image Classifiers using Conformal Prediction". *International Conference on Learning Representations*.

Angelopoulos, Anastasios N et al. (2024b). "Conformal Risk Control". *The Twelfth International Conference on Learning Representations*.

Angelopoulos, Anastasios N et al. (2025). "Learn then test: Calibrating predictive algorithms to achieve risk control". *The annals of applied statistics* 19 (2), pp. 1641–1662.

Arik, Sercan Ö and Tomas Pfister (2021). "TabNet: Attentive Interpretable Tabular Learning". *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence* 35 (8), pp. 6679–6687.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). "Wasserstein Generative Adversarial Networks". *International Conference on Machine Learning*. PMLR, pp. 214–223.

Arpogaus, Marcel et al. (2023). "Short-term density forecasting of low-voltage load using Bernstein-polynomial normalizing flows". *IEEE transactions on smart grid* 14 (6), pp. 4902–4911.

Bacry, Emmanuel and Jean-François Muzy (2014). "Hawkes model for price and trades high-frequency dynamics". *Quantitative Finance* 14.7, pp. 1147–1166.

Band, Neil et al. (2024). "Linguistic Calibration of Long-Form Generations". *International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, pp. 2732–2778.

Barrio, Eustasio del, Alberto González Sanz, and Marc Hallin (2024). "Nonparametric multiple-output center-outward quantile regression". *Journal of the American Statistical Association*, pp. 1–43.

Bates, Stephen et al. (2021). "Distribution-free, risk-controlling prediction sets". *Journal of the ACM* 68 (6), pp. 1–34.

Benavoli, Alessio, Giorgio Corani, and Francesca Mangili (2016). "Should We Really Use Post-Hoc Tests Based on Mean-Ranks?" *Journal of machine learning research: JMLR* 17 (5), pp. 1–10.

Berger, James O and Leonard A Smith (2019). "On the Statistical Formalism of Uncertainty Quantification". *Annual Review of Statistics and Its Application* 6.1, pp. 433–460.

Bhattacharya, Pallab K and Ashis K Gangopadhyay (1990). "Kernel and nearest-neighbor estimation of a conditional quantile". *The Annals of Statistics*, pp. 1400–1415.

Bian, Michael and Rina Foygel Barber (2023). "Training-conditional coverage for distribution-free predictive inference". *Electronic Journal of Statistics* 17.2, pp. 2044–2066.

Bishop, Christopher M (1994). *Mixture density networks*. Tech. rep. Birmingham.

– (2006). *Pattern recognition and machine learning*. Vol. 4. Springer.

Blasiok, Jaroslaw and Preetum Nakkiran (2023). "Smooth ECE: Principled Reliability Diagrams via Kernel Smoothing". *The Twelfth International Conference on Learning Representations*.

Blondel, Mathieu et al. (2020). "Fast Differentiable Sorting and Ranking". *The 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 950–959.

Blundell, Charles et al. (2015). "Weight Uncertainty in Neural Network". *The 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 1613–1622.

Bommasani, Rishi et al. (2021). "On the opportunities and risks of foundation models". arXiv: 2108.07258 [cs.LG].

Bond-Taylor, Sam et al. (2021). "Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models". *IEEE transactions on pattern analysis and machine intelligence* PP.

Borisov, Vadim et al. (2024). "Deep neural networks and tabular data: A survey". *IEEE transactions on neural networks and learning systems* 35 (6), pp. 7499–7519.

Bosser, Tanguy and Souhaib Ben Taieb (2023). "On the Predictive accuracy of Neural Temporal Point Process Models for Continuous-time Event Data". *Transactions of Machine Learning Research (TMLR)*.

Box, George E P and George C Tiao (1992). *Bayesian inference in statistical analysis: Box/Bayesian*. Wiley Classics Library. Nashville, TN: John Wiley & Sons. 608 pp.

Bracher, Johannes et al. (2021). "Evaluating epidemic forecasts in an interval format". *PLoS computational biology* 17 (2), e1008618.

Braun, Sacha et al. (2025). "Minimum Volume Conformal Sets for Multivariate Regression". arXiv: `2503.19068 [stat.ML]`.

Brehmer, Jonas R and Tilmann Gneiting (2021). "Scoring interval forecasts: Equal-tailed, shortest, and modal interval". *BJOG: an international journal of obstetrics and gynaecology* 27 (3), pp. 1993–2010.

Breiman, Leo (2001). "Random forests". *Machine learning* 45 (1), pp. 5–32.

Breiman, Leo et al. (1984). *Classification and Regression Trees*. Wadsworth.

Brier, Glenn W. (1950). "Verification of forecasts expressed in terms of probability". *Monthly Weather Review* 78.1, pp. 1–3.

Bröcker, Jochen (2009). "Reliability, sufficiency, and the decomposition of proper scores". *Quarterly Journal of the Royal Meteorological Society* 135 (643), pp. 1512–1519.

Buchweitz, Erez, João Vitor Romano, and Ryan J Tibshirani (2025). "Asymmetric Penalties Underlie Proper Loss Functions in Probabilistic Forecasting".

Cabezas, Luben Miguel Cruz et al. (2025). "Epistemic Uncertainty in Conformal Scores: A Unified Approach". *The 41st Conference on Uncertainty in Artificial Intelligence*.

Camehl, Annika, Dennis Fok, and Kathrin Gruber (2024). "On superlevel sets of conditional densities and multivariate quantile regression". *Journal of Econometrics* (105807), p. 105807.

Cao, Chengtai et al. (2024). "CCTR: Calibrating trajectory prediction for uncertainty-aware motion planning in autonomous driving". *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence* 38 (19), pp. 20949–20957.

Carlier, Guillaume, Victor Chernozhukov, and Alfred Galichon (2016). "Vector quantile regression: An optimal transport approach". *Annals of statistics* 44 (3), pp. 1165–1192.

Cauchois, Maxime, Suyash Gupta, and John C Duchi (2021). "Knowing what you know: valid and validated confidence sets in multiclass and multilabel prediction". *Journal of machine learning research: JMLR* 22 (1), pp. 3681–3722.

Cevid, Domagoj et al. (2022). "Distributional Random Forests: Heterogeneity Adjustment and Multivariate Distributional Regression". *Journal of machine learning research: JMLR* 23 (333), pp. 1–79.

Chen, Colin and Ying Wei (2005). "Computational issues for quantile regression". *Sankhyā: The Indian Journal of Statistics*, pp. 399–417.

Chen, T et al. (2018). "Neural ordinary differential equations". *Advances in neural information processing systems* 31, pp. 6572–6583.

Chen, Tianqi and Carlos Guestrin (2016). "XGBoost: A Scalable Tree Boosting System". *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and*

Data Mining. KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (San Francisco California USA). New York, NY, USA: ACM.

Cherian, John, Isaac Gibbs, and Emmanuel Candes (2024). "Large language model validity via enhanced conformal prediction methods". *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Chernozhukov, Victor, Iván Fernández-Val, and Blaise Melly (2022). "Fast algorithms for the quantile regression process". *Empirical economics*, pp. 1–27.

Chernozhukov, Victor and Christian Hansen (2005). "An IV model of quantile treatment effects". *Econometrica* 73.1, pp. 245–261.

Chernozhukov, Victor, Kaspar Wüthrich, and Yinchu Zhu (2021). "Distributional conformal prediction". *Proceedings of the National Academy of Sciences of the United States of America* 118 (48).

Choi, Yunjey et al. (2020). "StarGAN v2: Diverse Image Synthesis for Multiple Domains". *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Seattle, WA, USA). IEEE, pp. 8188–8197.

Chung, Kai-Min and Hsueh-I Lu (2005). "An optimal algorithm for the maximum-density segment problem". *SIAM journal on computing* 34 (2), pp. 373–387.

Chung, Youngseog (2025). *Methods for calibrated uncertainty quantification and understanding its utility*.

Chung, Youngseog, Ian Char, and Jeff Schneider (2024). "Sampling-based Multi-dimensional Recalibration". *International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, pp. 8919–8940.

Chung, Youngseog, Aaron Rumack, and Chirag Gupta (2023). "Parity calibration". *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*. Vol. 216. Proceedings of Machine Learning Research. PMLR, pp. 413–423.

Chung, Youngseog et al. (2021). "Beyond Pinball Loss: Quantile Methods for Calibrated Uncertainty Quantification". *Advances in neural information processing systems* 34, pp. 10971–10984.

Cinlar, E (2009). *Probability and Stochastics*. 2011th ed. Graduate texts in mathematics. New York, NY: Springer. 558 pp.

Colombo, Nicolo (2024). "Normalizing flows for conformal regression". *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, pp. 881–893.

Cui, Liwen et al. (2018). "Prediction of the healthcare resource utilization using multi-output regression models". *IISE transactions on healthcare systems engineering* 8 (4), pp. 291–302.

Cuturi, Marco, Olivier Teboul, and Jean-Philippe Vert (2019). "Differentiable Ranking and Sorting using Optimal Transport". *Advances in Neural Information Processing Systems* 32.

Cybenko, G (1989). "Approximation by superpositions of a sigmoidal function". *Mathematics of Control, Signals, and Systems* 2 (4), pp. 303–314.

Dawid, A Philip (1984). "Present Position and Potential Developments: Some Personal Views: Statistical Theory: The Prequential Approach". *Journal of the Royal Statistical Society. Series A* 147 (2), pp. 278–292.

De Bortoli, Valentin et al. (2025). "Distributional Diffusion Models with Scoring Rules". *Forty-second International Conference on Machine Learning*.

DeepSeek-AI et al. (2024). "DeepSeek-V3 Technical Report". arXiv: `2412.19437 [cs.CL]`.

DeGroot, Morris H and Stephen E Fienberg (1981). *Assessing Probability Assessors: Calibration and Refinement.* Research rep.

Demšar, Janez (2006). "Statistical Comparisons of Classifiers over Multiple Data Sets". *Journal of machine learning research: JMLR* 7, pp. 1–30.

Deutschmann, Nicolas, Mattia Rigotti, and María Rodríguez Martínez (2023). "Adaptive conformal regression with Jackknife+ rescaled scores". *arXiv:2305.19901.*

Dewolf, Nicolas, Bernard De Baets, and Willem Waegeman (2022). "Valid prediction intervals for regression problems". *Artificial Intelligence Review.*

Dewolf, Nicolas, Bernard De Baets, and Willem Waegeman (2025). "Conditional validity of heteroskedastic conformal regression". *Information and Inference: A Journal of the IMA* 14.2, iaaf013.

Dey, Biprateep et al. (2022). "Conditionally Calibrated Predictive Distributions by Probability-Probability Map: Application to Galaxy Redshift Estimation and Probabilistic Forecasting". *arXiv:2205.14568.*

Dheur, Victor and Souhaib Ben Taieb (2023). "A Large-Scale Study of Probabilistic Calibration in Neural Network Regression". *The 40th International Conference on Machine Learning.* PMLR.

– (2024). "Probabilistic Calibration by Design for Neural Network Regression". *The 27th International Conference on Artificial Intelligence and Statistics.* Vol. 238. Proceedings of Machine Learning Research. PMLR, pp. 3133–3141.

– (2025). "Multivariate Latent Recalibration for Conditional Normalizing Flows". *The 39th Annual Conference on Neural Information Processing Systems.*

Dheur, Victor et al. (2024). "Distribution-Free Conformal Joint Prediction Regions for Neural Marked Temporal Point Processes". *Machine Learning* 113, pp. 7055–7102.

Dheur, Victor et al. (2025). "A unified comparative study with generalized conformity scores for multi-output conformal regression". *The 42nd International Conference on Machine Learning.* PMLR.

Dillon, Joshua V et al. (2017). "TensorFlow Distributions". arXiv: `1711.10604 [cs.LG]`.

Ding, Tiffany et al. (2024). "Class-conditional conformal prediction with many classes". *Advances in Neural Information Processing Systems* 36.

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio (2017). "Density estimation using Real NVP". *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.*

Diquigiovanni, Jacopo, Matteo Fontana, and Simone Vantini (2021a). "Conformal Prediction bands for multivariate functional data". *Journal of Multivariate Analysis*, p. 104879.

– (2021b). "The importance of being a band: Finite-sample exact distribution-free prediction sets for functional data". arXiv: `2102.06746 [stat.ME]`.

– (2024). *Distribution-Free Prediction Bands for Multivariate Functional Time Series: an Application to the Italian Gas Market.* arXiv:2107.00527 [stat].

Dolatabadi, Hadi Mohaghegh, Sarah Erfani, and Christopher Leckie (2020). "Invertible Generative Modeling using Linear Rational Splines". *The 23rd International Conference on Artificial Intelligence and Statistics.* Vol. 108. Proceedings of Machine Learning Research. PMLR, pp. 4236–4246.

Du, Hailiang (2021). "Beyond Strictly Proper Scoring Rules: The Importance of Being Local". *Weather and Forecasting* 36 (2), pp. 457–468.

Du, Nan et al. (2016). "Recurrent Marked Temporal Point Processes: Embedding Event History to Vector". *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

Dua, Dheeru and Casey Graff (2017). *UCI Machine Learning Repository*.

Durkan, Bekasov, Murray, et al. (2019). "Neural spline flows". *Advances in neural information processing systems*.

Einbinder, Bat-Sheva et al. (2022). "Training Uncertainty-Aware Classifiers with Conformalized Deep Learning". *Advances in Neural Information Processing Systems*.

El Nahhas, Omar S M et al. (2024). "Regression-based Deep-Learning predicts molecular biomarkers from pathology slides". *Nature communications* 15 (1), p. 1253.

English, Eshant et al. (2024). "JANET: Joint Adaptive predictioN-region Estimation for time-series". arXiv: `2407.06390 [stat.ML]`.

Enguehard, Joseph et al. (2020). "Neural Temporal Point Processes For Modelling Electronic Health Records". *Proceedings of Machine Learning Research (PMLR)* 136, pp. 85–113.

Fakoor, Rasool et al. (2023). "Flexible Model Aggregation for Quantile Regression". *Journal of machine learning research: JMLR* 24 (162), pp. 1–45.

Fang, Zhenhan, Aixin Tan, and Jian Huang (2025). "CONTRA: Conformal prediction region via normalizing flow transformation". *The Thirteenth International Conference on Learning Representations*.

Farajtabar, Mehrdad et al. (2017). *COEVOLVE: A Joint Point Process Model for Information Diffusion and Network Co-evolution*. Journal of Machine Learning Research, 18, 1-49.

Feldman, Shai, Stephen Bates, and Yaniv Romano (2021). "Improving conditional coverage via orthogonal quantile regression". *Advances in neural information processing systems*.

– (2023). "Calibrated Multiple-Output Quantile Regression with Representation Learning". *Journal of machine learning research: JMLR* 24 (24), pp. 1–48.

Fermanian, Jean-Baptiste, Mohamed Hebiri, and Joseph Salmon (2025). "Class conditional conformal prediction for multiple inputs by p-value aggregation". *The 39th Annual Conference on Neural Information Processing Systems*.

Fontana, Matteo, Gianluca Zeni, and Simone Vantini (2023). "Conformal prediction: A unified review of theory and new challenges". *Bernoulli: official journal of the Bernoulli Society for Mathematical Statistics and Probability* 29 (1), pp. 1–23.

Fortuin, Vincent (2022). "Priors in Bayesian deep learning: A review". *International statistical review = Revue internationale de statistique*.

Foygel Barber, Rina et al. (2021a). "Predictive inference with the jackknife+". *The Annals of Statistics* 49 (1), pp. 486–507.

– (2021b). "The limits of distribution-free conditional predictive inference". *Information and Inference: A Journal of the IMA* 10 (2), pp. 455–482.

Freund, Yoav and Robert E Schapire (1997). "A decision-theoretic generalization of on-line learning and an application to boosting". *Journal of computer and system sciences* 55 (1), pp. 119–139.

Friedman, Milton (1940). "A Comparison of Alternative Tests of Significance for the Problem of m Rankings". *The Annals of Mathematical Statistics* 11 (1), pp. 86–92.

Gal, Yarin and Zoubin Ghahramani (2016). "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". *Advances in neural information processing systems*. Proceedings of Machine Learning Research 48, pp. 1050–1059.

Gamble, Cooper, Shahriar Faghani, and Bradley J Erickson (2025). "Applying conformal prediction to a deep learning model for intracranial hemorrhage detection to improve trustworthiness". *Radiology. Artificial intelligence* 7 (2), e240032.

Germain, Mathieu et al. (2015). "MADE: Masked Autoencoder for Distribution Estimation". *The 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 881–889.

Ghahramani, Zoubin (2015). "Probabilistic machine learning and artificial intelligence". *Nature* 521 (7553), pp. 452–459.

Gibbs, Isaac and Emmanuel Candes (2021). "Adaptive Conformal Inference Under Distribution Shift". *Advances in Neural Information Processing Systems* 34, pp. 1660–1672.

Gibbs, Isaac and Emmanuel J Candès (2024). "Conformal inference for online prediction with arbitrary distribution shifts". *Journal of machine learning research: JMLR* 25 (162), pp. 1–36.

Gibbs, Isaac, John J Cherian, and Emmanuel J Candès (2025). "Conformal prediction with conditional guarantees". *Journal of the Royal Statistical Society Series B: Statistical Methodology*, qkaf008.

Gijsbers, Pieter et al. (2019). "An Open Source AutoML Benchmark". arXiv: `1907.00909 [cs.LG]`.

Gneiting, Tilmann, Fadoua Balabdaoui, and Adrian E Raftery (2007). "Probabilistic forecasts, calibration and sharpness". *Journal of the Royal Statistical Society. Series B, Statistical methodology* 69 (2), pp. 243–268.

Gneiting, Tilmann and Matthias Katzfuss (2014). "Probabilistic Forecasting". *Annual Review of Statistics and Its Application* 1 (1). doi: 10.1146/annurev-statistics-062713-085831, pp. 125–151.

Gneiting, Tilmann and Adrian E Raftery (2007). "Strictly Proper Scoring Rules, Prediction, and Estimation". *Journal of the American Statistical Association* 102 (477), pp. 359–378.

Gneiting, Tilmann and Johannes Resin (2023). "Regression diagnostics meets forecast evaluation: conditional calibration, reliability diagrams, and coefficient of determination". *Electronic journal of statistics* 17 (2), pp. 3226–3286.

Gneiting, Tilmann et al. (2023). "Model Diagnostics and Forecast Evaluation for Quantiles". *Annual Review of Statistics and Its Application*. doi: 10.1146/annurev-statistics-032921-020240.

Good, I J (1952). "Rational decisions". *Journal of the Royal Statistical Society. Series B, Statistical methodology* 14 (1), pp. 107–114.

Good, I J et al. (1971). "Comment on "Measuring information and uncertainty."" *Foundation of Statistical Inference*, pp. 265–273.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. Vol. 1. MIT Press. 800 pp.

Goodfellow, Ian et al. (2014). "Generative Adversarial Nets". *Advances in neural information processing systems*.

Gorishniy, Yury, Akim Kotelnikov, and Artem Babenko (2025). "TabM: Advancing tabular deep learning with parameter-efficient ensembling". *The Thirteenth International Conference on Learning Representations*.

Gorishniy, Yury et al. (2021). "Revisiting deep learning models for tabular data". *Advances in neural information processing systems* 34, pp. 18932–18943.

Gorishniy, Yury et al. (2024). "TabR: Tabular Deep Learning Meets Nearest Neighbors". *The Twelfth International Conference on Learning Representations*.

Gretton, A et al. (2012). "A Kernel Two-Sample Test". *Journal of machine learning research: JMLR* 13, pp. 723–773.

Grimit, E P et al. (2006). "The continuous ranked probability score for circular variables and its application to mesoscale forecast ensemble verification". *Quarterly Journal of the Royal Meteorological Society* 132 (621C), pp. 2925–2942.

Grinsztajn, Léo, Edouard Oyallon, and G Varoquaux (2022). "Why do tree-based models still outperform deep learning on typical tabular data?" *Advances in neural information processing systems* 35, pp. 507–520.

Grover, Aditya et al. (2019). "Stochastic Optimization of Sorting Networks via Continuous Relaxations". *International Conference on Learning Representations*.

Gruber, Cornelia et al. (2023). "Sources of Uncertainty in Machine Learning – A Statisticians' View". arXiv: `2305.16703 [stat.ML]`.

Guan, Leying (2023). "Localized conformal prediction: A generalized inference framework for conformal prediction". *Biometrika* 110.1, pp. 33–50.

Guo, Chuan et al. (2017). "On Calibration of Modern Neural Networks". *The 34th International Conference on Machine Learning*. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 1321–1330.

Gupta, Kartik et al. (2021). "Calibration of Neural Networks using Splines". *International Conference on Learning Representations*.

Gustafsson, Fredrik K et al. (2020). "Energy-Based Models for Deep Probabilistic Regression". *Computer Vision – ECCV 2020*. Springer International Publishing, pp. 325–343.

Hallin, Marc and Miroslav Šiman (2017). "Multiple-output quantile regression". *Handbook of quantile regression*, pp. 185–207.

Hallin, Marc et al. (2021). "Distribution and quantile functions, ranks and signs in dimension d: A measure transportation approach". *The Annals of Statistics* 49 (2), pp. 1139–1165.

Han, Xing et al. (2022). "Split localized conformal prediction". *arXiv:2206.13092*.

Hastie, Trevor (2009). *The elements of statistical learning: data mining, inference, and prediction*.

Hendrickson, Arlo D and Robert J Buehler (1971). "Proper scores for probability forecasters". *The annals of mathematical statistics* 42 (6), pp. 1916–1921.

Hinton, Geoffrey, Simon Osindero, and Yee-Whye Teh (2006). "A fast learning algorithm for deep belief nets". *Neural computation* 18 (7), pp. 1527–1554.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). "Denoising diffusion probabilistic models". *Advances in neural information processing systems*.

Hofmann, Heike, Karen Kafadar, and Hadley Wickham (2011). *Letter-value plots: Boxplots for large data*. Research rep. had.co.nz.

Hollmann, Noah et al. (2022). "TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second". *The Eleventh International Conference on Learning Representations*.

Holm, Sture (1979). "A Simple Sequentially Rejective Multiple Test Procedure". *Scandinavian journal of statistics, theory and applications* 6 (2), pp. 65–70.

Horwitz, Eliahu and Yedid Hoshen (2022). "Conffusion: Confidence Intervals for Diffusion Models". arXiv: `2211.09795 [cs.CV]`.

Huang, Chin-Wei et al. (2021). "Convex Potential Flows: Universal Probability Distributions with Optimal Transport and Convex Optimization". *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Huang, Gao et al. (2017). "Snapshot Ensembles: Train 1, Get M for Free". *International Conference on Learning Representations*.

Hüllermeier, Eyke, Alireza Javanmardi, and David Stutz (2024). "Conformalized Credal Set Predictors". *Advances in neural information processing systems* 37, pp. 116987–117014.

Hüllermeier, Eyke and Willem Waegeman (2021). "Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods". *Machine learning* 110 (3), pp. 457–506.

Hyndman, Rob J (1996). "Computing and Graphing Highest Density Regions". *The American statistician* 50 (2), pp. 120–126.

Hyvärinen, Aapo (2005). "Estimation of Non-Normalized Statistical Models by Score Matching". *Journal of machine learning research: JMLR* 6 (24), pp. 695–709.

Ismail Fawaz, Hassan et al. (2019). "Deep learning for time series classification: a review". *Data mining and knowledge discovery* 33 (4), pp. 917–963.

Izbicki, Rafael, Gilson Shimizu, and Rafael Stern (2020). "Flexible distribution-free conditional predictive bands using density estimators". *The Twenty Third International Conference on Artificial Intelligence and Statistics*. Vol. 108. Proceedings of Machine Learning Research. PMLR, pp. 3068–3077.

Izbicki, Rafael, Gilson Shimizu, and Rafael B Stern (2022). "CD-split and HPD-split: Efficient Conformal Regions in High Dimensions". *Journal of machine learning research: JMLR* 23 (87), pp. 1–32.

Johansson, Ulf, Henrik Boström, and Tuwe Löfström (2021). "Investigating normalized conformal regressors". *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, pp. 01–08.

Johansson, Ulf et al. (2014). "Regression conformal prediction with random forests". *Machine learning* 97, pp. 155–176.

Johnstone, Chancellor and Eugene Ndiaye (2025). "Exact and Approximate Conformal Inference for Multi-Output Regression". *Fourteenth Symposium on Conformal and Probabilistic Prediction with Applications (COPA 2025)*. Fourteenth Symposium on Conformal and Probabilistic Prediction with Applications (COPA 2025). PMLR, pp. 153–172.

Jospin, Laurent Valentin et al. (2022). "Hands-On Bayesian Neural Networks—A Tutorial for Deep Learning Users". *IEEE Computational Intelligence Magazine* 17 (2), pp. 29–48.

Jung, Christopher et al. (2023). "Batch Multivalid Conformal Prediction". *International Conference on Learning Representations*.

Kadra, Arlind et al. (2021). "Well-tuned Simple Nets Excel on Tabular Datasets". *Advances in Neural Information Processing Systems* 34, pp. 23928–23941.

Kan, Kelvin et al. (2022). "Multivariate Quantile Function Forecaster". *The 25th International Conference on Artificial Intelligence and Statistics*. Vol. 151. Proceedings of Machine Learning Research. PMLR, pp. 10603–10621.

Karandikar, Archit et al. (2021). "Soft calibration objectives for neural networks". *Advances in neural information processing systems* 34, pp. 29768–29779.

Kendall, Alex and Y Gal (2017). "What uncertainties do we need in Bayesian deep learning for computer vision?" *Neural Information Processing Systems* abs/1703.04977.

Kim, Hyunho and Jong-Seok Lee (2023). "Scalable Monotonic Neural Networks". *The Twelfth International Conference on Learning Representations*.

Kim, Myung Jun, Léo Grinsztajn, and G Varoquaux (2024). "CARTE: pretraining and transfer for tabular learning". *International Conference on Machine Learning* abs/2402.16785, pp. 23843–23866.

Kingma, Diederik P and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kingma, Diederik P and Prafulla Dhariwal (2018). "Glow: Generative flow with invertible 1x1 convolutions". *Advances in neural information processing systems* abs/1807.03039.

Kingma, Diederik P and Max Welling (2014). "Auto-Encoding Variational Bayes". *International Conference on Learning Representations*.

Kiyani, Shayan, George J Pappas, and Hamed Hassani (2024). "Conformal Prediction with Learned Features". *International Conference on Machine Learning*.

Klein, Michal et al. (2025). "Multivariate Conformal Prediction using Optimal Transport". arXiv: `2502.03609 [stat.ML]`.

Klein, Nadja (2024). "Distributional regression for data analysis". *Annual review of statistics and its application* 11 (1), pp. 321–346.

Klein, Nadja, David J Nott, and Michael Stanley Smith (2021). "Marginally Calibrated Deep Distributional Regression". *Journal of computational and graphical statistics: a joint publication of American Statistical Association, Institute of Mathematical Statistics, Interface Foundation of North America* 30 (2), pp. 467–483.

Kobyzev, Ivan, Simon J D Prince, and Marcus A Brubaker (2021). "Normalizing Flows: An introduction and review of current methods". *IEEE transactions on pattern analysis and machine intelligence* 43 (11), pp. 3964–3979.

Koenker, Roger (2005). *Quantile regression*. Cambridge University Press.

Koenker, Roger and Gilbert Bassett Jr (1978). "Regression quantiles". *Econometrica: journal of the Econometric Society*, pp. 33–50.

Koenker, Roger and Kevin F Hallock (2001). "Quantile regression". *Journal of economic perspectives* 15.4, pp. 143–156.

Kohonen, Jukka and Jukka Suomela (2006). "Lessons learned in the challenge: Making predictions and scoring them". *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*. Vol. 95. Lecture notes in computer science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 95–116.

Krizhevsky, Alex, Vinod Nair, and Geoffrey Hinton (2014). "Cifar-10". *Online: http://www.cs.toronto.edu/kriz/cif*

Kuleshov, Volodymyr and Shachi Deshpande (2022). "Calibrated and Sharp Uncertainties in Deep Learning via Density Estimation". *The 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 11683–11693.

Kuleshov, Volodymyr, Nathan Fenner, and Stefano Ermon (2018). "Accurate Uncertainties for Deep Learning Using Calibrated Regression". *The 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 2796–2804.

Kull, Meelis and Peter Flach (2015). "Novel Decompositions of Proper Scoring Rules for Classification: Score Adjustment as Precursor to Calibration". *Machine Learning and Knowledge Discovery in Databases*. Springer International Publishing, pp. 68–85.

Kull, Meelis et al. (2019). "Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration". *Advances in neural information processing systems* 32.

Kumar, Ananya, Percy Liang, and Tengyu Ma (2019). "Verified Uncertainty Calibration". *Advances in neural information processing systems* abs/1909.10155.

Kumar, Aviral, Sunita Sarawagi, and Ujjwal Jain (2018). "Trainable Calibration Measures for Neural Networks from Kernel Mean Embeddings". *The 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 2805–2814.

Kumar, Srijan, Xikun Zhang, and Jure Leskovec (2019). "Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks". *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

Lakshminarayanan, Pritzel, et al. (2017). "Simple and scalable predictive uncertainty estimation using deep ensembles". *Advances in neural information processing systems*.

LeCun, Yann et al. (2006). "A tutorial on energy-based learning". *Predicting structured data* 1 (0).

Lei, Jing and Larry Wasserman (2014). "Distribution-free prediction bands for non-parametric regression". *Journal of the Royal Statistical Society. Series B, Statistical methodology* 76 (1), pp. 71–96.

Lei, Jing et al. (2018). "Distribution-Free Predictive Inference for Regression". *Journal of the American Statistical Association* 113 (523), pp. 1094–1111.

LeRoy, Benjamin and David Zhao (2021). "MD-split+: Practical local conformal inference in high dimensions". *arXiv:2107.03280*.

Li, Youjuan, Yufeng Liu, and Ji Zhu (2007). "Quantile regression in reproducing kernel Hilbert spaces". *Journal of the American Statistical Association* 102.477, pp. 255–268.

Liou, Lathan et al. (2024). "Assessing calibration and bias of a deployed machine learning malnutrition prediction model within a large healthcare system". *npj digital medicine* 7 (1), p. 149.

Lipman, Yaron et al. (2022). "Flow Matching for Generative Modeling". *The Eleventh International Conference on Learning Representations*.

Liu, Dong C and Jorge Nocedal (1989). "On the limited memory BFGS method for large scale optimization". *Mathematical programming* 45 (1-3), pp. 503–528.

Lu, Charles et al. (2023). "Federated Conformal Predictors for Distributed Uncertainty Quantification". *International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, pp. 22942–22964.

Luo, Rachel et al. (2022). "Local calibration: metrics and recalibration". *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. Vol. 180. Proceedings of Machine Learning Research. PMLR, pp. 1286–1295.

Makaremi, Saeed (2025). "A multi-output deep learning model for energy demand and port availability forecasting in EV charging infrastructure". *Energy (Oxford, England)* 317 (134582), p. 134582.

Marx, Charles, Sofian Zalouk, and Stefano Ermon (2023). "Calibration by distribution matching: Trainable kernel calibration metrics". *Advances in neural information processing systems* abs/2310.20211, pp. 25910–25928.

Marx, Charles et al. (2022). "Modular Conformal Calibration". *The 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 15180–15195.

Matheson, James E and Robert L Winkler (1976). "Scoring Rules for Continuous Probability Distributions". *Management science* 22 (10), pp. 1087–1096.

McCarthy, J (1956). "Measures of the value of information". *Proceedings of the National Academy of Sciences of the United States of America* 42 (9), pp. 654–655.

McCulloch, Warren S and Walter Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". *The Bulletin of mathematical biophysics* 5 (4), pp. 115–133.

McElfresh, Duncan C et al. (2023). "When do neural nets outperform boosted trees on tabular data?" *Neural Information Processing Systems* abs/2305.02997, pp. 76336–76369.

Messoudi, Soundouss, Sébastien Destercke, and Sylvain Rousseau (2021). "Copula-based conformal prediction for multi-target regression". *Pattern Recognition* 120, p. 108101.

– (2022). "Ellipsoidal conformal inference for Multi-Target Regression". *Conformal and Probabilistic Prediction with Applications*. Conformal and Probabilistic Prediction with Applications. PMLR, pp. 294–306.

Minderer, Matthias et al. (2021). "Revisiting the Calibration of Modern Neural Networks". *Advances in neural information processing systems* 34, pp. 15682–15694.

Moreno-Torres, Jose G et al. (2012). "A unifying view on dataset shift in classification". *Pattern recognition* 45 (1), pp. 521–530.

Müller, Thomas et al. (2019). "Neural Importance Sampling". *ACM transactions on graphics* 38 (5), pp. 1–19.

Murphy, Kevin P (2022). *Probabilistic Machine Learning: An introduction*. MIT Press.

– (2023). *Probabilistic machine learning: Advanced topics*. MIT Press.

Muschinski, Thomas et al. (2022). "Cholesky-based multivariate Gaussian regression". *Econometrics and statistics* 29, pp. 261–281.

Nalisnick, Eric et al. (2018). "Do Deep Generative Models Know What They Don't Know?" *International Conference on Learning Representations*.

Nouretdinov, Ilia, Thomas Melluish, and Volodya Vovk (2001). "Ridge regression confidence machine". *ICML*, pp. 385–392.

Omi, Takahiro, Naonori Ueda, and Kazuyuki Aihara (2019). "Fully Neural Network based Model for General Temporal Point Processes". *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*.

Oord, Aäron van den et al. (2016). "Conditional image generation with PixelCNN decoders". *Advances in neural information processing systems* abs/1606.05328.

Ovadia, Yaniv et al. (2019). "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift". *Advances in neural information processing systems* 32.

Paindaveine, Davy and Miroslav Šiman (2011). "On directional multiple-output quantile regression". *Journal of multivariate analysis* 102 (2), pp. 193–212.

Papadopoulos, Harris, Alex Gammerman, and Vladimir Vovk (2008). "Normalized nonconformity measures for regression Conformal Prediction". *The 26th IASTED International Conference on Artificial Intelligence and Applications* (Innsbruck, Austria). AIA '08. USA: ACTA Press, pp. 64–69.

Papadopoulos, Harris and Haris Haralambous (2011). "Reliable prediction intervals with regression neural networks". *Neural Networks* 24.8, pp. 842–851.

Papadopoulos, Harris et al. (2002). "Inductive Confidence Machines for Regression". *Machine Learning: ECML 2002*. Springer Berlin Heidelberg, pp. 345–356.

Papamakarios, George, Theo Pavlakou, et al. (2017). "Masked autoregressive flow for density estimation". *Advances in neural information processing systems*.

Papamakarios, George et al. (2021). "Normalizing Flows for Probabilistic Modeling and Inference". *Journal of Machine Learning Research* 22 (57), pp. 1–64.

Park, Ji Won, Robert Tibshirani, and Kyunghyun Cho (2024). *Semiparametric conformal prediction*. arXiv:2411.02114.

Pearce, Tim et al. (2018). "High-Quality Prediction Intervals for Deep Learning: A Distribution-Free, Ensembled Approach". *The 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 4075–4084.

Pereyra, Gabriel et al. (2017). "Regularizing Neural Networks by Penalizing Confident Output Distributions". arXiv: 1701.06548 [cs.NE].

Perini, Lorenzo et al. (2024). "Uncertainty-aware evaluation of auxiliary anomalies with the expected anomaly posterior". *Transactions on Machine Learning Research* 2025.

Pinson, Pierre and Renate Hagedorn (2012). "Verification of the ECMWF ensemble forecasts of wind speed against analyses and observations". *Meteorological Applications* 19 (4), pp. 484–500.

Pinson, Pierre and Julija Tastu (2013). "Discrimination ability of the Energy score".

Plassier, Vincent et al. (2025a). "Probabilistic Conformal Prediction with Approximate Conditional Validity". *The Thirteenth International Conference on Learning Representations.*

Plassier, Vincent et al. (2025b). "Rectifying conformity scores for better conditional coverage". *The 42nd International Conference on Machine Learning.* PMLR.

Platt, John et al. (1999). "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods". *Advances in large margin classifiers* 10 (3), pp. 61–74.

Pleiss, Geoff et al. (2017). "On Fairness and Calibration". *Advances in neural information processing systems* 30.

Popordanoska, Teodora and Matthew Blaschko (2025). "Advancing Calibration in Deep Learning: Theory, Methods, and Applications".

Popordanoska, Teodora, Raphael Sayer, and Matthew Blaschko (2022). "A consistent and differentiable lp canonical calibration error estimator". *Advances in neural information processing systems.*

Price, Ilan et al. (2025). "Probabilistic weather forecasting with machine learning". *Nature* 637 (8044), pp. 84–90.

Rajkomar, Alvin et al. (2018). "Scalable and accurate deep learning with electronic health records". *npj digital medicine* 1 (1), p. 18.

Reiß, Markus, Yves Rozenholc, and Charles-Andre Cuenod (2009). "Pointwise adaptive estimation for robust and quantile regression". *arXiv:0904.0543.*

Rezende, Danilo and Shakir Mohamed (2015). "Variational Inference with Normalizing Flows". *The 32nd International Conference on Machine Learning.* Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 1530–1538.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra (2014). "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". *The 31st International Conference on Machine Learning.* Vol. 32. Proceedings of Machine Learning Research. Beijing, China: PMLR, pp. 1278–1286.

Romano, Yaniv, Evan Patterson, and Emmanuel Candes (2019). "Conformalized quantile regression". *Advances in neural information processing systems.*

Romano, Yaniv, Matteo Sesia, and Emmanuel Candès (2020). "Classification with valid and adaptive coverage". *Advances in neural information processing systems* 33, pp. 3581–3591.

Rozner, Amit et al. (2024). "Anomaly Detection with Variance Stabilized Density Estimation". *The 40th Conference on Uncertainty in Artificial Intelligence.*

Rumelhart, David E, Geoffrey Hinton, and Ronald J Williams (1986). "Learning representations by back-propagating errors". *Nature* 323 (6088), pp. 533–536.

Russell, Stuart J and Peter Norvig (2020). *Artificial Intelligence: A Modern Approach (4th Edition).* Pearson.

Sadinle, Mauricio, Jing Lei, and Larry Wasserman (2019). "Least ambiguous set-valued classifiers with bounded error levels". *Journal of the American Statistical Association* 114 (525), pp. 223–234.

Sahoo, Roshni et al. (2021). "Reliable decisions with threshold calibration". *Advances in neural information processing systems.*

Schotter, Andrew and Isabel Trevino (2014). "Belief elicitation in the laboratory". *Annual review of economics* 6 (1), pp. 103–128.

Schuhmann, Christoph et al. (2022). "LAION-5B: An open large-scale dataset for training next generation image-text models". *Neural Information Processing Systems* abs/2210.08402, pp. 25278–25294.

Scott, David W (1992). "Multivariate Density Estimation". *Wiley Series in Probability and Statistics.*

Sensoy, Kaplan, et al. (2018). "Evidential deep learning to quantify classification uncertainty". *Advances in neural information processing systems.*

Sesia, Matteo and Yaniv Romano (2021). "Conformal prediction using conditional histograms". *Advances in neural information processing systems.*

Setiawan, Karli Eka, Hafizh Ash Shiddiqi, and Pandu Wicaksono (2024). "Multi-output machine learning regression for climate prediction: a comparative study of precipitation and temperature forecasting in Jakarta and East Kalimantan, Indonesia". *Communications in mathematical biology and neuroscience* 2024 (0), Article ID 97.

Shafer, Glenn and Vladimir Vovk (2008). "A Tutorial on Conformal Prediction". *Journal of machine learning research: JMLR.*

Shao, Chenze et al. (2024). "Language Generation with Strictly Proper Scoring Rules". *Forty-first International Conference on Machine Learning.*

Shchur, Oleksandr, Marin Biloš, and Stephan Günnemann (2020). "Intensity-Free Learning of Temporal Point Processes". *Proceedings of the 2020 International Conference on Learning Representations (ICLR).*

Shchur, Oleksandr et al. (2021). "Neural Temporal Point Processes: A Review". *Proceedings of 13th Joint Conference on Artificial Intelligence (IJCAI).*

Shen, Guohao et al. (2024). "Nonparametric Estimation of Non-Crossing Quantile Regression Process with Deep ReQU Neural Networks". *Journal of Machine Learning Research* 25.88, pp. 1–75.

Shen, Maohao et al. (2024). "Are Uncertainty Quantification Capabilities of Evidential Deep Learning a Mirage?" *The Thirty-eighth Annual Conference on Neural Information Processing Systems.*

Shwartz-Ziv, Ravid and Amitai Armon (2022). "Tabular data: Deep learning is not all you need". *An international journal on information fusion* 81, pp. 84–90.

Si, Phillip et al. (2023). "Semi-Autoregressive Energy Flows: Towards Determinant-Free Training of Normalizing Flows".

Sohl-Dickstein, Jascha et al. (2015). "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". *The 32nd International Conference on Machine Learning.* Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 2256–2265.

Song, Hao et al. (2019). "Distribution calibration for regression". *The 36th International Conference on Machine Learning.* Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 5897–5906.

Song, Yang and Diederik P Kingma (2021). "How to Train Your Energy-Based Models". arXiv: `2101.03288 [cs.LG]`.

Song, Yang et al. (2020). "Score-Based Generative Modeling through Stochastic Differential Equations". *International Conference on Learning Representations.*

Spokoiny, Vladimir, Weining Wang, and Wolfgang Karl Härdle (2013). "Local quantile regression". *Journal of Statistical Planning and Inference* 143.7, pp. 1109–1129.

Stankeviciute, Kamile, Ahmed M Alaa, and Mihaela van der Schaar (2021). "Conformal Time-series Forecasting". *Advances in neural information processing systems* 34.

Stimper, Vincent, Bernhard Schölkopf, and Jose Miguel Hernandez-Lobato (2022). "Resampling Base Distributions of Normalizing Flows". *International Conference on Artificial Intelligence and Statistics*. International Conference on Artificial Intelligence and Statistics. PMLR, pp. 4915–4936.

Stutz, David et al. (2022). "Learning Optimal Conformal Classifiers". *International Conference on Learning Representations*.

Sun, Jiankai et al. (2023). "Conformal prediction for uncertainty-aware planning with diffusion dynamics model". *Neural Information Processing Systems* 36, pp. 80324–80337.

Sun, Sophia Huiwen and Rose Yu (2024). "Copula Conformal prediction for multi-step time series prediction". *The Twelfth International Conference on Learning Representations*.

Tagasovska and Lopez-Paz (2019). "Single-model uncertainties for deep learning". *Advances in neural information processing systems*.

Takeuchi, Ichiro et al. (2006). "Nonparametric quantile estimation." *Journal of machine learning research* 7.7.

Teneggi, Jacopo et al. (2023). "How to Trust Your Diffusion Model: A Convex Optimization Approach to Conformal Risk Control".

Thiagarajan, Jayaraman J et al. (2020). "Designing accurate emulators for scientific processes using calibration-driven deep models". *Nature communications* 11 (1), p. 5622.

Thurin, Gauthier, Kimia Nadjahi, and Claire Boyer (2025). "Optimal transport-based conformal prediction". *Forty-second International Conference on Machine Learning*.

Tibshirani, Ryan J et al. (2019). "Conformal Prediction Under Covariate Shift". *Advances in neural information processing systems* 32, pp. 2526–2536.

Timans, Alexander et al. (2025). "Max-Rank: Efficient Multiple Testing for Conformal Prediction". *International Conference on Artificial Intelligence and Statistics*. International Conference on Artificial Intelligence and Statistics. PMLR, pp. 3898–3906.

Trivedi, Rakshit et al. (2019). "Representation Learning over Dynamic Graphs". *Proceedings of the 2019 International Conference on Learning Representations (ICLR)*.

Trotta, Belinda et al. (2024). "RainForests: a machine-learning approach to calibrating NWP precipitation forecasts". *Weather and forecasting* 39 (11), pp. 1715–1732.

Tsoumakas, Grigorios et al. (2011). "MULAN: A Java Library for Multi-Label Learning". *Journal of machine learning research: JMLR* 12 (71), pp. 2411–2414.

Tsyplakov, Alexander (2013). *Evaluation of Probabilistic Forecasts: Proper Scoring Rules and Moments*. Research rep. 45186.

Utpala, Saiteja and Piyush Rai (2020). "Quantile Regularization: Towards Implicit Calibration of Regression Models". arXiv: `2002.12860 [cs.LG]`.

Van Breugel, Boris and Mihaela Van Der Schaar (2024). "Position: Why Tabular Foundation Models Should Be a Research Priority". *International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, pp. 48976–48993.

Vapnik, Vladimir N (1999). "An overview of statistical learning theory". *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council* 10 (5), pp. 988–999.

Vasicek, Oldrich (1976). "A Test for Normality Based on Sample Entropy". *Journal of the Royal Statistical Society. Series B, Statistical methodology* 38 (1), pp. 54–59.

Vincent, Pascal (2011). "A connection between score matching and denoising autoencoders". *Neural computation* 23 (7), pp. 1661–1674.

Vovk, Vladimir (2012). "Conditional validity of inductive conformal predictors". *Asian conference on machine learning*. PMLR, pp. 475–490.

– (2025). "Inductive randomness predictors: beyond conformal". *Proceedings of Machine Learning Research* 266, pp. 1–28.

Vovk, Vladimir, Alexander Gammerman, and Craig Saunders (1999). "Machine-Learning Applications of Algorithmic Randomness". *The Sixteenth International Conference on Machine Learning*. ICML '99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 444–453.

Vovk, Vladimir, Alexander Gammerman, and Glenn Shafer (2005). *Algorithmic Learning in a Random World*. Springer International Publishing. 26 pp.

Vovk, Vladimir, Ilia Nouretdinov, and Alex Gammerman (2009). "On-line predictive linear regression". *The Annals of Statistics*, pp. 1566–1590.

Vovk, Vladimir et al. (2019). "Nonparametric predictive distributions based on conformal prediction". *Machine learning* 108 (3), pp. 445–474.

Vovk, Vladimir et al. (2020). "Conformal calibrators". *Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications*. Vol. 128. Proceedings of Machine Learning Research. PMLR, pp. 84–99.

Waghmare, Kartik and Johanna Ziegel (2025). "Proper scoring rules for estimation and forecast evaluation". arXiv: `2504.01781 [math.ST]`.

Wang, Cheng (2025). "Calibration in deep learning: A survey of the state-of-the-art". arXiv: `2308.01222 [cs.LG]`.

Wang, Deng-Bao, Lei Feng, and Min-Ling Zhang (2021). "Rethinking Calibration of Deep Neural Networks: Do Not Be Afraid of Overconfidence". *Advances in neural information processing systems* 34, pp. 11809–11820.

Wang, Jun et al. (2023). *Conformal Temporal Logic Planning using Large Language Models: Knowing When to Do What and When to Ask for Help*. arXiv:2309.10092 [cs].

Wang, Zhendong et al. (2023). "Probabilistic Conformal Prediction Using Conditional Random Samples". *International Conference on Artificial Intelligence and Statistics*. International Conference on Artificial Intelligence and Statistics. PMLR, pp. 8814–8836.

Wilcoxon, Frank (1945). "Individual Comparisons by Ranking Methods". *Biometrics Bulletin* 1 (6), pp. 80–83.

Xiao, Jiancong et al. (2025). "Restoring Calibration for Aligned Large Language Models: A Calibration-Aware Fine-Tuning Approach". *Forty-second International Conference on Machine Learning*.

Xie, Ran, Rina Foygel Barber, and Emmanuel J. Candès (2024). "Boosted Conformal Prediction Intervals". *Neural Information Processing Systems*.

Yang, Meicheng et al. (2024). "Development and validation of an interpretable conformal predictor to predict sepsis mortality risk: Retrospective cohort study". *Journal of medical internet research* 26 (1), e50369.

Ye, H et al. (2021). "Towards a theoretical framework of out-of-distribution generalization". *Advances in Neural Information Processing Systems* 34, pp. 23519–23531.

Yoon, Hee Suk et al. (2023). "ESD: Expected Squared Difference as a Tuning-Free Trainable Calibration Measure". *The Eleventh International Conference on Learning Representations*.

Zaffran, Margaux et al. (2022). "Adaptive Conformal Predictions for Time Series". *The 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 25834–25866.

Zhai, Shuangfei et al. (2025). "Normalizing Flows are Capable Generative Models". *Forty-second International Conference on Machine Learning*.

Zhang, Hongyi et al. (2018). "mixup: Beyond Empirical Risk Minimization". *International Conference on Learning Representations*.

Zhao, Shengjia, Tengyu Ma, and Stefano Ermon (2020). "Individual Calibration with Randomized Forecasting". *The 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 11387–11397.

Zhao, Shengjia et al. (2021). "Calibrating predictions to decisions: A novel approach to multi-class calibration". *Advances in neural information processing systems* abs/2107.05719, pp. 22313–22324.

Zhou, Tianhui et al. (2021). "Estimating Uncertainty Intervals from Collaborating Networks". *Journal of machine learning research: JMLR* 22.

Zhou, Xiaofan et al. (2025). "Conformal prediction: A data perspective". *ACM computing surveys*.

Zhou, Yanfei, Lars Lindemann, and Matteo Sesia (2024). "Conformalized adaptive forecasting of heterogeneous trajectories". *The 41st International Conference on Machine Learning*. ICML'24. Vienna, Austria: JMLR.org.

Ziegel, Johanna and Tilmann Gneiting (2014). "Copula calibration". *Electronic journal of statistics* 8 (2), pp. 2619–2638.

Appendices

APPENDIX A

Datasets

Tables A.1 and A.2 summarize the characteristics of the single-output and multi-output tabular datasets considered in this study. For each dataset, we report the total size $|\mathcal{D}|$, the number of input variables p, and the number of output variables d. To identify datasets that may be less suitable for regression, we also compute the proportion of instances in $\mathcal{D}$ whose outcome belongs to the top κ most frequent values, as described in Section D.1.4. This measure is reported for $\kappa = 1$ and $\kappa = 10$, with values greater than 0.5 highlighted in bold.

A.1. Single-Output Tabular Regression Datasets

Table A.1 presents the single-output datasets used in Chapters 3 and 4. As detailed in Section 3.4, these datasets are gathered from different repositories: Grinsztajn et al. (2022) (`oml_297` and `oml_299`), Gijsbers et al. (2019) (`oml_269`), and Dua and Graff (2017) (`uci`).

Splitting. In Chapters 3 and 4, we shuffle each dataset and split it into training, validation, calibration, and test sets using proportions of 65%, 10%, 15%, and 10%, respectively. Before splitting, we downsample each dataset (without replacement) to ensure that the training set contains at most 50,000 instances, corresponding to the large-data regime defined in Grinsztajn et al. (2022).

In Chapter 4, following Angelopoulos and Bates (2023), we observe that a large calibration set is not always necessary. For example, for $\alpha = 0.9$, Figure 2.9a shows that $\mathbb{P}(\hat{F}_{Y|X}(Y) \leqslant \alpha)$ is close to 0.9, typically between 0.88 and 0.9. Hence, we limit the calibration set to 2048 points and redistribute the remaining instances equally across the training, validation, and test sets.

Table A.1: Characteristics of all considered single-output tabular datasets ($d = 1$).

Group	Dataset	Abbrev.	$\lvert\mathcal{D}\rvert$	p	Proportion of top 1 most frequent values	Proportion of top 10 most frequent values
uci	CPU	CP1	209	7	0.043	0.297
oml_269	tecator	TEC	240	124	0.025	0.171
uci	Yacht	YAC	308	6	0.010	0.088
	MPG	MPG	392	7	0.051	0.362
oml_269	boston	BOS	506	22	0.032	0.142
uci	Energy	ENE	768	9	0.005	0.046
	Fish	FIS	908	6	0.004	0.031
	Concrete	CON	1030	8	0.005	0.036
oml_269	MIP-2016-regression	MIP	1090	111	0.006	0.043
	house_prices_nominal	HO2	1094	234	0.016	0.104
	socmob	SOC	1156	39	0.201	0.349
	Moneyball	MON	1232	18	0.009	0.080
uci	Airfoil	AI1	1503	5	0.002	0.015
	Crime	CRI	1993	102	0.052	0.380
oml_269	us_crime	US_	1993	101	0.052	0.380
	quake	QUA	2178	3	0.317	**0.998**
	space_ga	SPA	3107	6	0.001	0.004
oml_299	analcatdata_supreme	ANA	4052	12	**0.701**	**1.000**
oml_269	abalone	ABA	4177	10	0.165	**0.895**
oml_299	Mercedes_Benz _Greener_Manufacturing	MER	4209	735	0.002	0.015
oml_269	SAT11-HAND-runtime-regression	SAT	4440	118	0.007	0.061
	Santander_transaction _value	SAN	4459	3611	0.046	0.301
oml_297	wine_quality	WIN	6497	11	0.437	**1.000**
oml_269	colleges	COL	6695	34	0.009	0.029
oml_297	isolet	ISO	7797	613	0.038	0.385
uci	Kin8nm	KIN	8192	8	0.000	0.002
oml_297	cpu_act	CP2	8192	21	0.056	**0.507**
oml_299	visualizing_soil	VIS	8641	5	0.070	0.426
oml_269	topo_2_1	TOP	8885	252	0.005	0.038
oml_299	yprop_4_1	YPR	8885	82	0.005	0.038
uci	Power	POW	9568	4	0.001	0.008
oml_297	sulfur	SUL	10081	6	0.003	0.013
	Brazilian_houses	BRA	10692	8	0.004	0.015
uci	Naval	NAV	11934	17	0.038	0.385
oml_297	Ailerons	AIL	13750	33	0.135	**0.860**
	MiamiHousing2016	MIA	13932	13	0.014	0.115
	pol	POL	15000	26	**0.623**	**0.983**
	elevators	ELE	16599	16	0.151	**0.796**
	Bike_Sharing_Demand	BIK	17379	6	0.015	0.115
	fifa	FIF	18063	5	0.156	**0.693**
	california	CAL	20640	8	0.047	0.086
	superconduct	SUP	21263	79	0.007	0.053
	house_sales	HO3	21613	15	0.008	0.069
	house_16H	HO1	22784	16	0.137	0.170
oml_299	OnlineNewsPopularity	ONL	39644	73	0.058	0.353
uci	Protein	PRO	45730	9	0.006	0.010
oml_297	diamonds	DIA	53940	6	0.002	0.023
oml_269	Allstate_Claims_Severity	ALL	76924	477	0.001	0.001
	Yolanda	YOL	76924	100	0.076	**0.582**
oml_297	medical_charges	MED	76924	3	0.000	0.001
oml_299	SGEMM_GPU _kernel_performance	SGE	76924	15	0.000	0.003
	black_friday	BLA	76924	23	0.000	0.004
oml_297	nyc-taxi-green-dec-2016	NYC	76924	9	0.152	0.384
	year	YEA	76924	90	0.077	**0.582**
oml_269	Buzzinsocialmedia_Twitter	BUZ	76924	70	0.037	0.251
oml_299	particulate-matter -ukair-2017	PAR	76924	26	0.006	0.053
oml_269	Airlines_DepDelay_10M	AI2	76924	5	0.189	**0.625**

A.2. Multi-Output Tabular Regression Datasets

Table A.2 presents the multi-output datasets considered in Chapters 5 to 7 and gathered from other studies (Tsoumakas et al., 2011; Feldman et al., 2023; Z. Wang et al., 2023; Barrio et al., 2024; Camehl et al., 2024). In Chapters 5 and 6, we only include datasets with at least 7000 instances. Datasets introduced in Cevid et al. (2022) are used starting from Chapter 7.

Splitting. As for the univariate case, we first downsample each dataset (without replacement) to a maximum of 50,000 instances. In Chapters 5 and 6, we allocate 2048 points to calibration, and split the remainder into 55% for training, 15% for validation, and 30% for testing. In Chapter 7, we follow Y. Chung et al. (2024), using a split of 65% for training, 20% for validation, and 15% for testing, with the validation set serving also as the calibration set.

Table A.2: Characteristics of all considered multi-output tabular datasets.

Group	Dataset	Abbrev.	$\lvert\mathcal{D}\rvert$	p	d	Proportion of top 1 most frequent values	Proportion of top 10 most frequent values
Tsoumakas et al. (2011)	slump	SLU	103	7	3	0.010	0.097
	edm	EDM	154	16	2	0.390	**1.000**
	atp7d	AT2	296	355	6	0.030	0.230
	sf1	SF1	323	31	3	**0.820**	**0.988**
	oes97	OE2	334	263	16	0.003	0.030
	atp1d	AT1	337	354	6	0.018	0.116
	jura	JUR	359	15	3	0.003	0.028
	oes10	OE1	403	298	16	0.002	0.025
	enb	ENB	768	3	2	0.005	0.029
	wq	WQ	1060	16	14	0.006	0.039
	sf2	SF2	1066	31	3	**0.811**	**0.986**
	scpf	SCP	1137	8	3	0.383	**0.659**
Barrio et al. (2024)	ansur2	ANS	1986	1	2	0.003	0.024
Camehl et al. (2024)	households	HO2	7207	14	4	0.000	0.001
Tsoumakas et al. (2011)	scm20d	SC2	8966	60	16	0.000	0.001
	rf1	RF1	9005	64	8	0.000	0.002
	scm1d	SC1	9803	279	16	0.000	0.001
Cevid et al. (2022)	air	AIR	10000	15	6	0.000	0.001
	births2	BI2	10000	24	4	0.010	0.083
	births1	BI1	10000	23	2	0.005	0.043
	wage	WAG	10000	78	2	0.019	0.137
Feldman et al. (2023)	meps_21	ME3	15656	138	2	0.121	0.490
	meps_19	ME1	15785	138	2	0.133	**0.505**
	meps_20	ME2	17541	138	2	0.116	0.483
	house	HO1	21613	17	2	0.000	0.001
	bio	BIO	45730	8	2	0.000	0.002
	blog_data	BLO	50000	269	2	0.021	0.173
Barrio et al. (2024)	calcofi	CAL	50000	1	2	0.000	0.001
Z. Wang et al. (2023)	taxi	TAX	50000	4	2	0.000	0.000

A.3. Event Sequence Datasets

In Section 5.7.4, our evaluation is based on five marked event sequence datasets from real-world scenarios, which have been previously considered in neural TPP research. Section F.3.2 also reports results on additional datasets. Below, we provide a description of these datasets:

- **LastFM** (S. Kumar et al., 2019): This dataset comprises records of individuals' song-listening events over time, with each song's artist serving as the mark.

- **MOOC** (S. Kumar et al., 2019): It captures the activities of students on an online course platform, where the mark denotes the specific type of activity, such as watching a video.
- **Reddit** (S. Kumar et al., 2019): This consists of sequences of posts made to various subreddits. Each sequence is associated with a user, and the mark is the post that the user responds to.
- **Retweets** (Omi et al., 2019): It includes sequences of retweets occurring after an initial tweet over time. Each sequence is linked to a specific tweet, with the mark being a category assigned to the retweeter (small, medium, or large retweeter).
- **Stack Overflow** (N. Du et al., 2016): This dataset records the badges awarded to users on Stack Overflow. Each user is assigned a sequence, and the type of badge received is used as the mark.
- **Github** (Trivedi et al., 2019): Records of developers' actions on the open-source platform Github. A sequence refers to a developer, and the marks correspond to the action being performed.
- **MIMIC2** (N. Du et al., 2016) Electronic health records (EHR) of patients in an intensive care units for seven years. A sequence corresponds to a patient, and the marks are the types of diseases.
- **Wikipedia** (S. Kumar et al., 2019) Records of edits made to Wikipedia pages. Each sequence is a page, and the marks refer to the users that made the edits.

Table A.3: Real-world Datasets statistics

	#Seq.	#Events	Mean Length	Max Length	Min Length	#Marks
LastFM	856	193441	226.0	6396	2	50
MOOC	7047	351160	49.8	416	2	50
Reddit	4278	238734	55.8	941	2	50
Retweets	12000	1309332	109.1	264	50	3
Stack Overflow	7959	569688	71.6	735	40	22
Github	173	20656	119.4	4698	3	8
MIMIC2	599	1812	3.0	32	2	43
Wikipedia	590	30472	51.6	1163	2	50

APPENDIX B

Supplementary Material for Chapter 2

B.1. Closed-form scoring rules for Gaussian mixtures

This section details closed-form expressions for the strictly proper scoring rules discussed in Section 2.3.2 assuming that the predictive distribution is a mixture of multivariate Gaussians.

Setup. Consider a Gaussian mixture density

$$\hat{f}(y) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}\left(y; \hat{\mu}_m, \hat{L}_m \hat{L}_m^\top\right), \quad y \in \mathbb{R}^d,$$

with mixture weights $(\hat{\pi}_m)_{m=1}^M$, means $(\hat{\mu}_m)_{m=1}^M$, and Cholesky matrices $\left(\hat{L}_m\right)_{m=1}^M$. Let $\hat{\Sigma}_m = \hat{L}_m \hat{L}_m^\top$ and $\hat{\Lambda}_m = \hat{\Sigma}_m^{-1}$. The expression for the QS, SS and KS are based on the following identity:

$$\int_{\mathbb{R}^d} \mathcal{N}\left(y'; \hat{\mu}_m, \hat{\Sigma}_m\right) \mathcal{N}\left(y'; \hat{\mu}_j, \hat{\Sigma}_j\right) dy' = \mathcal{N}\left(\hat{\mu}_m; \hat{\mu}_j, \hat{\Sigma}_m + \hat{\Sigma}_j\right).$$

Quadratic score (QS).

$$S_{\mathrm{QS}}(\hat{P}, y) = \left\|\hat{f}\right\|_2^2 - 2\hat{f}(y) = \sum_{m=1}^{M}\sum_{j=1}^{M} \hat{\pi}_m \hat{\pi}_j \mathcal{N}\left(\hat{\mu}_m; \hat{\mu}_j, \hat{\Sigma}_m + \hat{\Sigma}_j\right) - 2\hat{f}(y).$$

Spherical score (SS).

$$S_{\mathrm{PS}}(\hat{P}, y) = -\frac{\hat{f}(y)}{\left\|\hat{f}\right\|_2} = -\frac{\hat{f}(y)}{\left(\sum_{m=1}^{M}\sum_{j=1}^{M} \hat{\pi}_m \hat{\pi}_j \mathcal{N}\left(\hat{\mu}_m; \hat{\mu}_j, \hat{\Sigma}_m + \hat{\Sigma}_j\right)\right)^{1/2}}.$$

Table B.1: Computational complexities for evaluating closed-form strictly proper scoring rules from Section B.1. We consider the general case with full covariance or the special case of diagonal covariance.

Scoring rule	**Full covariance**	**Diagonal covariance**
NLL	$O(Md^3)$	$O(Md)$
QS	$O(M^2d^3)$	$O(M^2d)$
SS	$O(M^2d^3)$	$O(M^2d)$
HS	$O(Md^2)$	$O(Md)$
KS-RBF	$O(M^2d^3)$	$O(M^2d)$

Hyvärinen score (HS). Define the responsibilities and component score vectors at y:

$$r_m(y) = \frac{\hat{\pi}_m \mathcal{N}\left(y; \hat{\mu}_m, \hat{\Sigma}_m\right)}{\hat{f}(y)}, \quad a_m(y) = \hat{\Lambda}_m \left(\hat{\mu}_m - y\right), \quad \bar{a}(y) = \sum_{m=1}^{M} r_m(y) a_m(y) = \nabla \log \hat{f}(y).$$

Then

$$\Delta \log \hat{f}(y) = \sum_{m=1}^{M} r_m(y) \left(\|a_m(y)\|^2 - \operatorname{tr}\left(\hat{\Lambda}_m\right)\right) - \|\bar{a}(y)\|^2,$$

and the Hyvärinen score is

$$S_{\mathrm{HS}}(\hat{P}, y) = \Delta \log \hat{f}(y) + \tfrac{1}{2} \left\| \nabla \log \hat{f}(y) \right\|^2 = \sum_{m=1}^{M} r_m(y) \left(\|a_m(y)\|^2 - \operatorname{tr}\left(\hat{\Lambda}_m\right)\right) - \tfrac{1}{2} \|\bar{a}(y)\|^2.$$

Kernel score with RBF kernel (KS-RBF). For $k(y, y') = \exp\left(-\frac{\|y-y'\|^2}{2\ell^2}\right)$ with $\ell > 0$,

$$\mathbb{E}_{\hat{Y},\tilde{Y}\sim\hat{P}}\left[k\left(\hat{Y}, \tilde{Y}\right)\right] = (2\pi\ell^2)^{\frac{d}{2}} \sum_{m=1}^{M}\sum_{j=1}^{M} \hat{\pi}_m \hat{\pi}_j \mathcal{N}\left(\hat{\mu}_m; \hat{\mu}_j, \hat{\Sigma}_m + \hat{\Sigma}_j + \ell^2 I\right),$$

$$\mathbb{E}_{\hat{Y}\sim\hat{P}}\left[k\left(\hat{Y}, y\right)\right] = (2\pi\ell^2)^{\frac{d}{2}} \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}\left(y; \hat{\mu}_m, \hat{\Sigma}_m + \ell^2 I\right),$$

and thus

$$\begin{aligned} &S_{\mathrm{KS}}^{\mathrm{RBF}}(\hat{P}, y) \\ &= (2\pi\ell^2)^{\frac{d}{2}} \left[\sum_{m=1}^{M}\sum_{j=1}^{M} \hat{\pi}_m \hat{\pi}_j \mathcal{N}\left(\hat{\mu}_m; \hat{\mu}_j, \hat{\Sigma}_m + \hat{\Sigma}_j + \ell^2 I\right) - 2 \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}\left(y; \hat{\mu}_m, \hat{\Sigma}_m + \ell^2 I\right) \right]. \end{aligned}$$

Computational complexities Table B.1 summarizes the computational costs. For full covariance matrices, assuming that covariance matrices are parameterized by their Cholesky factor, evaluating a multivariate Gaussian density requires $O(d^3)$ time due to determinant computation, while the quadratic form $(y-\hat{\mu}_m)^\top \hat{\Lambda}_m (y-\hat{\mu}_m)$ costs $O(d^2)$. With diagonal covariances, each Gaussian evaluation reduces to $O(d)$. Assuming full covariance matrices, the HS is the most efficient, followed by the NLL, and then the QS, SS, and KS-RBF. With diagonal covariances, the HS and NLL are fastest, followed by the QS, SS, and KS-RBF.

B.2. Formalization and Proofs of Auto-Calibration Properties

This section formalizes auto-calibration and provides proofs for the theorems and lemmas in Section 2.4.4. We use Chapter IV of Cinlar (2009) as reference.

B.2.1 Preliminaries

Assume $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space and $(\mathcal{Y}, \mathcal{B})$ is a standard Borel space. Let $\mathcal{P}(\mathcal{Y})$ be the set of all probability measures on $(\mathcal{Y}, \mathcal{B})$. A *random probability measure on* $(\mathcal{Y}, \mathcal{B})$ is a measurable map $P : \Omega \to \mathcal{P}(\mathcal{Y})$. Equivalently, it is a Markov kernel $P : \Omega \times \mathcal{B} \to [0, 1]$ such that for each $A \in \mathcal{B}$ the map $\omega \mapsto P(\omega, A)$ is $\mathcal{F}$-measurable and, for each $\omega \in \Omega$, the map $A \mapsto P(\omega, A)$ is a probability measure. When $\mathcal{Y}$ is standard Borel, these two descriptions agree; we identify them via

$$P(\omega, A) = P(\omega)(A).$$

Let $X : (\Omega, \mathcal{F}) \to (\mathcal{X}, \mathcal{A})$ and $Y : (\Omega, \mathcal{F}) \to (\mathcal{Y}, \mathcal{B})$ be random variables. In particular, a regular conditional distribution of Y given X, denoted $P_{Y|X}$, is a random probability measure on $(\mathcal{Y}, \mathcal{B})$.

Definition 13 (Scoring Rule and Divergence)**.** A *scoring rule* is a measurable function $S : \mathcal{P}(\mathcal{Y}) \times \mathcal{Y} \to \bar{\mathbb{R}}$. We extend S to $\mathcal{P}(\mathcal{Y}) \times \mathcal{P}(\mathcal{Y})$ by

$$S(\hat{P}, P) := \int_{\mathcal{Y}} S(\hat{P}, y) dP(y).$$

A scoring rule is *proper* if for all $\hat{P}, P \in \mathcal{P}(\mathcal{Y})$, $S(P, P) \leqslant S(\hat{P}, P)$, and *strictly proper* if equality holds only for $\hat{P} = P$. The associated *generalized entropy* is $H(P) = S(P, P)$ and the *divergence* is $D(\hat{P}, P) := S(\hat{P}, P) - H(P)$. Throughout, assume all expectations/integrals below are finite.

Definition 14 (Conditional Expectation)**.** Let $\mathcal{G} \subseteq \mathcal{F}$ be a sub-σ-algebra. For an integrable real-valued random variable Z, the *conditional expectation* $\mathbb{E}[Z \mid \mathcal{G}]$ is the (a.s. unique) $\mathcal{G}$-measurable random variable Z' such that

$$\int_G Z \, d\mathbb{P} = \int_G Z' \, d\mathbb{P} \qquad \text{for all } G \in \mathcal{G}.$$

Intuitively, a σ-algebra encodes information. If $\mathcal{G} \subseteq \mathcal{F}$, then $\mathcal{G}$ is a coarser view than $\mathcal{F}$. A $\mathcal{G}$-measurable function can only depend on what is visible under $\mathcal{G}$ and therefore can encode *less* information than an $\mathcal{F}$-measurable one. The random variable $Z' = \mathbb{E}[Z \mid \mathcal{G}]$ is the best $\mathcal{G}$-measurable summary of Z.

Now let P be a random probability measure on $(\mathcal{Y}, \mathcal{B})$. Its conditional expectation w.r.t. $\mathcal{G}$ is defined setwise by

$$(\mathbb{E}[P \mid \mathcal{G}])(A) := \mathbb{E}[P(A) \mid \mathcal{G}], \qquad A \in \mathcal{B}.$$

Equivalently, for any $\mathcal{G} \otimes \mathcal{B}$-measurable $g : \Omega \times \mathcal{Y} \to \mathbb{R}$ with

$$\mathbb{E}\left[\int_{\mathcal{Y}} |g(\omega, y)| \, P(\omega, dy)\right] < \infty,$$

we have

$$\int_{\mathcal{Y}} g(\omega, y) \; (\mathbb{E}[P \mid \mathcal{G}])(\omega, dy) = \mathbb{E}\left[\int_{\mathcal{Y}} g(\omega, y) \, P(\omega, dy) \,\middle|\, \mathcal{G}\right] \quad \text{a.s.}$$

This equivalence is justified by Theorem 2.10 (existence of regular conditional distributions) and Theorem 2.19 (disintegration property) in Cinlar (2009), since $(\mathcal{Y}, \mathcal{B})$ is standard Borel.

B.2.2 Auto-Calibration

Definition 15 (Auto-calibration)**.** A predictive distribution P taking values in $\mathcal{P}(\mathcal{Y})$ is *auto-calibrated* (w.r.t. Y) if for all $A \in \mathcal{B}$,

$$\mathbb{P}(Y \in A \mid \sigma(P)) = P(A) \quad \text{a.s.}$$

Equivalently, $\mathbb{P}(Y \in \cdot \mid \sigma(P)) = P$ a.s. (in what follows we always condition on σ-algebras, writing $\sigma(P)$ rather than P). Intuitively, $\sigma(P)$ is the information revealed by the predictive distribution. Auto-calibration requires that, once we condition on exactly this information and nothing more, the conditional law of Y matches the announced P.

Proposition 5 (Auto-calibrated version of any predictive distribution). Let $\hat{P}$ be a random probability measure and let $\bar{P}$ be a regular conditional probability of Y given $\sigma(\hat{P})$:

$$\bar{P}(A) = \mathbb{P}(Y \in A \mid \sigma(\hat{P})) \text{ for } A \in \mathcal{B}.$$

Then $\bar{P}$ is auto-calibrated.

Proof. Since $\bar{P}$ is $\sigma(\hat{P})$-measurable, we have $\sigma(\bar{P}) \subseteq \sigma(\hat{P})$. Intuitively, $\bar{P}$ uses no more information than $\hat{P}$. For any $A \in \mathcal{B}$,

$$\mathbb{P}(Y \in A \mid \sigma(\bar{P})) = \mathbb{E}\left[\mathbb{P}(Y \in A \mid \sigma(\hat{P}))\middle|\sigma(\bar{P})\right] = \mathbb{E}\left[\bar{P}(A)\middle|\sigma(\bar{P})\right] = \bar{P}(A) \quad \text{a.s.},$$

which shows that $\bar{P}$ is auto-calibrated. The first equality follows from the repeated conditioning property (Theorem 1.10b), the second from the definition of $\bar{P}$, and the third from the conditional determinism property (Theorem 1.10a) in Cinlar (2009). □

B.2.3 Properties of Auto-Calibration

Lemma 3 (Sharpness identity for auto-calibrated predictive distributions)**.** Let $\bar{P}$ be auto-calibrated w.r.t. Y. Then

$$\mathbb{E}[S(\bar{P}, Y)] = \mathbb{E}[H(\bar{P})].$$

This indicates that the expected score of an auto-calibrated model can be evaluated from the model alone.

Proof. By auto-calibration, we have $\mathbb{P}(Y \in \cdot \mid \sigma(\bar{P})) = \bar{P}$ a.s. Equivalently (Definition 14), for any $\sigma(\bar{P}) \otimes \mathcal{B}$-measurable function g,

$$\mathbb{E}\left[g(\omega, Y)\middle|\sigma(\bar{P})\right](\omega) = \int g(\omega, y)\, \bar{P}(\omega, dy) \quad \text{a.s.}$$

Choosing $g(\omega, y) := S(\bar{P}(\omega), y)$ gives

$$S(\bar{P}, \bar{P}) = \int S(\bar{P}, y)\, \bar{P}(dy) = \mathbb{E}\left[S(\bar{P}, Y)\middle|\sigma(\bar{P})\right] \quad \text{a.s.}$$

Taking expectations yields

$$\mathbb{E}[S(\bar{P}, \bar{P})] = \mathbb{E}[S(\bar{P}, Y)] = \mathbb{E}[H(\bar{P})].$$

□

Lemma 4 (Projection identity and decomposition)**.** Let $\mathcal{G} \subseteq \mathcal{F}$, let P_1 be $\mathcal{G}$-measurable, and let P_3 be any random probability measure. Define $P_2 := \mathbb{E}[P_3 \mid \mathcal{G}]$. Then,

$$\begin{aligned} &\textit{(i)} \quad S(P_1, P_2) = \mathbb{E}[S(P_1, P_3) \mid \mathcal{G}] \quad \text{a.s.}, \\ &\textit{(ii)} \quad S(P_2, P_2) = \mathbb{E}[S(P_2, P_3) \mid \mathcal{G}] \quad \text{a.s.}, \\ &\textit{(iii)} \quad \mathbb{E}[D(P_1, P_3)] = \mathbb{E}[D(P_1, P_2)] + \mathbb{E}[D(P_2, P_3)]. \end{aligned}$$

Proof. Define $h_1(\omega, y) := S(P_1(\omega), y)$ and $h_2(\omega, y) := S(P_2(\omega), y)$. Since P_1, P_2 are $\mathcal{G}$-measurable and S is measurable, h_1, h_2 are $\mathcal{G} \otimes \mathcal{B}$-measurable and integrable. By the defining property of conditional expectation of random measures,

$$S(P_1, P_2) = \int h_1 \, P_2 = \mathbb{E}\left[\int h_1 \, P_3 \middle| \mathcal{G} \right] = \mathbb{E}[S(P_1, P_3) \mid \mathcal{G}],$$

and similarly,

$$S(P_2, P_2) = \int h_2 \, P_2 = \mathbb{E}\left[\int h_2 \, P_3 \middle| \mathcal{G} \right] = \mathbb{E}[S(P_2, P_3) \mid \mathcal{G}].$$

Taking conditional expectations given $\mathcal{G}$ and using that $D(P_1, P_2)$ is $\mathcal{G}$-measurable,

$$\begin{aligned} \mathbb{E}[D(P_1, P_2) + D(P_2, P_3) \mid \mathcal{G}] &= D(P_1, P_2) + \mathbb{E}[D(P_2, P_3) \mid \mathcal{G}] \\ &= S(P_1, P_2) - S(P_2, P_2) + \mathbb{E}[S(P_2, P_3) \mid \mathcal{G}] - \mathbb{E}[S(P_3, P_3) \mid \mathcal{G}] \\ &= \mathbb{E}[S(P_1, P_3) \mid \mathcal{G}] - S(P_2, P_2) + S(P_2, P_2) - \mathbb{E}[S(P_3, P_3) \mid \mathcal{G}] \\ &= \mathbb{E}[S(P_1, P_3) - S(P_3, P_3) \mid \mathcal{G}] \\ &= \mathbb{E}[D(P_1, P_3) \mid \mathcal{G}], \end{aligned}$$

and the claim (iii) follows by taking expectations. □

Theorem 9. Define the initial prediction $\hat{P} = h_\theta(X)$ for some measurable $h_\theta : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$, the auto-calibrated prediction $\bar{P} = \mathbb{P}(Y \in \cdot \mid \sigma(\hat{P}))$ and the true conditional law $P = \mathbb{P}(Y \in \cdot \mid \sigma(X))$. Then

$$\mathbb{E}[D(\hat{P}, P)] = \mathbb{E}[D(\hat{P}, \bar{P})] + \mathbb{E}[D(\bar{P}, P)].$$

Proof. Since $\hat{P} = h_\theta(X)$, we have $\sigma(\hat{P}) \subseteq \sigma(X)$, so by the tower property

$$\mathbb{E}[P(A) \mid \sigma(\hat{P})] = \mathbb{E}[\mathbb{P}(Y \in A \mid \sigma(X)) \mid \sigma(\hat{P})] = \mathbb{P}(Y \in A \mid \sigma(\hat{P})) = \bar{P}(A) \quad \text{a.s.}$$

Apply Lemma 4(iii) with $P_1 = \hat{P}$, $P_2 = \bar{P}$, $P_3 = P$, $\mathcal{G} = \sigma(\hat{P})$. □

Remark. Propriety of S is not needed for the algebraic identity, but if S is (strictly) proper then $D(P, Q) \geqslant 0$ (and equals 0 iff $P = Q$; for random P, Q, equality is a.s.), so each term has the usual nonnegative interpretation.

B.3. Proofs on Calibration for Decision-Making

This section provides proofs for theorems and lemmas in Section 2.6.

B.3.1 Bayes Decision Rule for Threshold Loss Function

Lemma 2. Under **H**1, the Bayes Decision Rule for a threshold loss function is a threshold decision rule. Specifically,

$$\delta^*(x) = \mathbb{1}\left(F_{\hat{G}|X=x}(t) \leqslant \frac{c_{1,0} - c_{1,1}}{c_{1,0} - c_{1,1} + c_{0,1} - c_{0,0}}\right). \tag{2.72}$$

Proof. Define $\hat{p}_x = F_{\hat{G}|X=x}(t) = \mathbb{P}(g(X, \hat{Y}) \leqslant t \mid X = x)$. The expected loss for an arbitrary action $a \in \{0, 1\}$ and a given input x is:

$$\mathbb{E}\Big[l_t(X, \hat{Y}, a) \mid X = x\Big] = \mathbb{E}\Big[c_{\mathbb{1}(g(X,\hat{Y})>t),a} \mid X = x\Big] \tag{B.1}$$

$$= c_{0,a}\hat{p}_x + c_{1,a}(1 - \hat{p}_x). \tag{B.2}$$

The Bayes rule $\delta^*(x)$ selects action $a = 1$ if its expected loss is less than or equal to the loss for $a = 0$. We choose $a = 1$ if:

$$\mathbb{E}\Big[l_t(X, \hat{Y}, 1 \mid X = x)\Big] \leqslant \mathbb{E}\Big[l_t(X, \hat{Y}, 0 \mid X = x)\Big] \tag{B.3}$$

$$c_{0,1}\hat{p}_x + c_{1,1}(1 - \hat{p}_x) \leqslant c_{0,0}\hat{p}_x + c_{1,0}(1 - \hat{p}_x) \tag{B.4}$$

$$c_{1,1} + \hat{p}_x(c_{0,1} - c_{1,1}) \leqslant c_{1,0} + \hat{p}_x(c_{0,0} - c_{1,0}) \tag{B.5}$$

$$\hat{p}_x(c_{0,1} - c_{1,1} + c_{1,0} - c_{0,0}) \leqslant c_{1,0} - c_{1,1}. \tag{B.6}$$

By **H**1, the term $(c_{0,1} - c_{1,1} + c_{1,0} - c_{0,0})$ is positive. Dividing yields the threshold:

$$\hat{p}_x \leqslant \frac{c_{1,0} - c_{1,1}}{c_{1,0} - c_{1,1} + c_{0,1} - c_{0,0}}. \tag{B.7}$$

Setting the decision threshold α to the right-hand side, we get $\delta^*(x) = \mathbb{1}(\hat{p}_x \leqslant \alpha)$, which concludes the proof. □

B.3.2 Equivalence to Zero Reliability Gap

Theorem 5. Let $\mathcal{L}$ be the space of all threshold loss functions and Δ be the space of all threshold decision rules. Assume that, for any $x \in \mathcal{X}$, $F_{\hat{G}|X=x}$ is strictly increasing. A predictive distribution $\hat{F}_{Y|X}$ satisfies threshold calibration if and only if

$$\gamma(\delta, l) = 0 \quad \forall \delta \in \Delta, l \in \mathcal{L}. \tag{2.74}$$

First, we establish two helper lemmas. The first shows that if a model is threshold-calibrated on a set, it is also calibrated on its complement. Let $A_{t,\alpha}$ be the event $\{x \in \mathcal{X} \mid F_{\hat{G}|X=x}(t) \leqslant \alpha\}$.

Lemma 5. If a model satisfies threshold calibration, i.e., $\forall t \in \mathbb{R}, \alpha \in [0, 1], c \in [0, 1], \mathbb{P}(\hat{U} \leqslant c \mid A_{t,\alpha}) = c$, then it also holds that $\forall t \in \mathbb{R}, \alpha \in [0, 1], c \in [0, 1], \mathbb{P}(\hat{U} \leqslant c \mid A_{t,\alpha}^c) = c$.

Proof. By the definition of pre-rank calibration (the specific case of threshold calibration with $\alpha = 1$), we have $\mathbb{P}(\hat{U} \leqslant c) = c$. By the law of total probability:

$$\mathbb{P}(\hat{U} \leqslant c) = \mathbb{P}(\hat{U} \leqslant c \mid A_{t,\alpha})\mathbb{P}(A_{t,\alpha}) + \mathbb{P}(\hat{U} \leqslant c \mid A_{t,\alpha}^c)\mathbb{P}(A_{t,\alpha}^c) \tag{B.8}$$

$$c = c \cdot \mathbb{P}(A_{t,\alpha}) + \mathbb{P}(\hat{U} \leqslant c \mid A_{t,\alpha}^c)\mathbb{P}(A_{t,\alpha}^c) \tag{B.9}$$

$$c(1 - \mathbb{P}(A_{t,\alpha})) = \mathbb{P}(\hat{U} \leqslant c \mid A_{t,\alpha}^c)\mathbb{P}(A_{t,\alpha}^c) \tag{B.10}$$

$$c \cdot \mathbb{P}(A_{t,\alpha}^c) = \mathbb{P}(\hat{U} \leqslant c \mid A_{t,\alpha}^c)\mathbb{P}(A_{t,\alpha}^c) \tag{B.11}$$

$$c = \mathbb{P}(\hat{U} \leqslant c \mid A_{t,\alpha}^c) \tag{B.12}$$

□

The second lemma establishes that threshold calibration, which is a property on sets, implies a stronger pointwise calibration property.

Lemma 6. Let $\hat{p}_X = F_{\hat{G}|X}(t)$. If a model satisfies threshold calibration for a fixed $t \in \mathbb{R}$, i.e.,

$$\mathbb{P}(\hat{U} \leqslant c \mid \hat{p}_X \leqslant \alpha) = c \quad \forall \alpha \in [0,1], c \in [0,1], \tag{B.13}$$

then it follows that $\mathbb{P}(\hat{U} \leqslant z \mid \hat{p}_X = z) = z$ for almost all z in the support of $\hat{p}_X$.

Proof. Let $c \in [0,1]$ be a fixed constant. The assumption of threshold calibration is $\mathbb{E}\Big[\mathbb{I}(\hat{U} \leqslant c) \mid \hat{p}_X \leqslant \alpha\Big] = c$. By the law of total expectation (tower property), we can introduce conditioning on the random variable $\hat{p}_X$:

$$\mathbb{E}\Big[\mathbb{E}\Big[\mathbb{I}(\hat{U} \leqslant c) \mid \hat{p}_X\Big] \mid \hat{p}_X \leqslant \alpha\Big] = c. \tag{B.14}$$

Let us define the function $h_c(z) := \mathbb{E}\Big[\mathbb{I}(\hat{U} \leqslant c) \mid \hat{p}_X = z\Big] = \mathbb{P}(\hat{U} \leqslant c \mid \hat{p}_X = z)$. The equation becomes:

$$\mathbb{E}[h_c(\hat{p}_X) \mid \hat{p}_X \leqslant \alpha] = c. \tag{B.15}$$

This states that the expected value of the function h_c over the event set $\{\hat{p}_X \leqslant \alpha\}$ is equal to the constant c. Since this holds for all $\alpha \in [0,1]$, the function itself must be equal to the constant almost everywhere. Thus, for a fixed c:

$$h_c(z) = \mathbb{P}(\hat{U} \leqslant c \mid \hat{p}_X = z) = c \quad \text{for almost all } z. \tag{B.16}$$

Since this identity holds for any arbitrary constant $c \in [0,1]$, it must also hold when we substitute the specific value z for c, leading to the desired result: $\mathbb{P}(\hat{U} \leqslant z \mid \hat{p}_X = z) = z$. □

We now proceed to the main theorem.

Proof. Let $\hat{p}_X = F_{\hat{G}|X}(t)$ and $p_X = F_{G|X}(t) = \mathbb{P}(G \leqslant t \mid X)$. The decision rule is $\delta_\alpha(X) = \mathbb{1}(\hat{p}_X \leqslant \alpha)$.

($\Rightarrow$) Assume the model satisfies threshold calibration. The reliability gap is $\gamma(\delta_\alpha, l_t) = \left|\mathbb{E}\Big[l_t(X, \hat{Y}, \delta_\alpha(X))\Big] - \mathbb{E}[l_t($
We show the difference is zero.

$$\mathbb{E}[l_t(X, Y, \delta_\alpha(X))] - \mathbb{E}\Big[l_t(X, \hat{Y}, \delta_\alpha(X))\Big] \tag{B.17}$$
$$= \mathbb{E}\Big[\mathbb{E}\Big[l_t(X, Y, \delta_\alpha(X)) - l_t(X, \hat{Y}, \delta_\alpha(X)) \mid X\Big]\Big] \tag{B.18}$$
$$= \mathbb{E}\big[\big(c_{0,\delta_\alpha(X)} p_X + c_{1,\delta_\alpha(X)}(1 - p_X)\big) - \big(c_{0,\delta_\alpha(X)} \hat{p}_X + c_{1,\delta_\alpha(X)}(1 - \hat{p}_X)\big)\big] \tag{B.19}$$
$$= \mathbb{E}\big[(c_{0,\delta_\alpha(X)} - c_{1,\delta_\alpha(X)})(p_X - \hat{p}_X)\big] \tag{B.20}$$
$$= (c_{0,1} - c_{1,1})\mathbb{E}[(p_X - \hat{p}_X)\mathbb{1}(\hat{p}_X \leqslant \alpha)] + (c_{0,0} - c_{1,0})\mathbb{E}[(p_X - \hat{p}_X)\mathbb{1}(\hat{p}_X > \alpha)] \tag{B.21}$$

This difference is zero if $\mathbb{E}[(p_X - \hat{p}_X) \mid \hat{p}_X \leqslant \alpha] = 0$ and $\mathbb{E}[(p_X - \hat{p}_X) \mid \hat{p}_X > \alpha] = 0$. We show the first; the second follows from Lemma 5. We must show $\mathbb{E}[p_X | \hat{p}_X \leqslant \alpha] = \mathbb{E}[\hat{p}_X \mid \hat{p}_X \leqslant \alpha]$.

$$\mathbb{E}[p_X \mid \hat{p}_X \leqslant \alpha] = \mathbb{E}[\mathbb{P}(G \leqslant t|X) \mid \hat{p}_X \leqslant \alpha] \tag{B.22}$$
$$= \mathbb{E}[\mathbb{1}(G \leqslant t) \mid \hat{p}_X \leqslant \alpha] \tag{B.23}$$
$$= \mathbb{E}\Big[\mathbb{1}(F_{\hat{G}|X}(G) \leqslant F_{\hat{G}|X}(t)) \mid \hat{p}_X \leqslant \alpha\Big] \tag{B.24}$$
$$= \mathbb{E}\Big[\mathbb{1}(\hat{U} \leqslant \hat{p}_X) \mid \hat{p}_X \leqslant \alpha\Big] \tag{B.25}$$
$$= \mathbb{E}\Big[\mathbb{P}(\hat{U} \leqslant \hat{p}_X \mid \hat{p}_X) \mid \hat{p}_X \leqslant \alpha\Big] \tag{B.26}$$
$$= \mathbb{E}[\hat{p}_X \mid \hat{p}_X \leqslant \alpha] \tag{B.27}$$

Line (B.23) follows from the tower property of conditional expectation, since $\hat{p}_X \leqslant \alpha$ is a function of X. Line (B.24) assumes $F_{\hat{G}|X}$ is strictly increasing. Line (B.26) also follows from the tower property of conditional expectation. Line (B.27) follows from Lemma 6. Thus the reliability gap is zero.

($\Leftarrow$) Assume $\gamma(\delta_\alpha, l_t) = 0$ for all $\delta_\alpha \in \Delta$ and $l_t \in \mathcal{L}$. From the forward proof, this means the following identity holds for all t, α and any cost matrix $\{c_{i,j}\}$ satisfying **H**1:

$$(c_{0,1} - c_{1,1}) \underbrace{\mathbb{E}[(p_X - \hat{p}_X)\mathbb{1}(\hat{p}_X \leqslant \alpha)]}_{E_1} + (c_{0,0} - c_{1,0}) \underbrace{\mathbb{E}[(p_X - \hat{p}_X)\mathbb{1}(\hat{p}_X > \alpha)]}_{E_2} = 0. \tag{B.28}$$

The terms E_1 and E_2 are fixed for a given t and α, but the cost coefficients can be varied. Let $A = c_{0,1} - c_{1,1}$ and $B = c_{0,0} - c_{1,0}$. The equation is $AE_1 + BE_2 = 0$. By choosing two different valid cost matrices, we can form a system of two linear equations for E_1 and E_2.

1. Choose $c_{0,1} = 1, c_{1,1} = 0, c_{0,0} = 0, c_{1,0} = 2$. This satisfies **H**1. Here, $A = 1, B = -2$. The equation is $E_1 - 2E_2 = 0$.
2. Choose $c_{0,1} = 2, c_{1,1} = 0, c_{0,0} = 0, c_{1,0} = 2$. This also satisfies the assumption. Here, $A = 2, B = -2$. The equation is $2E_1 - 2E_2 = 0$, which implies $E_1 = E_2$.

Substituting $E_1 = E_2$ into the first equation gives $E_1 - 2E_1 = 0$, which implies $E_1 = 0$. It follows that $E_2 = 0$. Therefore, for all $t \in \mathbb{R}$ and $\alpha \in [0, 1]$:

$$\mathbb{E}[(p_X - \hat{p}_X)\mathbb{1}(\hat{p}_X \leqslant \alpha)] = 0 \tag{B.29}$$
$$\mathbb{E}[p_X - \hat{p}_X \mid \hat{p}_X \leqslant \alpha]\,\mathbb{P}(\hat{p}_X \leqslant \alpha) = 0 \tag{B.30}$$
$$\mathbb{E}[p_X - \hat{p}_X \mid \hat{p}_X \leqslant \alpha] = 0. \tag{B.31}$$

This is equivalent to the condition $\mathbb{E}[p_X \mid \hat{p}_X \leqslant \alpha] = \mathbb{E}[\hat{p}_X \mid \hat{p}_X \leqslant \alpha]$. From the forward proof, $\mathbb{E}[\hat{p}_X \mid \hat{p}_X \leqslant \alpha] = \mathbb{E}\Big[\mathbb{P}(\hat{U} \leqslant \hat{p}_X \mid \hat{p}_X) \mid \hat{p}_X \leqslant \alpha\Big]$. This implies that the functions inside the outer expectation are equal almost everywhere. Thus, $\mathbb{P}(\hat{U} \leqslant z \mid \hat{p}_X = z) = z$ for almost all z in the support of $\hat{p}_X$. As argued in Lemma 6, this property implies that for a fixed $c \in [0,1]$, $\mathbb{P}(\hat{U} \leqslant c \mid \hat{p}_X = z) = c$ for almost all z.

Using this fact, we can prove threshold calibration directly:

$$\mathbb{P}(\hat{U} \leqslant c \mid \hat{p}_X \leqslant \alpha) = \mathbb{E}\Big[\mathbb{P}(\hat{U} \leqslant c \mid \hat{p}_X) \mid \hat{p}_X \leqslant \alpha\Big] \tag{B.32}$$

$$= \mathbb{E}[c \mid \hat{p}_X \leqslant \alpha] = c. \tag{B.33}$$

This holds for all t, α, c, which is the definition of threshold calibration. □

APPENDIX C

Supplementary Material for Chapter 3

C.1. Proofs

C.1.1 Integral of the Absolute Difference between CDFs or QFs

Proposition 6. Let $F_A, F_B : [0,1] \to [0,1]$ denote two strictly increasing CDFs of random variables defined on $[0,1]$ with corresponding QFs Q_A and Q_B. Then,

$$\int_0^1 |F_A(q) - F_B(q)| dq = \int_0^1 |Q_A(p) - Q_B(p)| dp. \tag{C.1}$$

Proof. We define two functions $r, s : [0,1] \times [0,1] \to \{\, 0, 1 \,\}$ where

$$r(q,p) = \begin{cases} 1 & \text{if } F_A(q) \leqslant p \leqslant F_B(q) \text{ or } F_B(q) \leqslant p \leqslant F_A(q) \\ 0 & \text{otherwise,} \end{cases} \tag{C.2}$$

$$\text{and } s(q,p) = \begin{cases} 1 & \text{if } Q_A(p) \leqslant q \leqslant Q_B(p) \text{ or } Q_B(p) \leqslant q \leqslant Q_A(p) \\ 0 & \text{otherwise.} \end{cases} \tag{C.3}$$

Let us show that r and s are equal. Considering $q \in [0,1]$ and $p \in [0,1]$, we can write

$$\begin{aligned} & F_A(q) \leqslant p \leqslant F_B(q) && \text{(C.4)} \\ \iff & (F_A(q) \leqslant p) \wedge (p \leqslant F_B(q)) && \text{(C.5)} \\ \iff & (q \leqslant Q_A(p)) \wedge (Q_B(p) \leqslant q) && \text{(C.6)} \\ \iff & Q_B(p) \leqslant q \leqslant Q_A(p), && \text{(C.7)} \end{aligned}$$

where (C.6) holds since both F_A and F_B are strictly increasing.

Similarly, $F_B(q) \leqslant p \leqslant F_A(q) \iff Q_A(p) \leqslant q \leqslant Q_B(p)$. Hence $r(q,p) = 1 \iff s(q,p) = 1$ and r and s are equal.

By Fubini's theorem, we have

$$\int_0^1 \int_0^1 r(q,p)dpdq = \int_0^1 \int_0^1 s(q,p)dqdp. \tag{C.8}$$

Furthermore, upon evaluating the inner integrals, we obtain

$$\int_0^1 r(q,p)dp = \begin{cases} \int_{F_A(q)}^{F_B(q)} 1dp & \text{if } F_A(q) \leqslant F_B(q) \\ \int_{F_B(q)}^{F_A(q)} 1dp & \text{otherwise} \end{cases} \tag{C.9}$$

$$= |F_A(q) - F_B(q)|. \tag{C.10}$$

Similarly, we have $\int_0^1 s(q,p)dq = |Q_A(p) - Q_B(p)|$. Finally, by substituting these results in (C.8), we prove (C.1).

□

C.2. Additional Results

This section presents additional experimental results.

C.2.1 Comparison between Recalibration, Conformal Prediction and Regularization Approaches per Base Predictor

First, we present the results of our experiments comparing recalibration, conformal prediction, and regularization approaches. Our objective is to determine which metrics are improved by these methods compared to a vanilla model. We divide our comparisons based on the three base predictors considered: `MIX-NLL` (Figure C.1), `MIX-CRPS` (Figure C.2) and `SQR-CRPS` (Figure C.3).

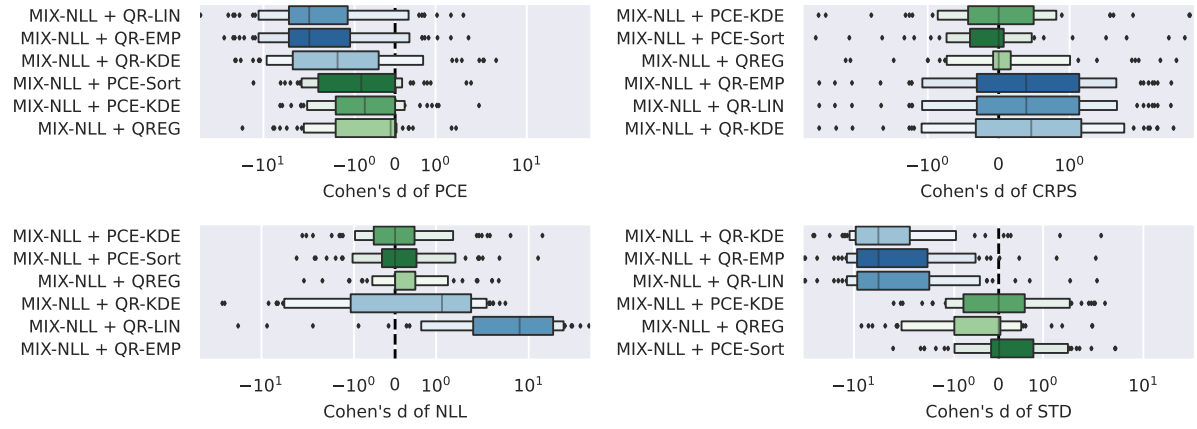
Since NLL, CRPS, and standard deviation cannot be directly compared across different datasets, we utilize Cohen's d as an effect size measure, with the baseline being a vanilla model of the same base predictor. For instance, the baseline for `MIX-CRPS + QR-EMP` is `MIX-CRPS`. Additionally, we provide critical difference diagrams to assess the significance of differences.

Overall, recalibration and conformal prediction demonstrate significantly improved PCE compared to the baseline, although there is a trade-off with other metrics. For both `MIX-NLL` and `MIX-CRPS`, `QR-EMP` yields infinite NLL, `QR-LIN` substantially increases NLL, while `QR-KDE` has a lesser impact on NLL. However, `QR-KDE` results in a significant degradation of CRPS compared to other recalibration methods when the base predictor is MIX-CRPS. In the case of quantile predictions, `CQR` significantly improves PCE.

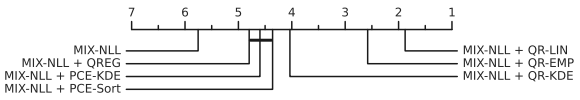
While regularization methods generally lead to improved PCE, they are still outperformed by recalibration and conformal prediction in this regard. However, we observe that with the `MIX-NLL` base predictor, regularization methods (`PCE-KDE`, `PCE-Sort` and `QREG`) have minimal impact on CRPS, NLL, and STD compared to recalibration methods. With the `MIX-CRPS` base predictor, the difference in CRPS between recalibration and regularization is less pronounced. Nevertheless,

it is evident that regularization methods `PCE-KDE` and `PCE-Sort`, which rely on PCE, result in less sharp predictions compared to recalibration methods, which produce sharper predictions.

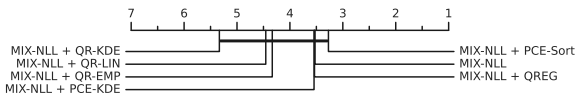
Regarding quantile predictions, the case is reversed: conformal prediction (`SQR-CRPS + CQR`) yields less sharp predictions, while regularization with `SQR-CRPS + Trunc` leads to sharper predictions.



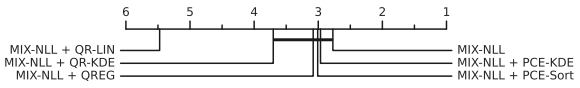
(a) Boxplots of Cohen's d of different metrics on all datasets, w.r.t. `MIX-NLL`.



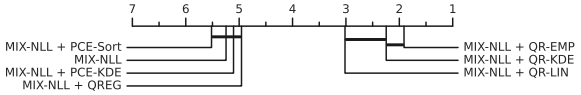
(b) PCE



(c) CRPS

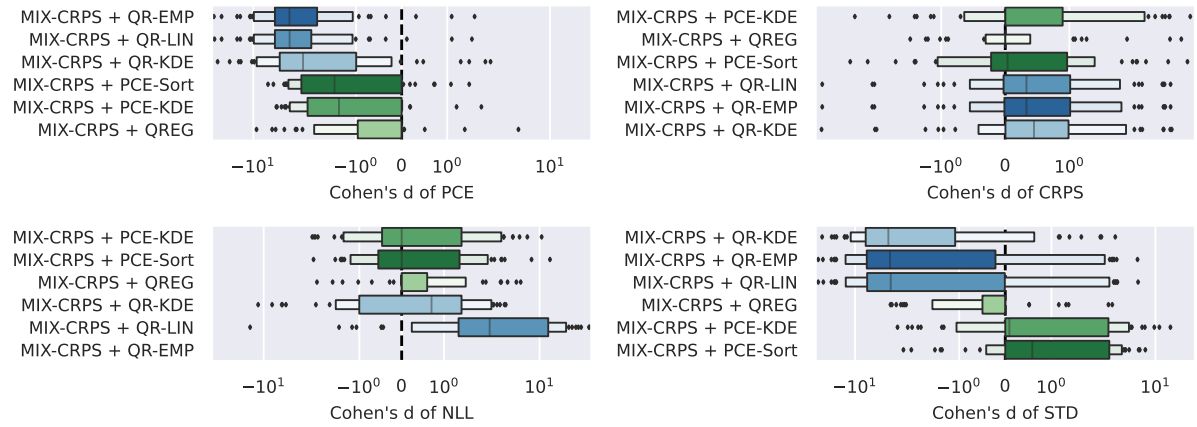


(d) NLL

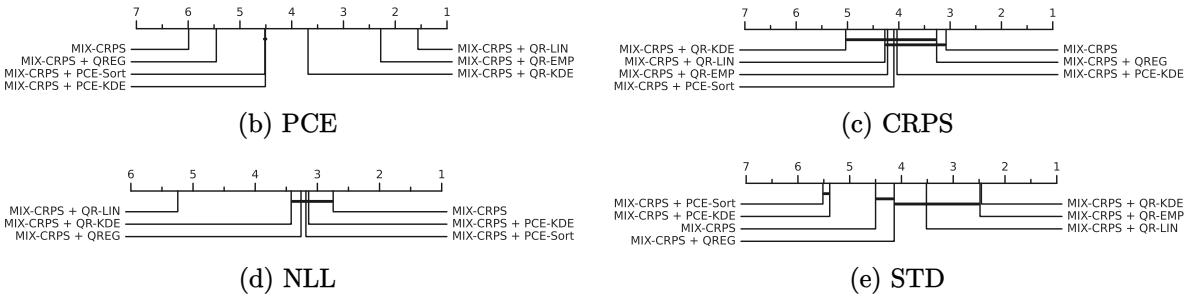


(e) STD

Figure C.1: Comparison of different metrics where the base predictor is `MIX-NLL`.



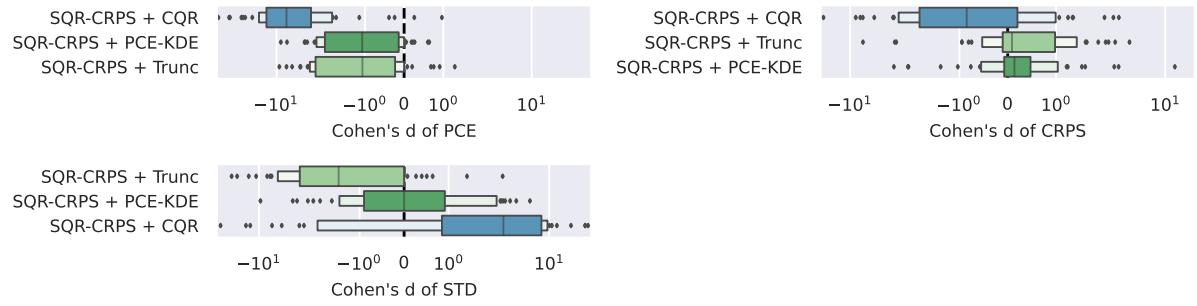
(a) Boxplots of Cohen's d of different metrics on all datasets, w.r.t. **MIX-CRPS**.



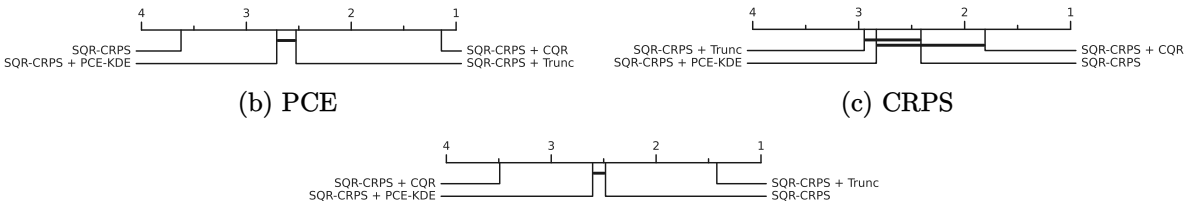
(b) PCE (c) CRPS

(d) NLL (e) STD

Figure C.2: Comparison of different metrics where the base predictor is **MIX-CRPS**.



(a) Boxplots of Cohen's d of different metrics on all datasets, w.r.t. **SQR-CRPS**.



(b) PCE (c) CRPS

(d) STD

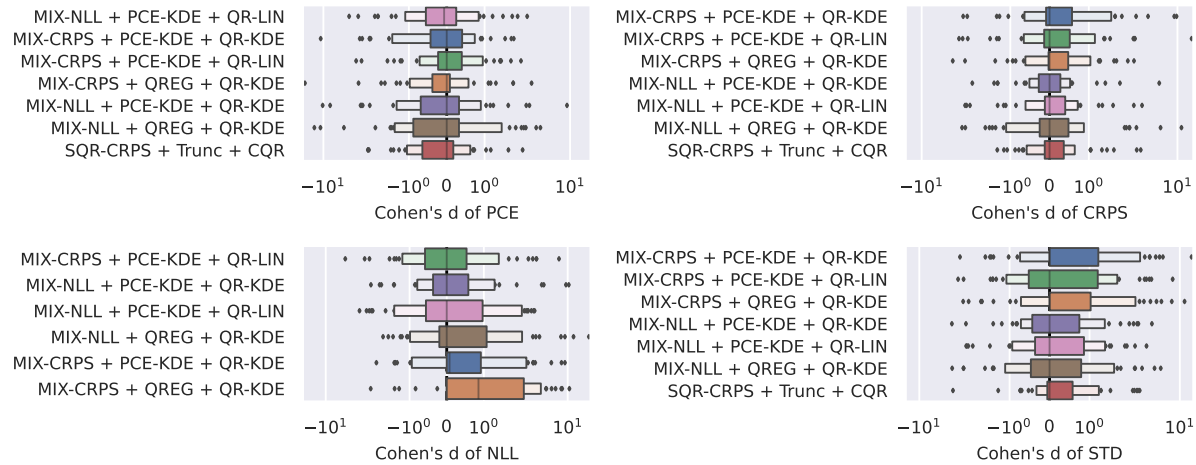
Figure C.3: Comparison of different metrics where the base predictor is **SQR-CRPS**.

C.2.2 Combining Regularization and Post-hoc Methods

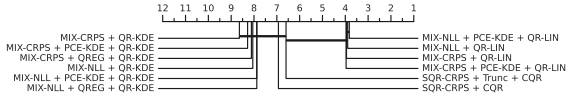
In this paper, we have established that post-hoc methods are generally more favorable than regularization methods when the primary objective is to enhance probabilistic calibration. Since regularization methods operate during training and do not alter the form of predictions (e.g., Gaussian mixture predictions), they can be easily combined with post-hoc methods. In this section, we address the question: "Which metrics do regularization methods improve when combined with a post-hoc method compared to the same model without regularization?"

To ensure clarity, we focus our presentation on a selection of paired regularization and post-hoc methods. Figure C.4 illustrates the impact of regularization on various metrics for these pairs. In Figure C.4a, the baseline corresponds to the same post-hoc method without regularization, enabling a direct measurement of the effect of adding regularization to a post-hoc method. It is important to note that the boxplots in this figure cannot be directly compared due to the different baselines.

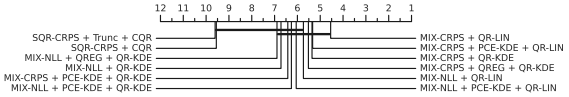
The critical difference diagrams provide a comparison of all methods, with and without regularization. Overall, when combined with post-hoc methods, regularization has a negative impact: no regularization method significantly improves probabilistic calibration, and they tend to negatively affect CRPS, NLL, and STD metrics.



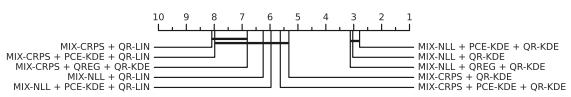
(a) Boxplots of Cohen's d of different metrics on all datasets, w.r.t. the same model except that regularization is not applied.



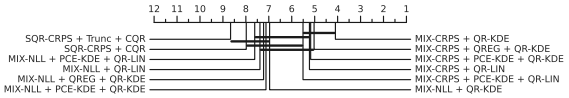
(b) PCE



(c) CRPS



(d) NLL



(e) STD

Figure C.4: Comparison of different metrics showing the effect of regularization when combined with a post-hoc method, compared to the same model without regularization.

C.2.3 Post-hoc Calibration based on the Training Dataset

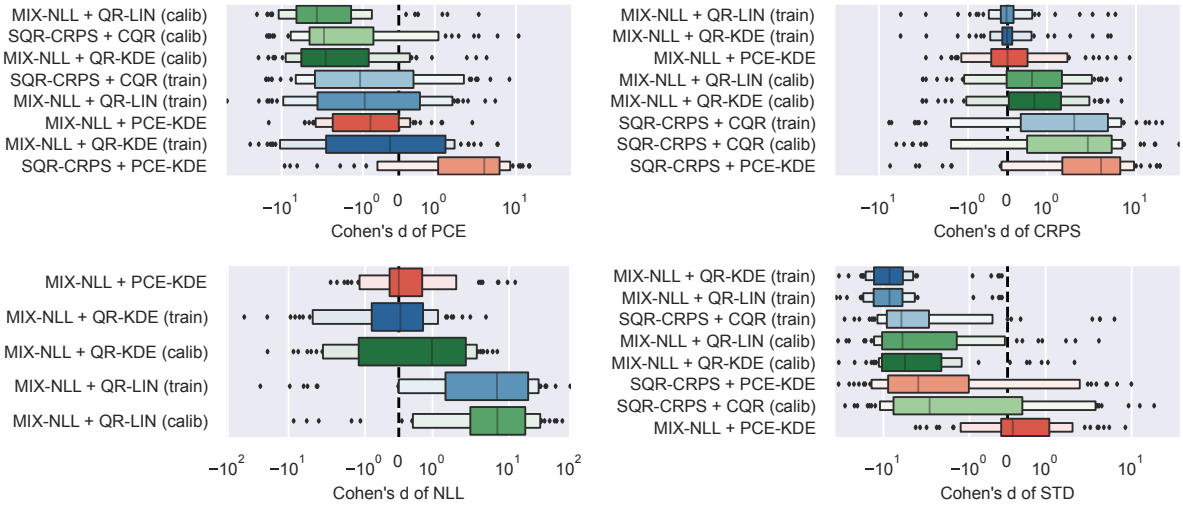
In this paper, the calibration map or conformity scores have been computed on a separate calibration dataset, following common practice in the literature. However, holding out data for post-hoc calibration reduces the quantity of training data. For the sake of clarity, we focus our analysis on the `MIX-NLL` and `SQR-CRPS` base losses.

In this section, we compare post-hoc calibration based on the training dataset to post-hoc calibration based on the calibration dataset. We aim to answer the question: "Can it be beneficial to use post-hoc calibration based on the training dataset, and should it be preferred over regularization methods when there is no calibration dataset available?" One advantage of regularization methods and post-hoc calibration methods based on the training dataset is that the base predictor can be trained on more data (80% in our experiments, compared to 65% when holding out the calibration dataset).

Figure C.5 presents a comparison of different methods, with post-hoc methods trained on the calibration dataset indicated by `(calib)` and those trained on the training dataset indicated by `(train)`. We observe that post-hoc methods based on the calibration dataset tend to significantly outperform their counterparts based on the training dataset in terms of probabilistic calibration.

Specifically, `MIX-NLL + QR-LIN` and `MIX-NLL + QR-KDE` achieve significantly better calibration when the calibration map is learned on the calibration dataset. Similarly, `SQR-CRPS + CQR` tends to improve calibration when conformal prediction is based on the calibration dataset. It is worth noting that even without a calibration dataset, post-hoc methods tend to be better calibrated than regularization methods.

Finally, we observe that post-hoc methods based on the training dataset tend to achieve better CRPS and NLL scores, although not significantly. Additionally, they are also significantly sharper. This may be attributed to the larger training dataset available to the base predictor when there is no held-out dataset.



(a) Boxplots of Cohen's d of different metrics on all datasets, w.r.t. `MIX-NLL`.

(b) PCE (c) CRPS

(d) NLL (e) STD

Figure C.5: Comparison of different metrics.

C.2.4 Calibration of Vanilla Models

Figure C.7 and Figure C.8 provide additional results from our empirical study in Section 3.4, specifically focusing on the PCE obtained with `MIX-CRPS` and `SQR-CRPS`. The datasets are ordered in the same manner as shown in Figure 3.2 for comparison. We observe that `SQR-CRPS` is less calibrated compared to `MIX-NLL`.

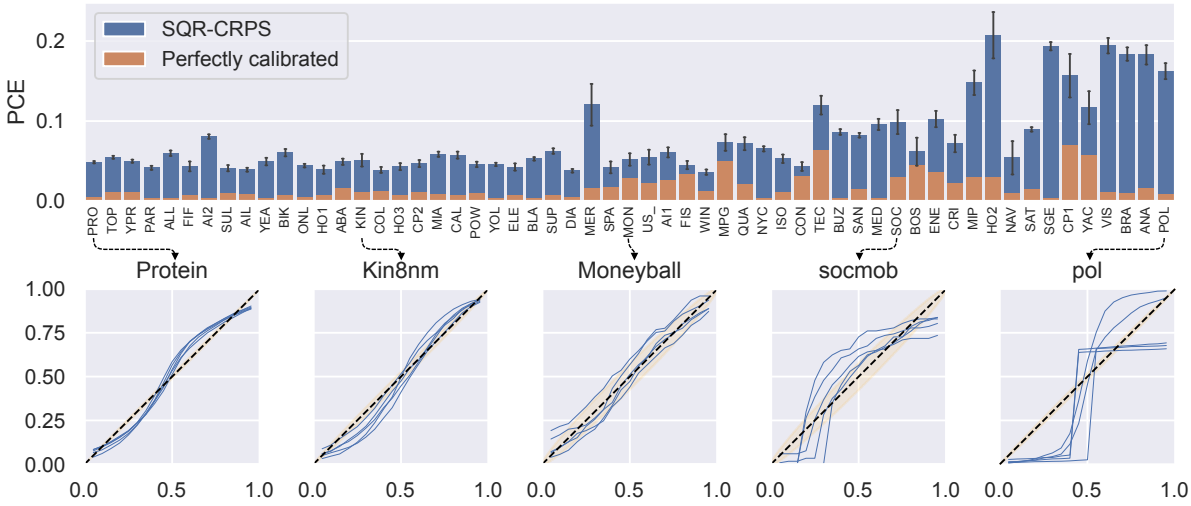


Figure C.6: PCE obtained on different datasets, with examples of reliability diagrams. The height of each bar is the mean PCE of 5 runs with different dataset splits while the error bar represents the standard error of the mean. For 5 datasets, the PIT reliability diagrams of 5 runs are displayed in the bottom row.

Figure C.7: PCE of `SQR-CRPS`, on all datasets.

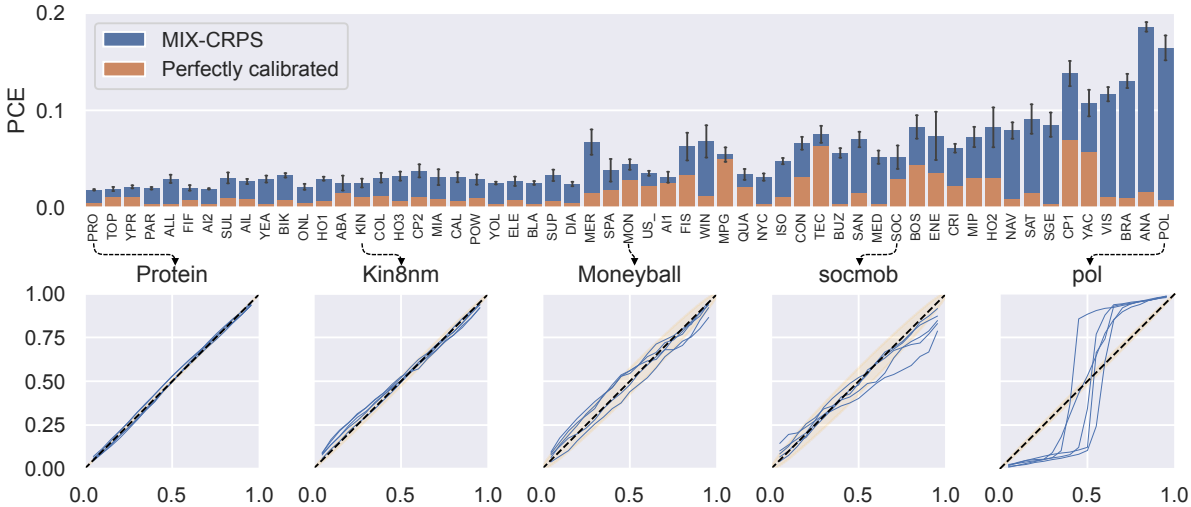


Figure C.8: PCE of `MIX-CRPS`, on all datasets.

C.2.5 Distribution of the Test Statistic

Figure C.9 shows the distribution of the test statistic, as described in Section 3.4. We observe that, in a lot of cases, the average PCE of the compared models is larger than all the 10^4 samples of the average PCE from a probabilistically calibrated model. Among the different calibration methods, post-hoc calibration with `MIX-NLL + QR-EMP` achieves the highest level of calibration performance in the majority of cases.

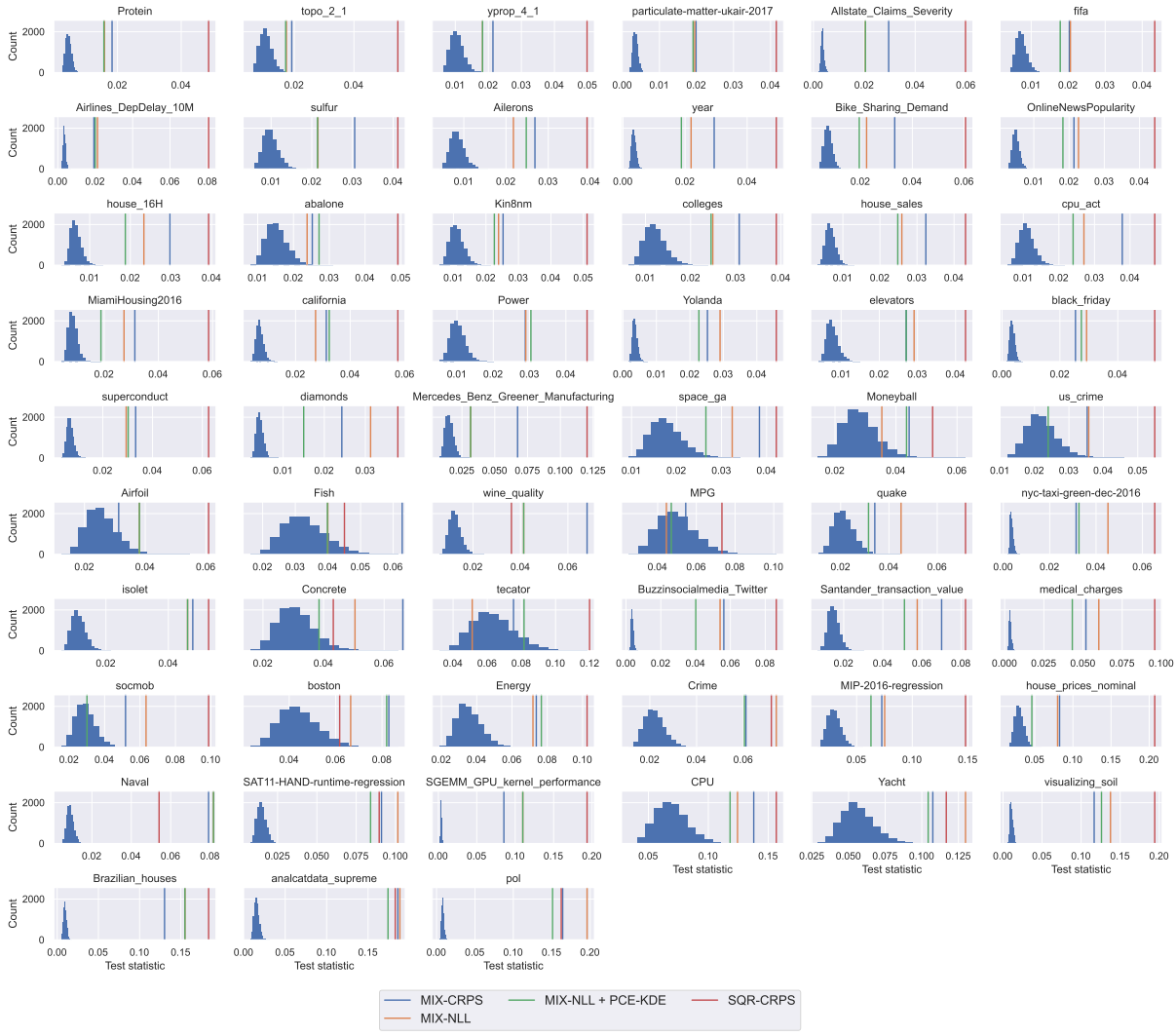


Figure C.9: Distribution of the test statistic on all datasets for different models.

C.2.6 Reliability Diagrams

Figure C.10 and Figure C.11 compare reliability diagrams obtained on models with and without post-hoc calibration, respectively. With only a few exceptions, the post-hoc calibrated models exhibit a visual proximity to the diagonal line.

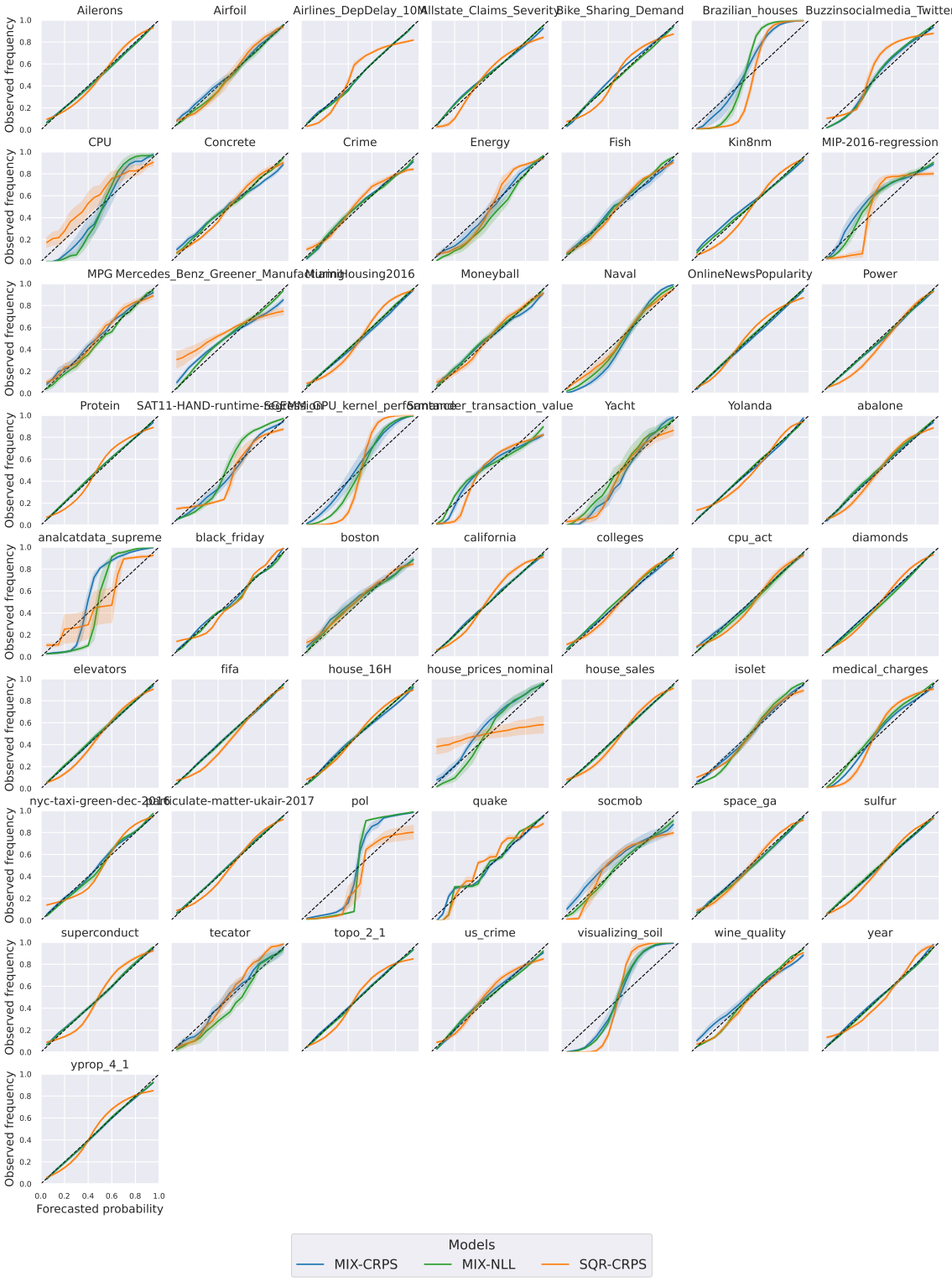


Figure C.10: Reliability diagrams on all datasets for different models.

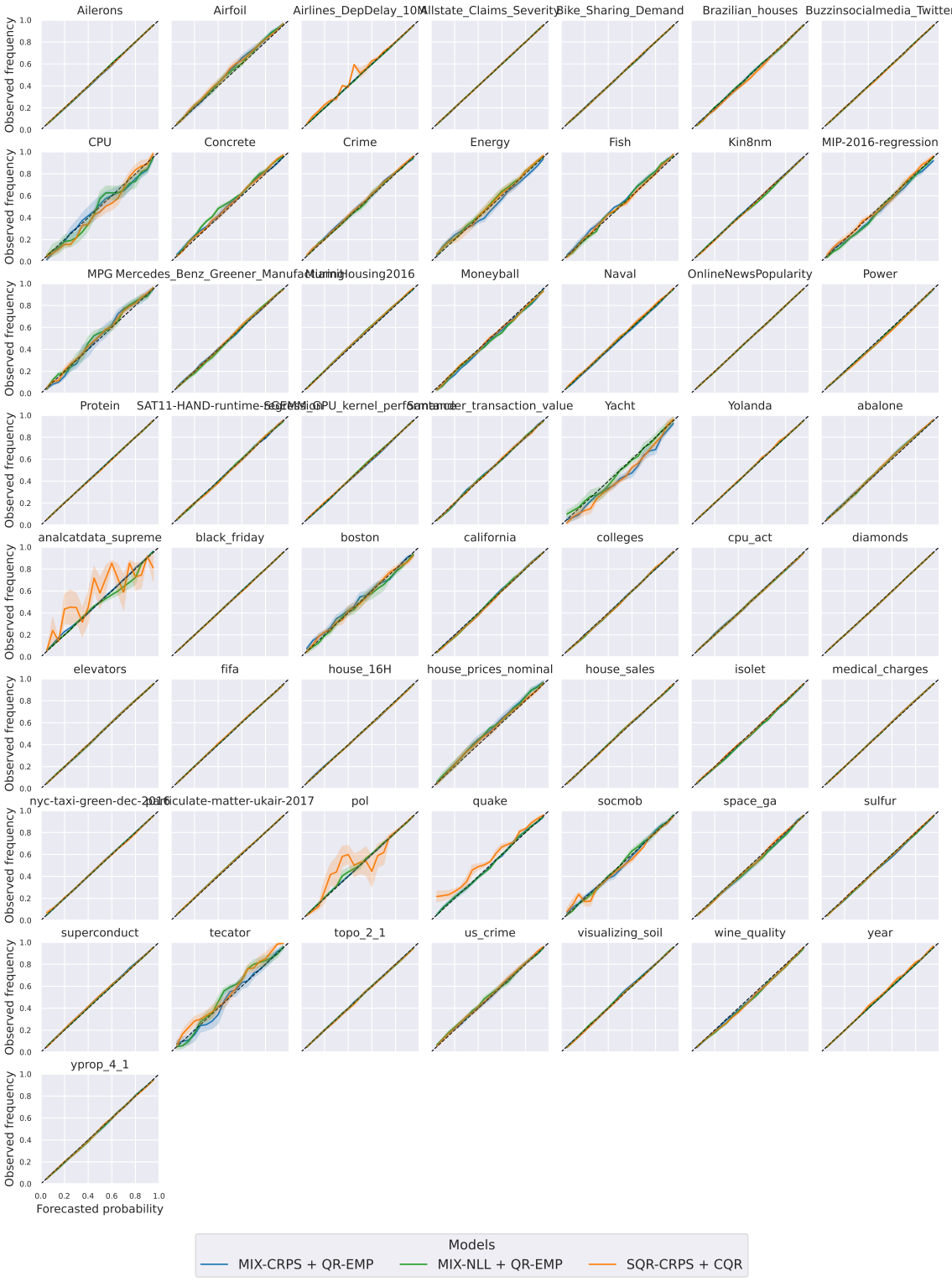


Figure C.11: Reliability diagrams on all datasets for different models with post-hoc calibration.

C.3. Hyperparameters

In our experiments, we adopt a specific architecture consisting of 3 hidden layers with 100 units per layer, ReLU non-linearities, and a dropout rate of 0.2 on the last hidden layer. Early stopping with a patience of 30 is applied to select the epoch with the lowest base loss on the validation dataset.

In this section, we delve into the performance of different model parameters, including the number of components in Gaussian mixture predictions, the number of quantiles in quantile predictions, and the number of hidden layers in the underlying models.

Figure C.12 compares models that predict mixtures with varying numbers of components compared to the reference of 3 components. Notably, when there is only 1 component (yielding a single Gaussian prediction), the model's performance significantly deteriorates in terms of CRPS, NLL, and sharpness. However, as the number of components increases beyond 3, the differences become less pronounced.

Figure C.13 compares models with different numbers of quantiles compared to a reference of 64 quantiles. The results reveal a consistent pattern: predicting more quantiles consistently enhances performance in terms of probabilistic calibration, CRPS, and sharpness.

Figure C.14 compares models with different numbers of layers relative to a 3-layer model. It highlights that models with 2, 3, or 5 layers tend to yield superior performance in terms of CRPS and NLL.

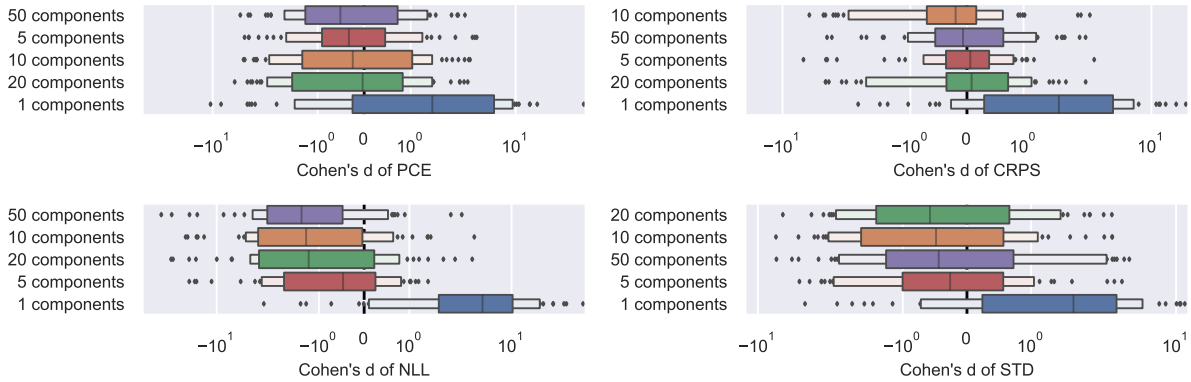


Figure C.12: Comparison of models whose predictions are Gaussian mixtures with different numbers of components. All models are trained with NLL loss, without regularization or post-hoc method. The box plots show Cohen's d of different metrics on all datasets. Cohen's d is computed w.r.t. a model whose predictions are Gaussian mixtures with 3 components.

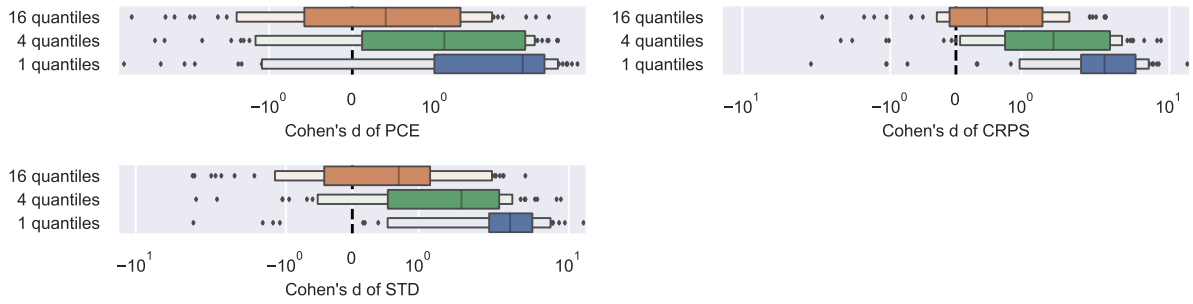


Figure C.13: Comparison of models whose predictions are different numbers of quantiles. All models are trained with CRPS loss, without regularization or post-hoc method. The box plots show Cohen's d of different metrics on all datasets. Cohen's d is computed w.r.t. a model whose predictions are 64 quantiles.

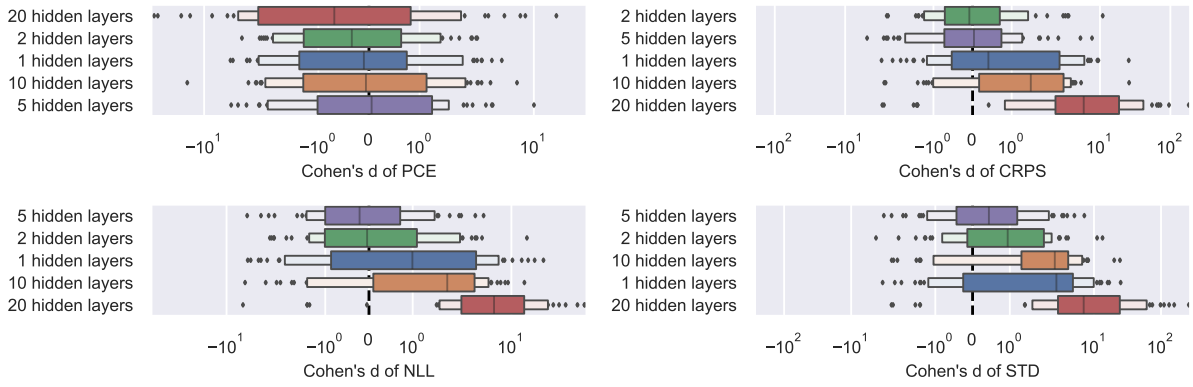


Figure C.14: Comparison of models with different number of layers. All models predict Gaussian mixtures and are trained with NLL loss, without regularization or post-hoc method. The box plots show Cohen's d of different metrics on all datasets. Cohen's d is computed w.r.t. a model with 3 hidden layers.

C.4. Tabular Regression Datasets

Table A.1 presents the datasets considered in our experiments. To ensure consistency, when datasets are available from multiple sources, we select one specific source per dataset, as indicated in Figure 3.1. Our selection prioritizes the suites 297, 299, and 269 of OpenML, followed by UCI datasets.

In the OpenML suite 297, we discovered that the datasets `houses` and `california` are identical, and thus, we only included the `california` dataset in our analysis. Moreover, the UCI archive for the dataset `wine_quality` contains two separate datasets for red and white wine. As there was no indication regarding the specific dataset(s) used in previous studies, we solely considered the dataset related to white wine as in Grinsztajn et al. (2022). In Figure 3.1, other studies may have employed the alternative dataset or a combination of both datasets.

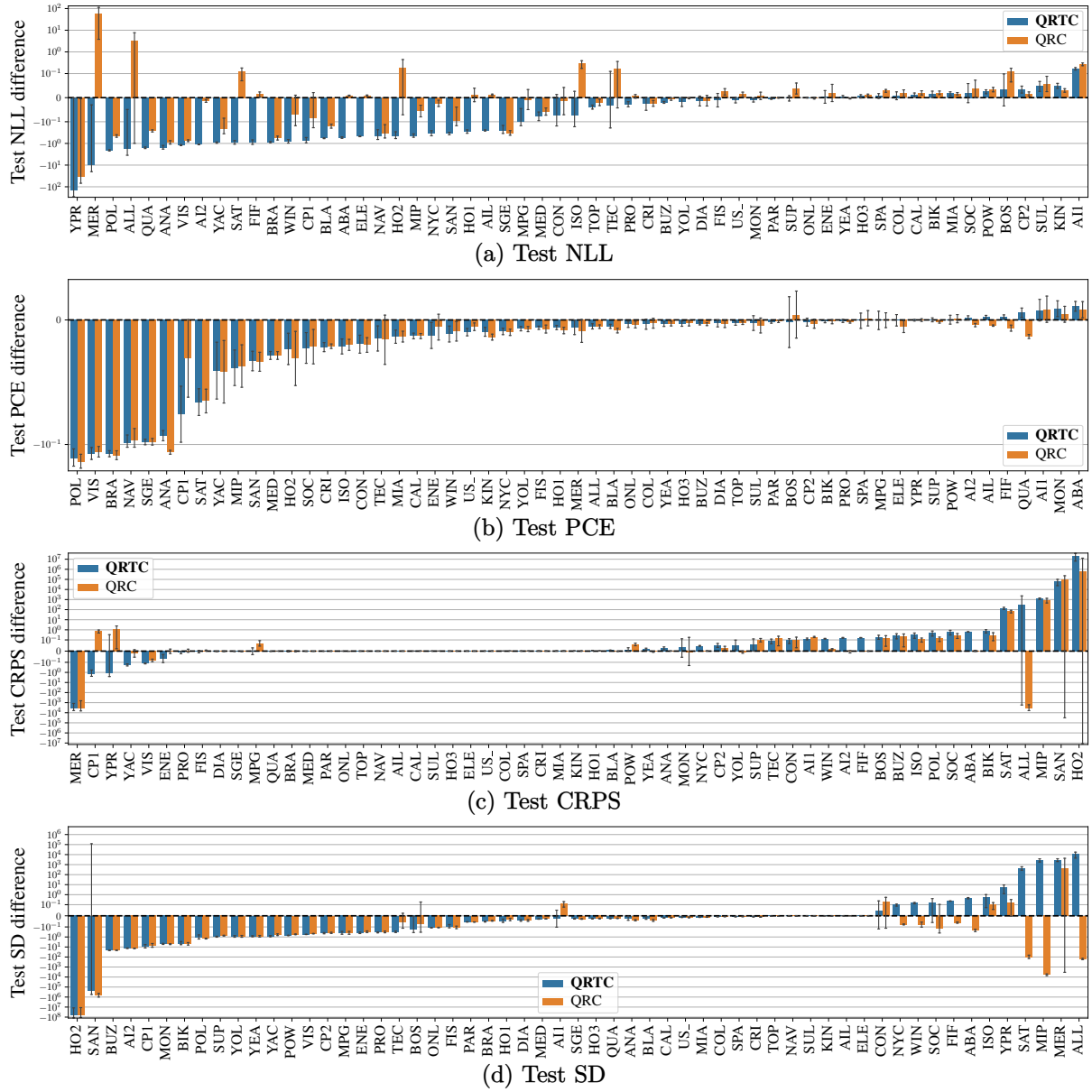
APPENDIX D

Supplementary Material for Chapter 4

D.1. Additional Results

D.1.1 Detailed Metrics on Individual Datasets

For a more comprehensive view, Figure D.1 presents a diagram analogous to Figure 4.2 from the main text, but extends the comparison across NLL, PCE, CRPS, and SD metrics. Additionally, these diagrams incorporate datasets previously omitted in Section 4.4.1. With respect to NLL, `QRTC` consistently surpasses `QRC` in the majority of datasets. In terms of PCE, both post-hoc methods perform similarly. In terms of CRPS, both post-hoc methods display comparable performances but are sometimes outperformed by `BASE`. Analyzing SD, `QRC` exhibits greater sharpness than `BASE` in nearly all instances, while `QRTC` sometimes does not exhibit increased sharpness.



(a) Test NLL

(b) Test PCE

(c) Test CRPS

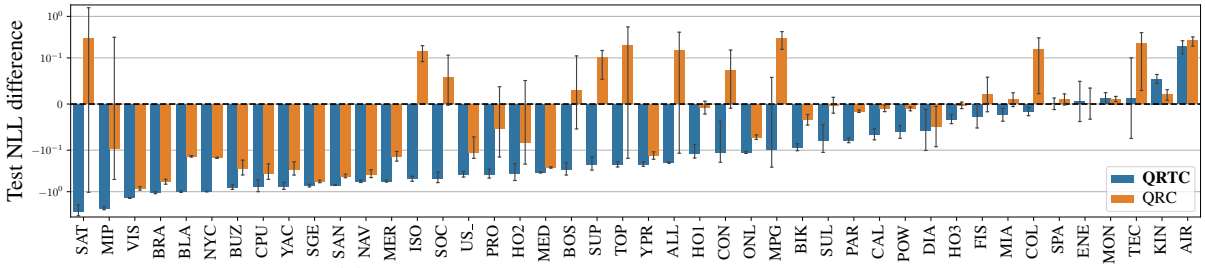
(d) Test SD

Figure D.1: Comparison of `QRTC` and `QRC` w.r.t. `BASE` by showing the difference between the compared methods and `BASE` according to a given metric, in average over 5 runs.

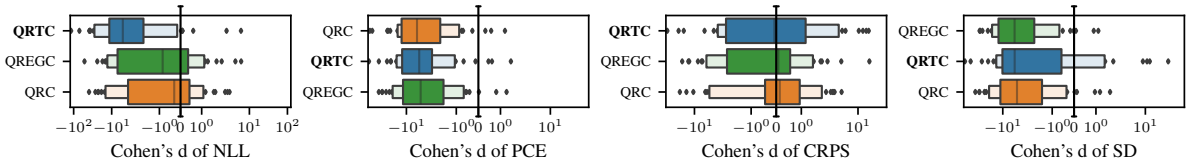
D.1.2 Results on Different Base Predictors

We present detailed results on the significance of the base predictor in influencing the performance of QRT, as discussed in Section 4.4.4 of the main paper. While our primary experiments utilize a 3-layer MLP predicting a mixture of three Gaussians, we also explore both less flexible mixtures with a single Gaussian and more flexible mixtures comprising ten Gaussians. Additionally, we evaluate a neural network adopting a ResNet-like architecture, referred to as ResNet. In Figures D.2 to D.4, we follow the exact same setup as in the main experiments except that the underlying neural network is modified.

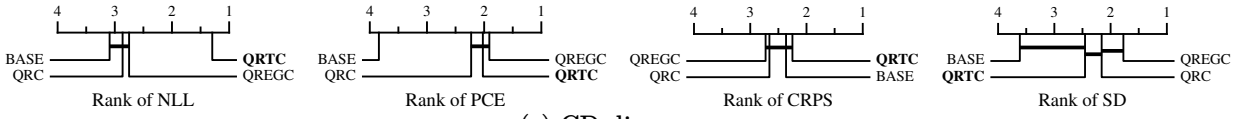
Figure D.2 presents the results where the neural network is 3-layer MLP predicting a single Gaussian (i.e., one mean and one standard deviation). In this misspecified case, we observe on Figure D.2a that, on many datasets, both `QRTC` and `QRC` provide an improvement in NLL compared to `BASE`, despite `BASE` having access to the calibration data. Moreover, `QRTC` provides an improvement in NLL compared to `QRC` in almost all cases. As in the main experiments, `QRTC`, `QRC` and `QREG` are all able to provide a significant improvement in PCE compared to `BASE`, with no significant difference between these three post-hoc models. There is also no significant difference in CRPS.



(a) Difference of test NLL compared to `BASE`.



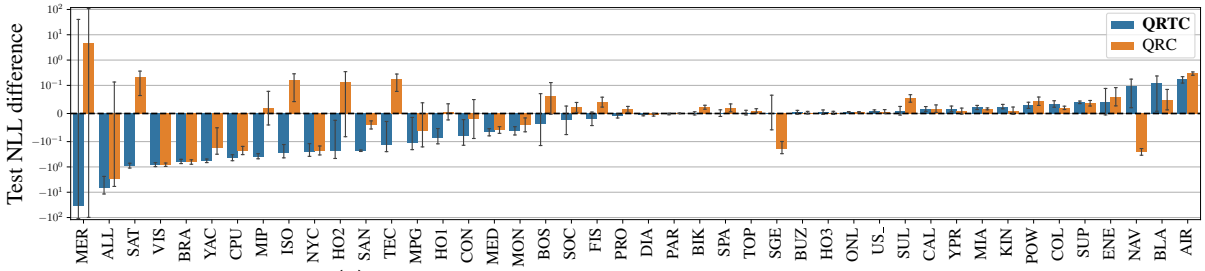
(b) Letter-value plots showing Cohen's d for different metrics w.r.t. `BASE`.



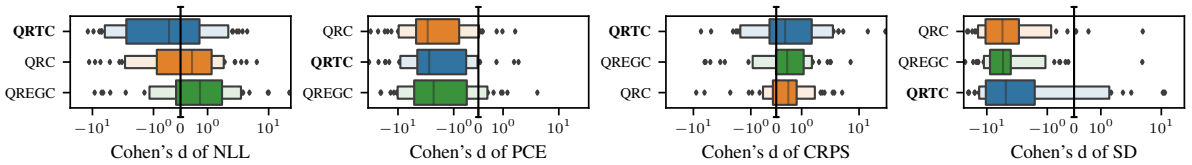
(c) CD diagrams

Figure D.2: Same setup as the main experiments (Figure 4.3 in the main text), except that the underlying neural network produces a single Gaussian instead of a mixture of 3 Gaussians.

Figure D.3 shows the same experiment except that the underlying neural network produces a mixture of 10 Gaussians (i.e., 10 means, 10 standard deviations, and 10 weights for each mixture component), offering high flexibility. In this case, `QRTC` provides an improvement in NLL in slightly more than half the datasets and the improvement is not significant, in contrast to the case of mixtures of size one and three. However, if we compare the post-hoc models, `QRTC` is still significantly better than `QRC` and `QREGC` in terms of NLL while achieving a similar PCE as `QRC` and `QREGC`. In terms of CRPS, `BASE` is slightly better than the post-hoc methods, but not significantly, which could be explained by the fact that the training dataset of `BASE` also contains the calibration data. All post-hoc methods achieve a similar CRPS. Finally, all post-hoc methods are significantly sharper than `BASE`, with `QREG` being slightly sharper than `QRTC` and `QRC`.



(a) Difference of test NLL compared to `BASE`.



(b) Letter-value plots showing Cohen's d for different metrics w.r.t. `BASE`.

(c) CD diagrams

Figure D.3: Same setup as the main experiments (Figure 4.3 in the main text), except that the underlying neural network produces a mixture of 10 Gaussians instead of a mixture of 3 Gaussians.

Figure D.4 shows the same experiments except that the model is a ResNet-like architecture predicting a mixture of size 3, with 18 fully-connected hidden layers in total. The architecture was proposed by Gorishniy et al. (2021) and implemented with the default hyperparameters of Grinsztajn et al. (2022). The architecture from Gorishniy et al. (2021) is reproduced here for completeness:

$$
\begin{aligned}
\texttt{ResNet}(x) &= \texttt{Prediction}(\texttt{ResNetBlock}(\ldots\texttt{ResNetBlock}(\texttt{Linear}(x)))) \\
\texttt{ResNetBlock}(x) &= x + \texttt{Dropout}(\texttt{Linear}(\texttt{Dropout}(\texttt{ReLU}(\texttt{Linear}(\texttt{BatchNorm}(x)))))) \\
\texttt{Prediction}(x) &= \texttt{Linear}(\texttt{ReLU}(\texttt{BatchNorm}(x)))
\end{aligned}
$$

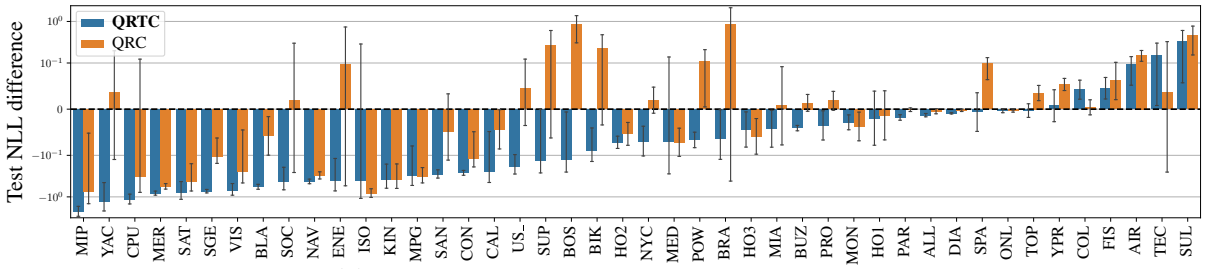
Since we predict a mixture of size $K = 3$, $\texttt{Output}(x)$ is of dimension $K * 3 = 9$. As for our MLP model, $\texttt{Output}(x)$ is split into $\mu(x)$, $\rho(x)$ and $l(x)$. Then, we define $\sigma(x) = \texttt{Softplus}(\rho(x))$ and $w(x) = \texttt{Softmax}(l(x))$. Finally, the mixture is defined as:

$$
f_\theta(y \mid x) = \sum_{k=1}^{K} w_k(x)\mathcal{N}(y; \mu_k(x), \sigma_k^2(x)),
$$

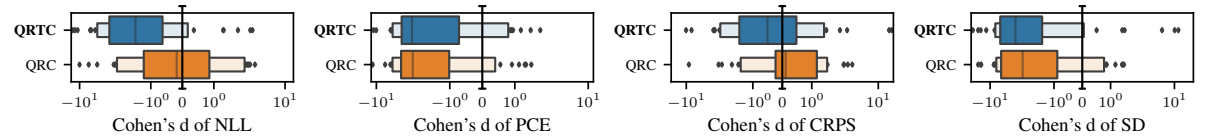
where $\mathcal{N}(y; \mu, \sigma^2)$ is the density of a normal distribution with mean μ and standard deviation σ evaluated at y.

Figure D.4a shows that `QRTC` remains advantageous even in deep models, with a notable improvement in NLL on most datasets compared to `QRC`. Similarly, observations from Figure D.4 align

with previous findings in PCE. Finally, `QRTC` is both significantly better than `QRC` in CRPS and significantly sharper.



(a) Difference of test NLL compared to `BASE`.

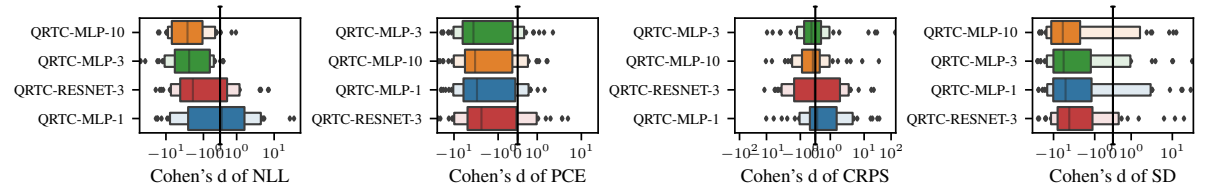


(b) Letter-value plots showing Cohen's d for different metrics w.r.t. `BASE`.

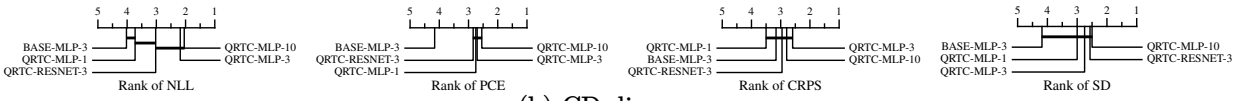
(c) CD diagrams

Figure D.4: Same setup as the main experiments (Figure 4.3 in the main text), except that the underlying neural network is a ResNet.

Finally, we provide a comparison of the performance of `QRTC` on all base predictors under consideration. Each model is denoted by `QRTC-<BM>-`K where `<BM>` is the base predictor and K is the mixture size. As illustrated in Figure D.5, mixtures of size 3 and 10 achieve the best NLL and CRPS, and a simple MLP achieves a better performance than a ResNet on these datasets.



(a) Letter-value plots showing Cohen's d for different metrics w.r.t. `BASE` (using an MLP model and mixture predictions of size 3 as in the main text).



(b) CD diagrams

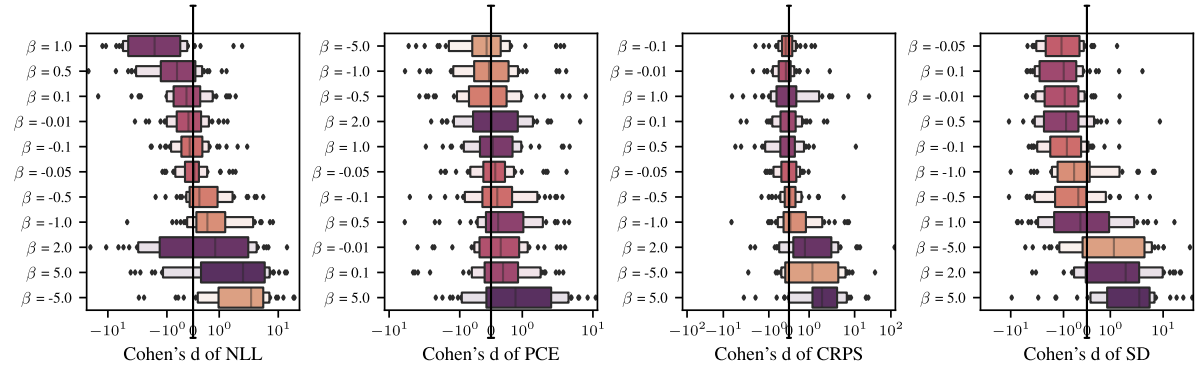
Figure D.5: Comparison of `QRTC` with different base predictors.

D.1.3 Results With Different Values of β

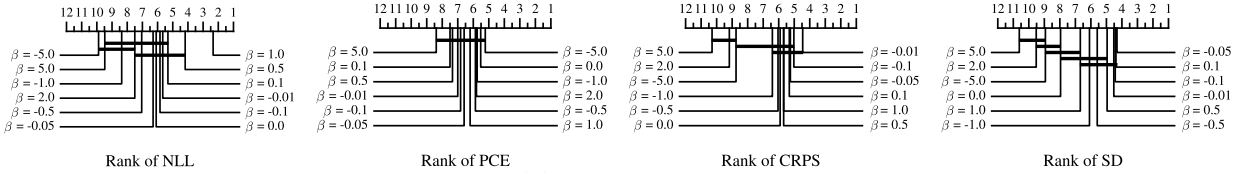
We provide detailed results regarding the hyperparameter β of Algorithm 8 in the main text. As discussed in Section 4.3.3, it is possible to design an algorithm unifying QRTC, QRC and QREGC, where the methods only differ by the hyperparameter β. We assume here that quantile recalibration is applied (C = True) and only consider variations of the hyperparameter β. As discussed previously, a value of $\beta = 1$ corresponds to `QRTC`, $\beta = 0$ corresponds to `QRC` and tuning β in order to minimize $\text{PCE}(F_\theta)$ corresponds to `QREGC` with regularization strength $\lambda = -\beta$.

In Figure D.6, we provide results with different values of β. Values of β between 0 and 1 can be considered as an intermediate version between `QRC` and `QRTC`, while negative values correspond to `QREGC`. We also explore values greater than 1 to visualize trends.

As expected, $\beta = 1$, corresponding to the NLL decomposition of the recalibrated model, obtains the best NLL, and is significantly better than other values of β. In terms of CRPS, there is no significant differences for values of β between 0 and 1.



(a) Boxplots of Cohen's d of different metrics on all datasets, w.r.t. `QRC`.



(b) CD diagrams

Figure D.6: Comparison of different values of the hyperparameter β.

D.1.4 Relationship Between the Discreteness of a Dataset and the Performance of Different Models

In this section, we discuss the issue that certain datasets from the UCI and OpenML benchmarks may not be suitable for regression, as introduced in Section 4.4.1. Although we consider regression benchmarks, we observe that, in many datasets, the target Y presents some level of discreteness. This is not surprising due to the finite precision of numbers and to the roundings that can appear during data collection. For example, Table A.1 shows that, on 44 out of 57 datasets, more than half of the targets Y appear at least twice. This potential issue is more important for certain datasets where some values of the targets Y appear very frequently.

We propose to identify these datasets using the proportions of values Y in the dataset that are among the 10 most frequent values, and we call this proportion the level of discreteness. For example, if a dataset only contains 10 distinct values, the level of discreteness would be 100%. Table A.1 in the Supplementary Material shows that 13 out of 57 datasets have a level of discreteness above 0.5, i.e., more than half of the targets are among the 10 most frequent ones. These datasets appear in all 4 benchmark suites.

In Figures D.7 and D.8, we plot for each dataset the Cohen's d of different metrics, averaged over 5 runs, compared to the discreteness level of the dataset. For the NLL, CRPS and PCE, negative values of the Cohen's d correspond to an improvement. In order to show the average Cohen's d conditional on the discreteness level, we provide an isotonic regression estimate in red.

Figure D.7 shows that `QRTC` tends to provide a decreased NLL and increased CRPS for higher discreteness levels. This can be explained by the ability of `QRTC` to put a high likelihood on a few values by minimizing the NLL but neglect other aspects of the distributions. While previous work (Kohonen and Suomela, 2006) has highlighted the unsuitability of NLL as a metric for discrete datasets, they are still commonly found in regression benchmarks. For example, Lakshminarayanan, Pritzel, et al. (2017) and Amini et al. (2020) trained a model based on NLL on the `wine_quality` dataset for which the output variable only takes 7 distinct values.

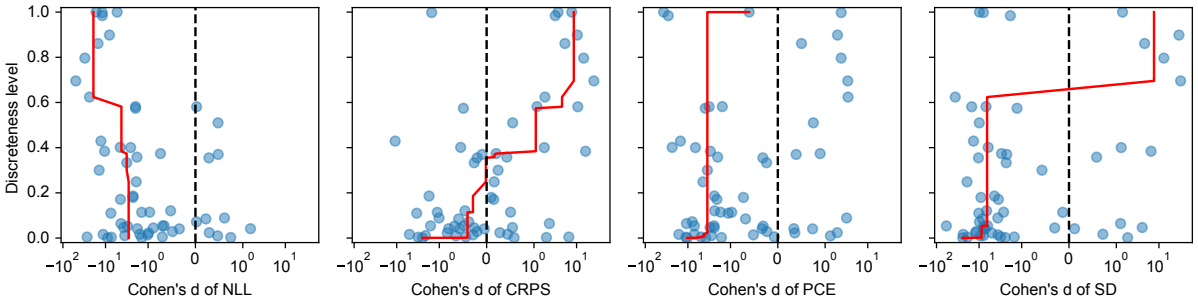


Figure D.7: Cohen's d of different metrics compared to the discreteness level of a dataset for the `QRTC` model relative to the `BASE` model.

Figure D.8 shows the same metrics for `QRC`, where we observe that the improvement in NLL is less marked on datasets with a higher discreteness level. The CRPS, however, is not decreased as much as with `QRTC`, which suggests that `QRTC` is not suitable for datasets with a high level of discreteness. We do not observe a notable trend in terms of PCE.

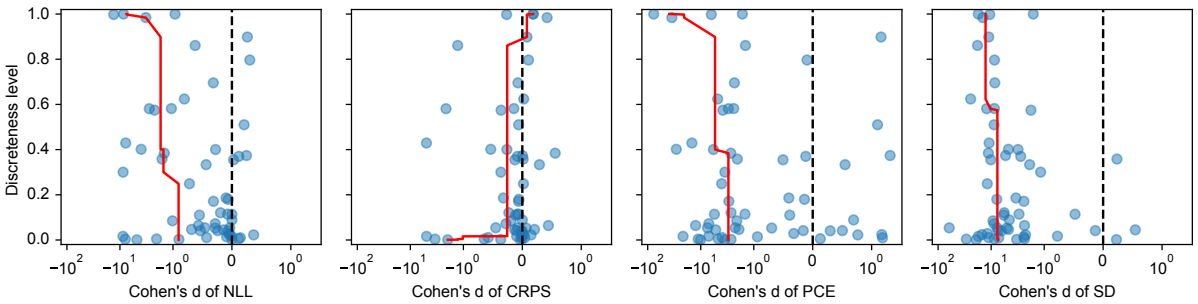
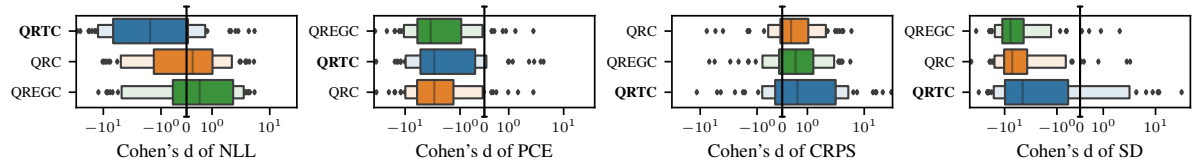


Figure D.8: Cohen's d of different metrics compared to the discreteness level of a dataset for the `QRC` model relative to the `BASE` model.

D.1.5 Results on All Datasets

As discussed in Section 4.4.1, we provide the full results including the datasets with a high discreteness level. Despite the potential issues discussed in Section D.1.4, the conclusions drawn in Section 4.4 remain unchanged. `QRTC` demonstrates a significant improvement in NLL with a negligible loss in CRPS. Figure detailing metrics on the individual datasets are also available in Section D.1.1.



(a) Letter-value plots showing Cohen's d for different metrics w.r.t. `BASE`.

(b) Critical difference diagrams for different metrics.

Figure D.9: Same setup as the main experiments (Figure 4.3 in the main text), with all the datasets.

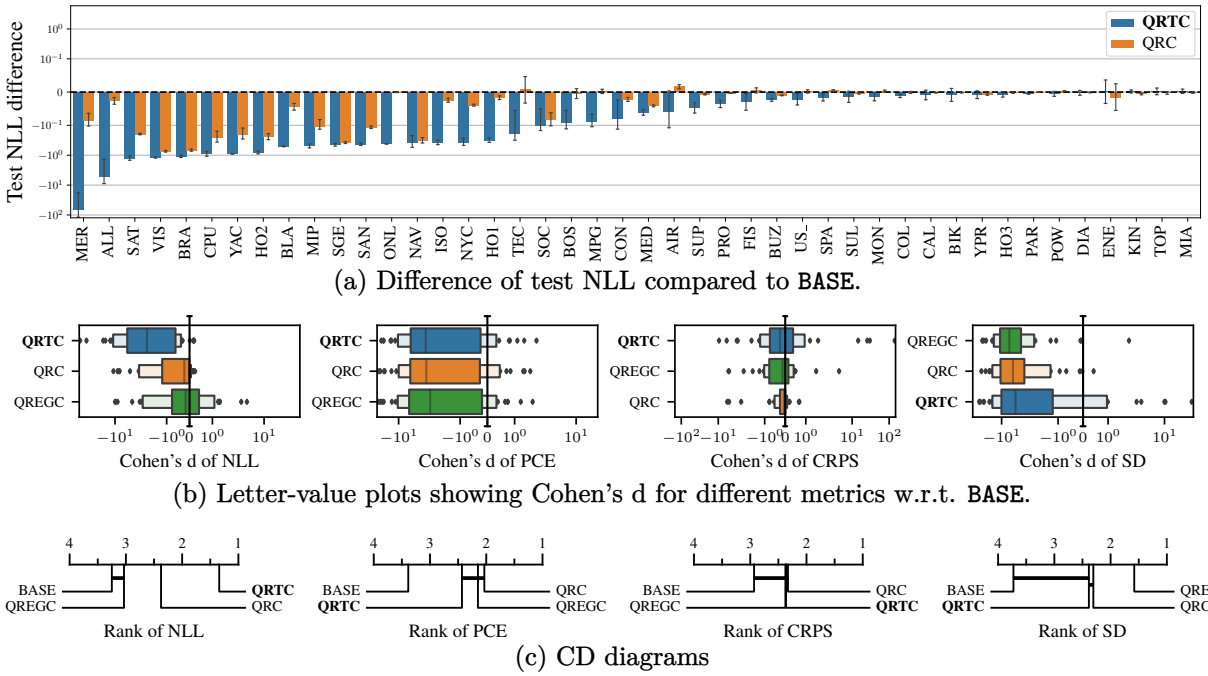
D.1.6 Results Where Base Does Not Have Access to Calibration Data

As discussed in Section 4.4.2, we aimed to provide a fair comparison between `BASE` and the post-hoc methods by training it on a larger dataset than the post-hoc methods `QRTC`, `QRC` and `QREGC` in all of our experiments. Since the post-hoc methods benefit from the calibration data during the post-hoc step, all methods end up benefiting from the same amount of data.

In order to gain deeper insights into the effect of the post-hoc step, we repeat our main experiments with the exception that `BASE` does not have access to calibration data. Thus, `QRC` has the same base predictor than `BASE` and performs an additional post-hoc step.

In Figure D.10a, `QRC` shows that the post-hoc step never degrades NLL and sometimes results in a notable NLL improvement, which suggests that a post-hoc step on additional calibration

data is always beneficial. Figure D.10c shows that `QRC` results in a significant NLL improvement compared to `BASE`, and `QRTC` results in an additional significant NLL improvement compared to `QRC`. In terms of CRPS, there is no significant difference, and post-hoc methods result in sharper predictions.



(a) Difference of test NLL compared to `BASE`.

(b) Letter-value plots showing Cohen's d for different metrics w.r.t. `BASE`.

(c) CD diagrams

Figure D.10: Same setup as the main experiments (Figure 4.3 in the main text), except that `BASE` is not trained on the calibration data.

D.1.7 Computational Time

In Table D.1, we present a comparative analysis of the training time for various methods across all datasets. Notably, `QRTC` occasionally exhibits a training time that is approximately two times slower per epoch compared to `BASE`. As discussed in Section 4.3.4, this disparity can be attributed to the extra computational overhead associated with the computation of the calibration map, i.e., $-\frac{1}{B}\sum_{i=1}^{B}\log\phi_{\text{REFL}}(Z_i)$.

Table D.1: Comparison of the training time for different methods on all datasets.

	Training time			Number of epochs			Time per epoch		
Dataset	BASE	QREGC	QRTC	BASE	QREGC	QRTC	BASE	QREGC	QRTC
Airlines_DepDelay_10M	485.59	521.41	732.92	31.40	35.40	44.20	15.46	14.73	16.58
Allstate_Claims_Severity	284.53	1152.66	425.80	7.00	65.80	5.20	40.65	17.52	81.89
Buzzinsocialmedia_Twitter	923.56	997.29	863.84	84.60	93.00	53.20	10.92	10.72	16.24
MIP-2016-regression	14.97	22.07	39.65	167.00	181.40	228.00	0.09	0.12	0.17
Moneyball	7.55	22.62	14.63	78.00	141.00	74.60	0.10	0.16	0.20
SAT11-HAND-runtime-regression	119.73	150.92	113.64	241.40	232.60	182.60	0.50	0.65	0.62
Santander_transaction_value	24.01	26.02	26.60	3.80	5.00	3.80	6.32	5.20	7.00
Yolanda	364.63	834.97	406.50	6.60	55.40	9.80	55.25	15.07	41.48
abalone	35.27	31.20	77.67	52.60	49.60	107.80	0.67	0.63	0.72
boston	7.38	5.56	7.32	72.60	57.00	48.20	0.10	0.10	0.15
colleges	40.12	59.70	53.58	29.50	32.00	25.00	1.36	1.87	2.14
house_prices_nominal	6.74	7.07	10.22	51.80	7.50	28.20	0.13	0.94	0.36
quake	26.22	44.89	166.82	100.20	132.20	577.40	0.26	0.34	0.29
socmob	9.04	20.84	15.47	112.60	118.60	67.80	0.08	0.18	0.23
space_ga	42.02	59.70	67.64	137.80	167.80	142.40	0.30	0.36	0.48
tecator	11.87	7.37	7.83	195.00	185.80	111.80	0.06	0.04	0.07
topo_2_1	41.00	38.62	33.81	15.80	14.20	11.40	2.60	2.72	2.97
us_crime	10.19	28.18	16.38	21.00	67.40	22.60	0.49	0.42	0.72
Ailerons	66.42	87.46	136.04	21.00	37.80	41.00	3.16	2.31	3.32
Bike_Sharing_Demand	291.76	355.97	328.56	161.00	162.20	136.80	1.81	2.19	2.40
Brazilian_houses	159.17	337.55	169.54	137.40	230.20	128.20	1.16	1.47	1.32
MiamiHousing2016	118.08	181.59	125.74	62.20	101.80	64.20	1.90	1.78	1.96
california	254.69	414.27	278.82	110.60	151.00	125.00	2.30	2.74	2.23
cpu_act	62.56	105.46	65.00	63.00	84.20	46.60	0.99	1.25	1.39
diamonds	646.27	755.47	782.51	107.00	88.60	85.40	6.04	8.53	9.16
elevators	92.17	97.59	308.20	21.00	30.20	103.00	4.39	3.23	2.99
fifa	392.70	634.12	297.08	207.80	288.20	107.60	1.89	2.20	2.76
house_16H	200.43	539.11	388.17	67.80	185.00	115.40	2.96	2.91	3.36
house_sales	138.38	325.24	121.14	34.60	113.80	29.40	4.00	2.86	4.12
isolet	128.97	222.28	187.13	136.20	222.60	167.00	0.95	1.00	1.12
medical_charges	687.66	984.32	840.41	62.20	75.00	49.80	11.06	13.12	16.88
nyc-taxi-green-dec-2016	1058.06	2083.84	1289.36	109.40	172.20	94.20	9.67	12.10	13.69
pol	164.51	364.97	409.18	91.80	165.00	214.60	1.79	2.21	1.91
sulfur	324.37	323.69	272.26	319.00	264.60	218.40	1.02	1.22	1.25
superconduct	342.96	553.18	514.55	147.80	196.20	164.40	2.32	2.82	3.13
wine_quality	83.41	163.38	157.51	110.20	175.00	148.60	0.76	0.93	1.06
year	327.90	392.19	465.72	7.40	9.00	9.40	44.31	43.58	49.54
Mercedes_Benz_Greener_Manufacturing	16.23	24.78	26.14	8.60	9.80	7.80	1.89	2.53	3.35
OnlineNewsPopularity	137.57	182.44	232.98	7.80	10.20	14.60	17.64	17.89	15.96
SGEMM_GPU_kernel_performance	1229.68	1046.17	901.34	117.80	80.20	83.80	10.44	13.04	10.76
analcatdata_supreme	103.73	213.92	141.89	255.80	323.40	203.60	0.41	0.66	0.70
black_friday	558.56	742.27	1164.06	42.20	65.00	77.40	13.24	11.42	15.04
particulate-matter-ukair-2017	690.45	756.80	760.01	35.80	59.40	34.60	19.29	12.74	21.97
visualizing_soil	124.87	242.37	135.87	123.80	166.60	99.60	1.01	1.45	1.36
yprop_4_1	36.91	53.51	34.24	11.80	13.80	17.50	3.13	3.88	1.96
Airfoil	32.33	52.81	44.84	270.20	370.60	329.40	0.12	0.14	0.14
CPU	4.59	6.80	5.64	160.60	170.60	140.60	0.03	0.04	0.04
Concrete	12.40	20.60	16.92	179.80	149.40	119.40	0.07	0.14	0.14
Crime	4.76	8.83	7.27	43.00	53.80	42.20	0.11	0.16	0.17
Energy	14.45	25.44	22.17	219.00	219.00	201.80	0.07	0.12	0.11
Fish	6.13	10.07	11.06	76.20	66.60	72.20	0.08	0.15	0.15
Kin8nm	49.79	72.99	58.75	45.00	51.40	40.60	1.11	1.42	1.45
MPG	3.54	6.02	5.27	64.20	87.00	73.00	0.06	0.07	0.07
Naval	180.63	318.65	261.46	145.40	200.60	190.60	1.24	1.59	1.37
Power	174.46	207.21	238.07	203.80	186.20	193.00	0.86	1.11	1.23
Protein	1030.51	1329.29	1275.15	244.60	235.00	216.60	4.21	5.66	5.89
Yacht	6.73	8.85	11.13	189.40	169.40	228.40	0.04	0.05	0.05

D.1.8 Metrics Per Epoch

In this section, we compare the behavior of training and validation NLL and PCE for both `QRT` and `BASE` throughout the training process. We present learning curves for all the datasets considered in this study, sorted by the number of training points. For detailed information and the complete names of these datasets, please refer to Table A.1.

The setup mirrors the illustrative example presented in Section 4.3.2. The training curves are averaged over 5 runs, while the shaded area corresponds to one standard error. The vertical bars represent the epoch selected through early stopping, which minimizes the validation NLL, averaged over the 5 runs. The horizontal bars represent the average metric value at the selected epoch

across the 5 runs. We draw the same conclusions as in the illustrative example (Section 4.3.2). The CRPS and SD are not provided due to the high computational time required to computed these metrics after quantile recalibration.

In Figures D.11 and D.13, we observe that the NLL of `QRT` tends to be lower after the same number of epochs, indicating improved probabilistic predictions on both the training and validation datasets. Importantly, this improvement in NLL is consistent across datasets of different size. While the most significant enhancement in NLL is seen in datasets with a high level of discreteness such as `WIN`, `ANA` and `QUA`, noticeable improvements are also observed in most non-discrete datasets like `CP1`, `YAC` and `PAR`.

Referring to Figure D.12 and Figure D.14, we can observe that the PCE of `BASE` often exhibits higher variability across epochs compared to `QRT` on both the validation and training datasets. This phenomenon indicates the regularization effect of `QRT`. Additionally, we notice that the PCE is frequently lower at the same epochs for `QRT`, although there are instances where this is not the case.

After reaching a certain epoch (indicated by the vertical bar), the model starts to overfit, leading to an expected increase or stabilization of NLL and PCE on the validation dataset. On the other hand, the NLL on the training dataset continues to decrease as anticipated, while the PCE exhibits high variation depending on the dataset.

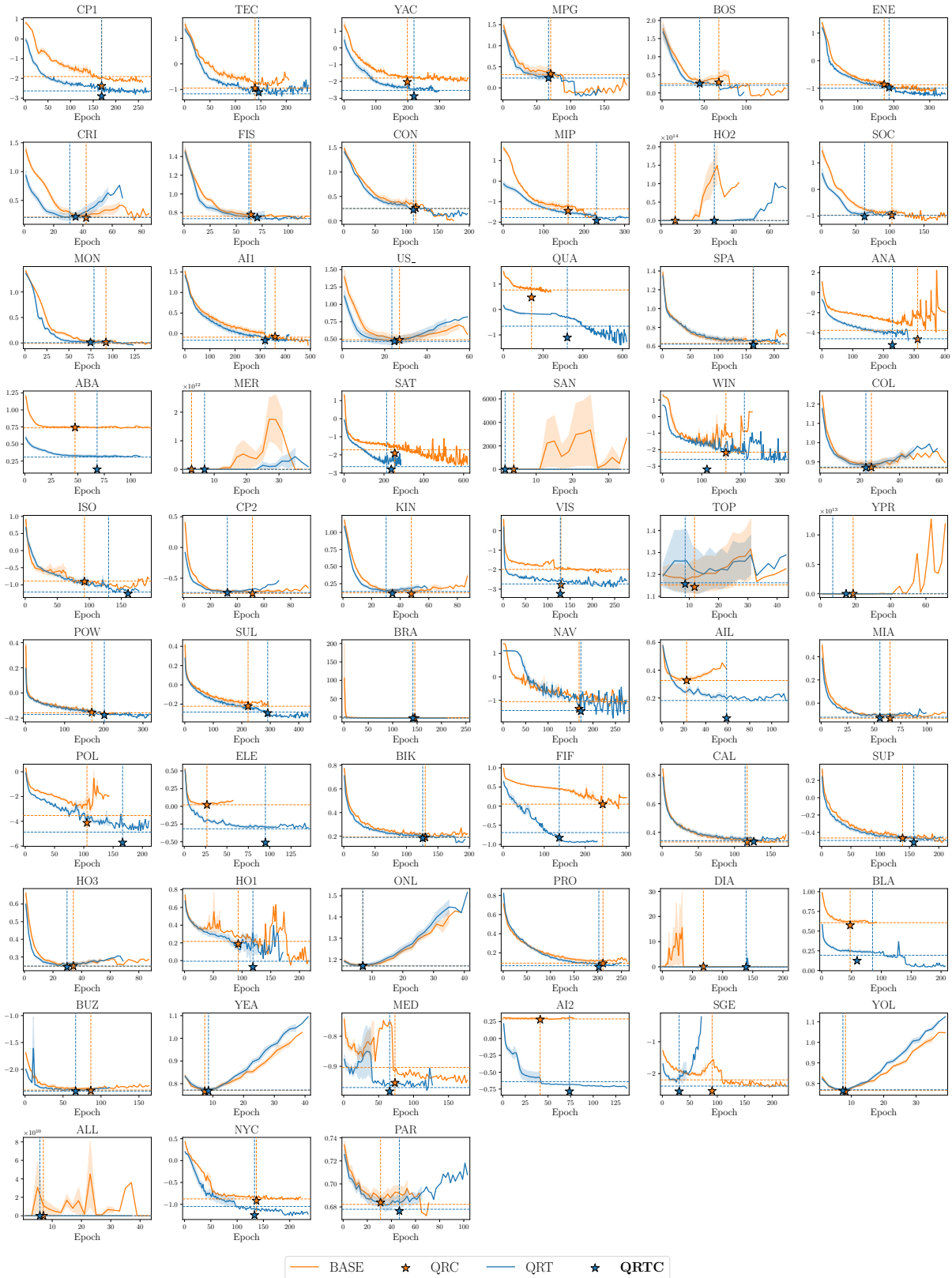


Figure D.11: NLL on the validation dataset per epoch.

Figure D.12: PCE on the validation dataset per epoch.

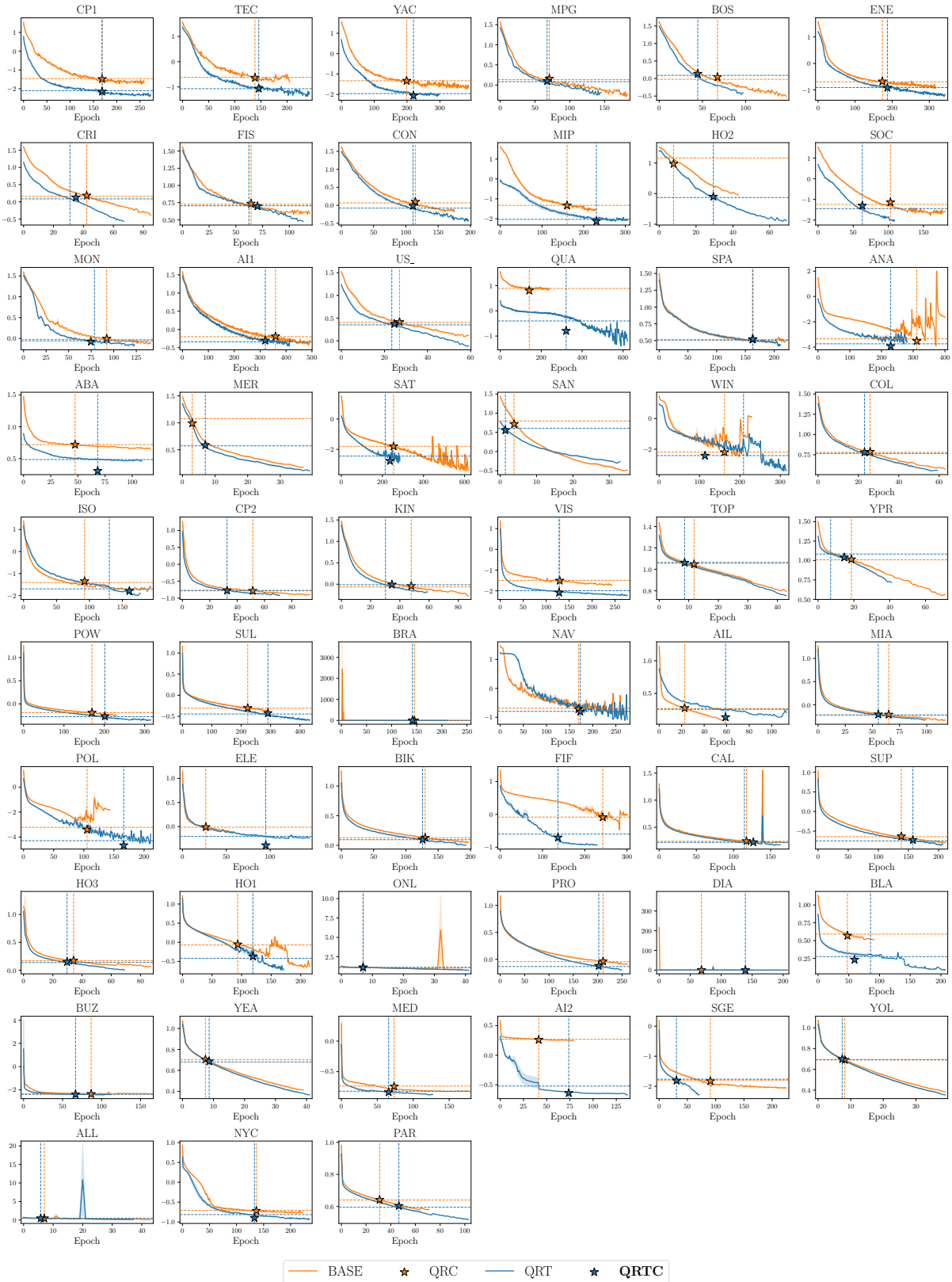


Figure D.13: NLL on the training dataset per epoch.

Figure D.14: PCE on the training dataset per epoch.

D.1.9 Examples of Predictions

To offer a deeper understanding of the shape of the predictions, Figures D.15 to D.20 display prediction examples across various datasets. In these figures, each row illustrates density predictions from the same model, while every column denotes the same instance, with the realization y marked by a green vertical bar. The associated NLL for each prediction is also presented.

Figures D.15 to D.18 are from datasets where `QRTC` outperformed `QRC` in terms of NLL. Notably, within these, `QRTC` exhibits heightened confidence in its predictions for Figures D.16 and D.17. However, in other datasets, the NLL improvements are more subtle. Figure D.20 represents predictions on a dataset with a high level of discreteness which has not been considered in the main experiments. In this case, `QRTC` assigns a high density to individual values y, highlighting a limitation of NLL minimization, as discussed in Section D.1.4. Overall, the shape of the predictions can vary greatly w.r.t. the dataset.

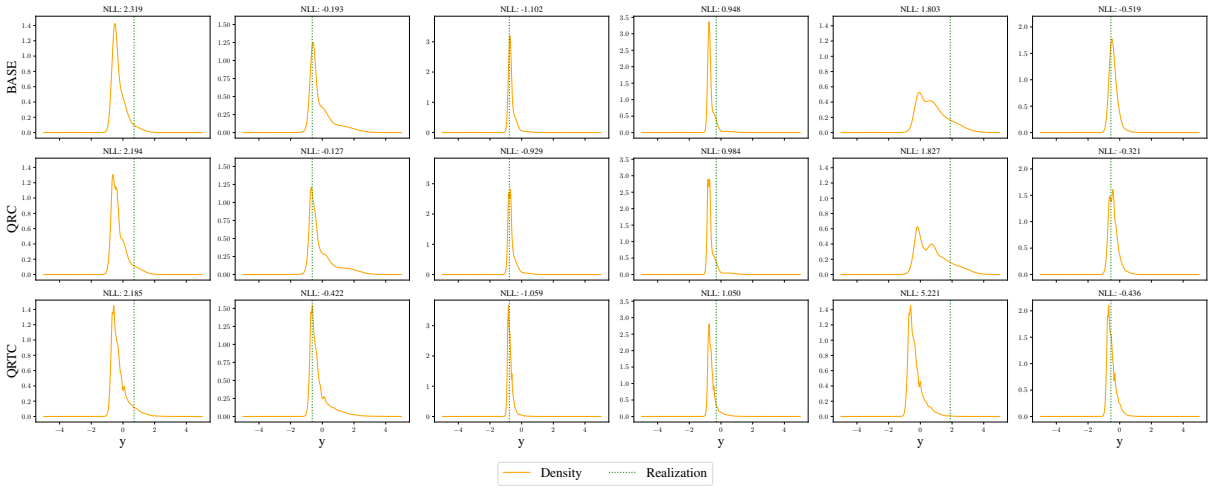


Figure D.15: Examples of predictions of `BASE`, `QRC` and `QRTC` on dataset `Allstate_Claims_Severity` (ALL).

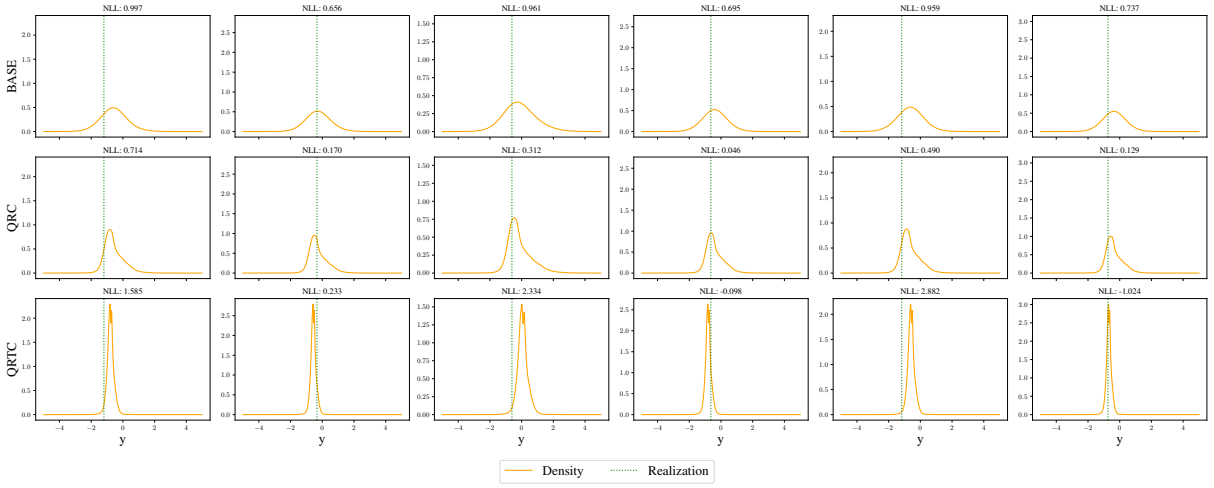


Figure D.16: Predictions of `BASE`, `QRC` and `QRTC` on dataset `house_prices_nominal` (HO2).

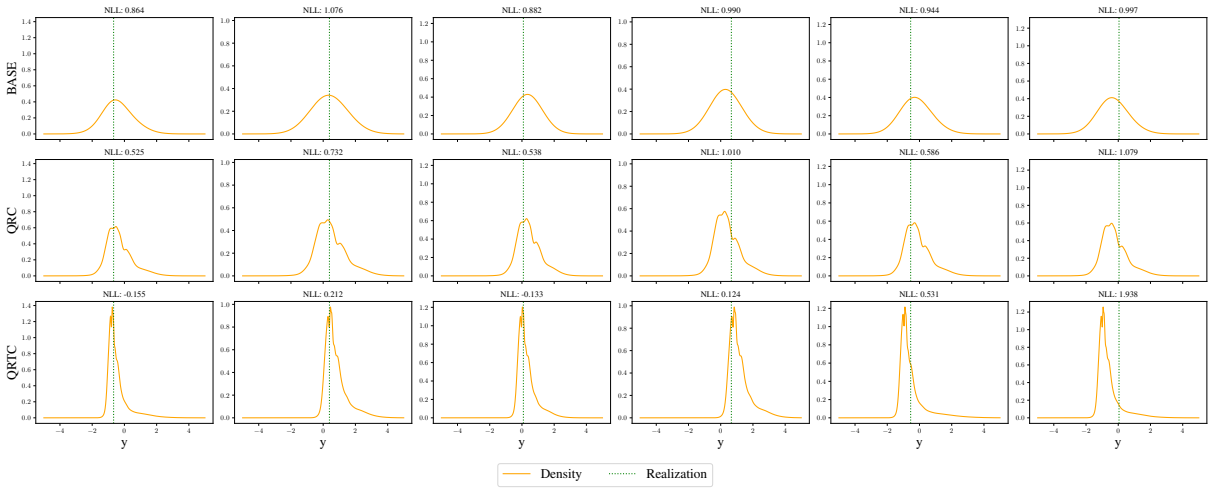


Figure D.17: Predictions of `BASE`, `QRC` and `QRTC` on dataset `Mercedes_Benz_Greener_-Manufacturing` (MER).

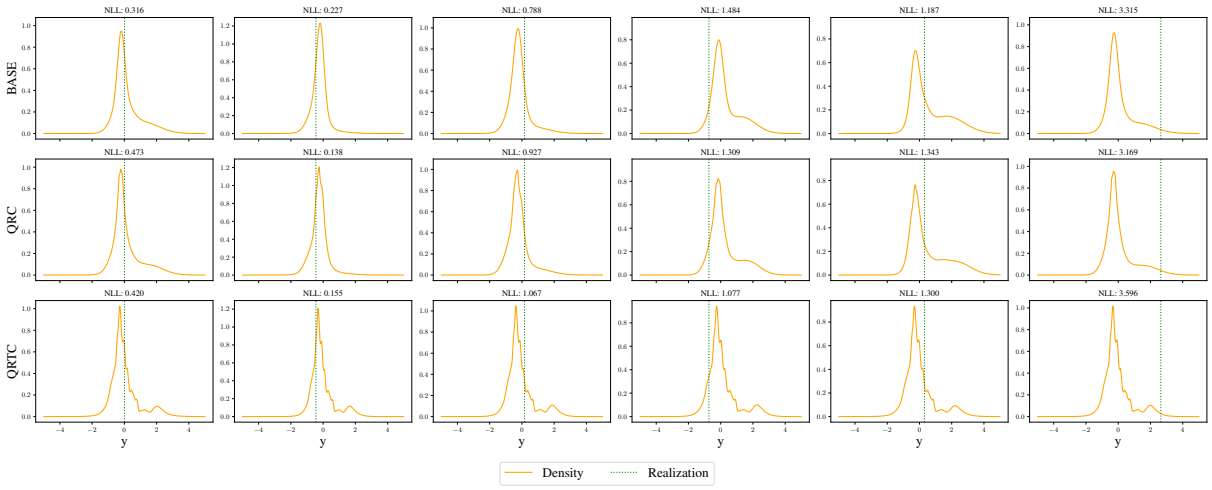


Figure D.18: Predictions of `BASE`, `QRC` and `QRTC` on dataset `yprop_4_1` (YPR).

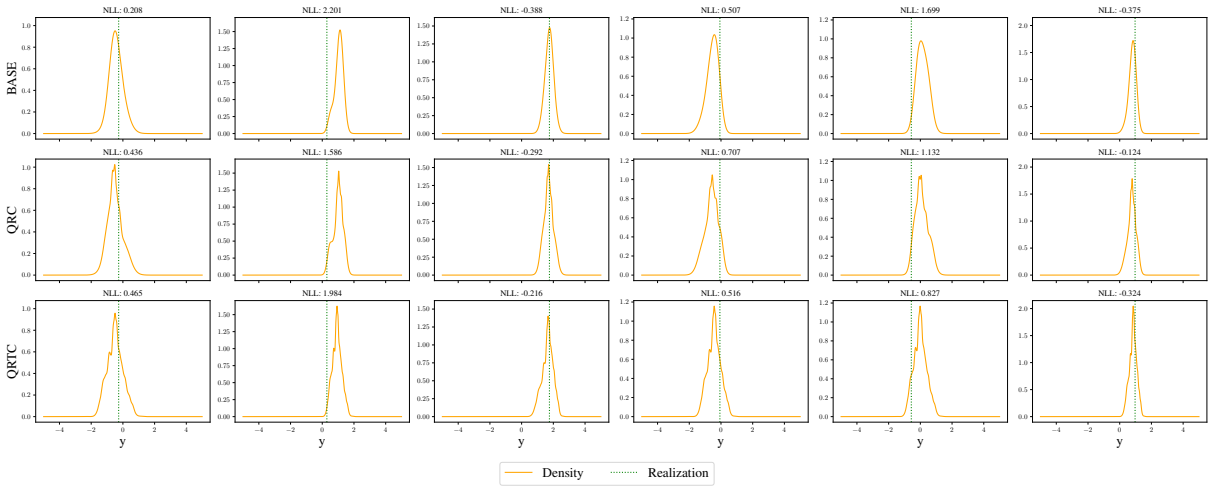


Figure D.19: Predictions of `BASE`, `QRC` and `QRTC` on dataset `space_ga` (SPA).

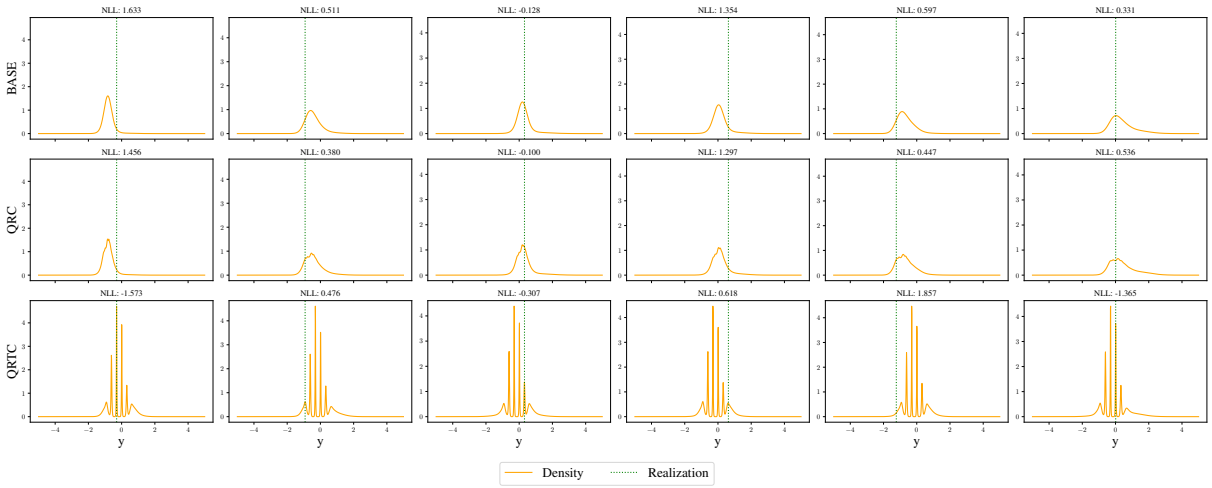
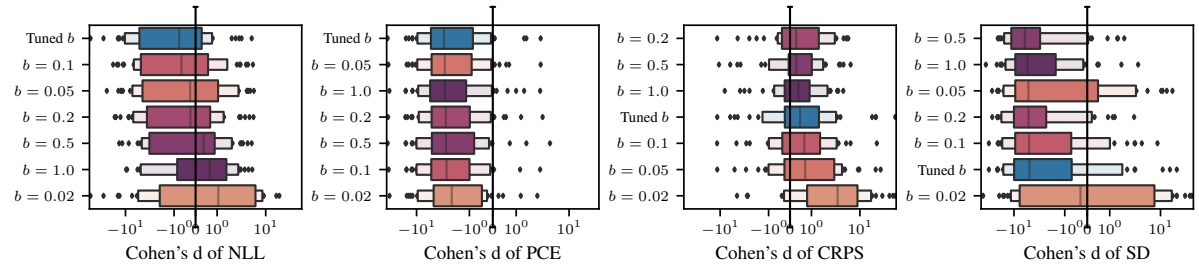


Figure D.20: Predictions of `BASE`, `QRC` and `QRTC` on dataset `abalone` (ABA).

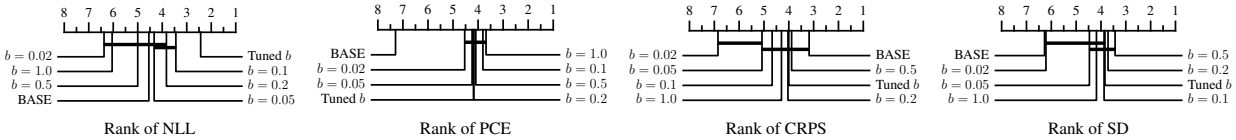
D.2. Hyperparameters

D.2.1 Impact of the Bandwidth Hyperparameter b

We evaluate the effect of tuning the bandwidth hyperparameter b in QRT. In Figure D.21, the bandwidth is either selected by minimizing the validation NLL from the set 0.02, 0.05, 0.1, 0.2, 0.5, 1 (denoted by Tuned b), or it is set to a fixed value. The results show that tuning b results in a significant improvement in NLL compared to fixed values of b. Values of 0.1, 0.2 and 0.05 yield the best NLL improvement while values of 0.2 and 0.5 yield the best CRPS improvement compared to `BASE`.



(a) Boxplots of Cohen's d of different metrics on all datasets, w.r.t. **BASE**.



(b) CD diagrams

Figure D.21: Comparison of QRTC with different values of the hyperparameter b.

D.2.2 Impact of the Size of the Calibration Map

As discussed in Section 4.3.3, we investigate the impact of computing the calibration map from a dataset of size M sampled randomly from the training dataset instead of the current batch. Specifically, this would correspond to changing the training loop of Algorithm 8 as depicted by Algorithm 10.

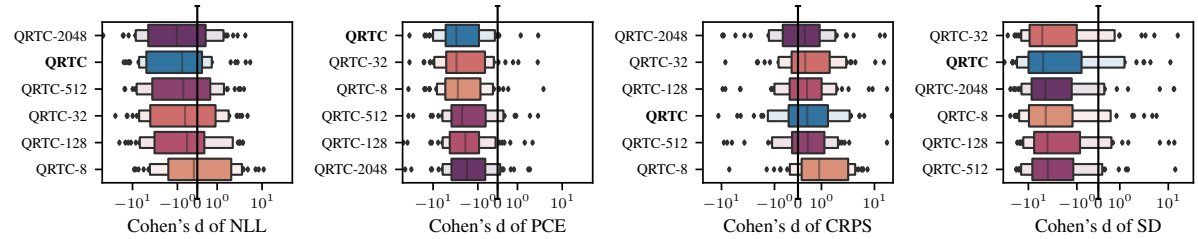
Algorithm 10 QRT framework where ϕ_{REFL} is computed from a random sample of the training dataset.

Input: Predictive CDF $\hat{F}_{Y|X}$, training dataset $\mathcal{D}_{\text{train}}$, sample size m, regularization strength $\beta \in \mathbb{R}$
$m \leftarrow \min\{m, |\mathcal{D}_{\text{train}}|\}$
for each minibatch $\{(X^{(i)}, Y^{(i)})\}_{i=1}^{B} \subseteq \mathcal{D}_{\text{train}}$, until early stopping **do**
  Sample $\{(X'^{(i)}, Y'^{(i)})\}_{i=1}^{m}$ from $\mathcal{D}_{\text{train}}$ without replacement
  $\hat{U}'^{(i)} \leftarrow \hat{F}_{Y|X=X'^{(i)}}(Y'^{(i)})$ for $i = 1, \ldots, m$
  Define ϕ_{REFL} from $\hat{U}'^{(1)}, \ldots, \hat{U}'^{(m)}$ using (4.8)
  $\hat{U}^{(i)} \leftarrow \hat{F}_{Y|X=X^{(i)}}(Y^{(i)})$ for $i = 1, \ldots, B$
  $$\mathcal{L}(\theta) = -\frac{1}{B}\sum_{i=1}^{B} \underbrace{\left[\log \hat{f}_{Y|X=X^{(i)}}(Y^{(i)}) + \beta \log \phi_{\text{REFL}}(\hat{U}^{(i)})\right]}_{\log \hat{f}'_{Y|X=X^{(i)}}(Y^{(i)})}$$
  Update parameters θ using $\nabla_\theta \mathcal{L}(\theta)$
end for

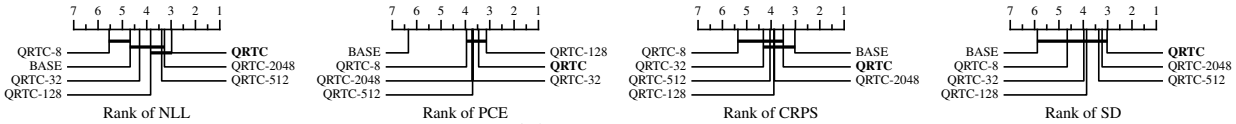
While the approach proposed in the main text requires B neural network evaluations per minibatch, Algorithm 10 requires $M + B$ neural network evaluations per minibatch, making it relatively slower.

In Figure D.22, we investigate the performance of QRTC using Algorithm 10 with calibration

maps of size M, and denote these models `QRTC-`M. The model `QRTC` in blue corresponds to the same model as in the main paper, with a calibration map computed from the current batch of size $B = 512$. It is worth noting that the post-hoc step is still performed on a calibration dataset of the same size for all models. In terms of NLL, models with a larger calibration map tend to perform better. In terms of PCE, all post-hoc models perform similarly. While no decisive conclusions can be drawn, Figure D.22b suggests that larger calibration maps tend to result in improved CRPS and sharper predictions. Overall, estimating the NLL of QRT using a larger calibration map tends to give more accurate predictions.



(a) Boxplots of Cohen's d of different metrics on all datasets, w.r.t. `BASE`.



(b) CD diagrams

Figure D.22: Comparison of `QRTC`, where the calibration map has been computed on calibration datasets of different sizes.

D.3. Kernel Density Estimation on a Finite Domain

We provide more motivation and details regarding the calibration map Φ_{REFL} discussed in Section 4.3.1 in the main paper.

The limitation of a standard kernel density estimation within a finite domain $[a, b]$ using a kernel like the logistic distribution is that the resulting distribution becomes ill-defined due to non-null density values extending below a and beyond b. In the following, to simplify notation, we denote Φ_{KDE} and ϕ_{KDE} by F and f respectively.

We would like to highlight that following our independent development of the "Reflected Kernel", we later discovered that this concept had originally been introduced by Blasiok and Nakkiran (2023).

D.3.1 Truncated distribution

A standard approach is to truncate the distribution and redistribute the density below a and above b, namely $F(b) - F(a)$, such that the distribution is normalized. The resulting CDF is:

$$\Phi_{\text{TRUNC}}(x) = \begin{cases} {}^{F(x)-F(a)}/_{F(b)-F(a)} & \text{if } x \in [a, b] \\ 0 & \text{if } x < a \\ 1 & \text{if } x > b \end{cases} \tag{D.1}$$

and the resulting PDF is:

$$\phi_{\text{TRUNC}}(x) = \begin{cases} {}^{f(x)}/_{F(b)-F(a)} & \text{if } x \in [a, b] \\ 0 & \text{if } x \notin [a, b]. \end{cases} \tag{D.2}$$

A drawback of truncating the distribution on a finite domain is that the resulting distribution will be biased to have lower density close to a and b and higher density elsewhere, as illustrated on Figure D.23.

D.3.2 Proposed approach: Reflected distribution

To remedy this problem, we define a new PDF ϕ_{REFL} that "reflects" the base density f around a and b. More precisely, for a given $z > 0$, the density in $a - z$ is redistributed to $a + z$ and the density in $b + z$ is redistributed to $b - z$. We assume that the density f is not too spread out, specifically $f(x) = 0$ for $x \notin [a - (b - a), b + (b - a)]$. The resulting CDF is defined by:

$$\Phi_{\text{REFL}}(x) = \begin{cases} F(x) - F(2a - x) + 1 - F(2b - x) & \text{if } x \in [a, b] \\ 0 & \text{if } x < a \\ 1 & \text{if } x > b \end{cases} \tag{D.3}$$

and the corresponding PDF is defined by:

$$\phi_{\text{REFL}}(x) = \begin{cases} f(x) + f(2a - x) + f(2b - x) & \text{if } x \in [a, b] \\ 0 & \text{if } x \notin [a, b]. \end{cases} \tag{D.4}$$

Figure D.23 compares four methods to estimate the calibration map from PIT realizations $Z_1, \dots, Z_N$. The method Φ_{EMP} was introduced in Section 4.2 and corresponds to the empirical CDF, which is not smooth. In contrast, the methods Φ_{KDE}, Φ_{TRUNC} and Φ_{REFL} offer smooth estimations. This figure shows that Φ_{REFL} is closer to the empirical CDF than Φ_{TRUNC} and the value of the corresponding PDF ϕ_{REFL} is not overestimated, suggesting the superiority of this estimator.

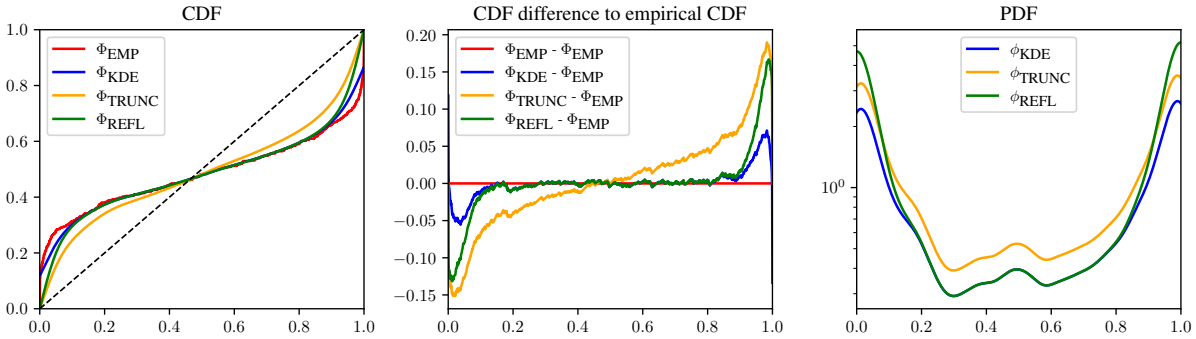
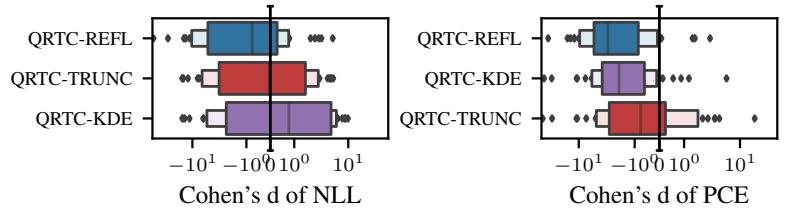
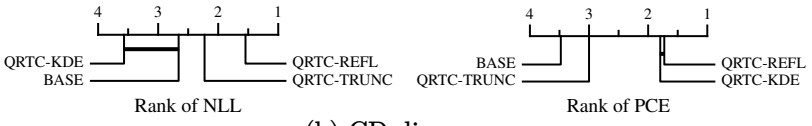


Figure D.23: Comparison of different methods to estimate the calibration map. In this example, 512 PITs have been sampled from a beta distribution $Z \sim \text{Beta}(0.2, 0.2)$ and the calibration map is estimated using Φ_{KDE} with $b = 0.1$ (Equation (4.1) in the main text).

Figure D.24 compares `QRTC` where the calibration map of the post-hoc model has been estimated using either Φ_{KDE}, Φ_{TRUNC} and Φ_{REFL}. We denote these methods `QRTC-KDE`, `QRTC-TRUNC` and `QRTC-REFL` respectively. It is worth noting that `QRTC-REFL` corresponds to the method `QRTC` in the main text. In terms of NLL, `QRTC-REFL` performs significantly better in terms of NLL and `QRTC-KDE` is the least effective. In terms of PCE, `QRTC-REFL` and `QRTC-KDE` perform similarly and `QRTC-TRUNC` is the least effective. This confirms that the method of Reflected Kernel should be preferred.



(a) Boxplots of Cohen's d of different metrics on all datasets, w.r.t. `BASE`.



(b) CD diagrams

Figure D.24: Comparison between different kernel density estimation approaches. Note that the metrics CRPS and SD are not provided because they are ill-defined for `QRTC-KDE`. More precisely, since the quantile function $(\Phi_{\text{KDE}})^{-1}$ returns values outside the interval $[0, 1]$, we can not correctly sample from the model.

APPENDIX E

Supplementary Material for Chapter 5

E.1. Related Work

Our research builds on a broad body of literature that spans several closely related themes. This supplementary section provides a concise overview of these topics.

In the realm of multivariate functional data, Diquigiovanni et al. (2021a) introduced conformal predictors that create adaptive, finite-sample valid prediction bands, with extensions into time series applications, particularly in energy markets (Diquigiovanni et al., 2024). In image processing, recent applications (Horwitz and Hoshen, 2022; Teneggi et al., 2023) apply CP in a pixel-wise manner, resulting in hyperrectangular regions that may not capture pixel dependencies effectively.

For multi-step-ahead or multi-horizon forecasting, predictions can be made across multiple outputs simultaneously rather than sequentially, aligning with a multi-output forecasting framework. Stankeviciute et al. (2021) explored multi-horizon time series forecasting using recurrent neural networks (RNNs), incorporating univariate conformal techniques with nominal coverage adjustments via Bonferroni correction. Similarly, English et al. (2024) adapted the Amplitude-Modulated L-inf norm method from Diquigiovanni et al. (2021b) for multi-output, multi-step forecasting.

In multi-target regression, Messoudi et al. (2021) applied copula functions in deep neural networks to provide multivariate predictions with guaranteed coverage. Their findings suggest that simple parametric copulas can work for certain datasets, but more complex copulas may be required for well-calibrated predictions, which introduces challenges, as complex copulas typically require significant calibration data. Building on this, S. H. Sun and Yu (2024) proposed a copula-based method for multi-step time series forecasting, optimizing the calibration and efficiency of confidence intervals. However, this method requires two calibration phases and is primarily feasible with large calibration datasets. Moreover, its coverage relies on the empirical copula, limiting applicability to other learnable copula classes. One very recent advancement on the subject, following ideas expressed by Messoudi et al. (2021) in their conclusions, is Park et al. (2024), where the dependence structure between marginal distributions is recovered via the use

of vine copulas.

Another set of methodologies that tackle multi-output problems are based on multiplicity-correction approaches for multiple testing. Timans et al. (2025) improve over Bonferroni correction using permutation tests, and obtain tighter and globally valid prediction. Methods based on multiplicity correction such as controlling the Family-Wise Error Rate (FWER) are valuable for providing error control guarantees across multiple outputs. In contrast, the methods we survey and propose aim for potentially tighter prediction sets by directly modeling the multivariate structure.

In the context of conformal prediction, the flexibility in configuring the prediction set is a degree of freedom for the modeler. To overcome the limitations of traditional hyper-rectangular prediction sets, Messoudi et al. (2022) introduced ellipsoidal uncertainty sets that enable instance-specific adaptation of confidence regions. Johnstone and Ndiaye (2025) advanced multi-output regression by developing efficient techniques for approximating conformal prediction sets without retraining the model, although their approach relies heavily on the predictive model being a linear function of Y. S. H. Sun and Yu (2024) constructed ellipsoidal prediction sets for time series, capable of modeling dependencies between outputs, though this method does not handle multimodality. Our work closely connects with the multivariate conformal prediction literature, where multi-horizon prediction is viewed as a prediction across multiple outputs.

Overall, as this study demonstrates, the flexibility of conformal prediction allows for coherent handling of diverse data types. Multi-output problems represent one facet of a broader taxonomy, as explored by X. Zhou et al. (2025), who discuss further developments in multi-output conformal prediction.

E.2. Additional multi-output conformal methods

In this section, we describe the prediction sets $\hat{R}$ for `M-CP` and `CopulaCPTS`, which both produce hyperrectangular regions.

M-CP. Y. Zhou et al. (2024) applied a univariate conformal method to each output $i \in [d]$ of the multivariate response. Specifically, given a conformity score s_i for the i-th dimension, joint coverage across all dimensions can be achieved using the following conformity score:

$$s_{\text{M-CP}}(x, y) = \max_{i \in [d]} s_i(x, y_i). \tag{E.1}$$

A similar score has been considered by Diquigiovanni et al. (2021b) in the context of functional regression.

In this work, we use conformalized quantile regression (CQR, Romano et al., 2019) for each output $i \in [d]$, where the conformity score is given by:

$$s_i(x, y_i) = \max\{\hat{l}_i(x) - y_i, y_i - \hat{u}_i(x)\}, \tag{E.2}$$

with $\hat{l}_i(x)$ and $\hat{u}_i(x)$ representing the conditional quantiles of $Y_i|X = x$ at levels α_l and α_u, respectively. In our experiments, we consider equal-tailed prediction intervals, where $\alpha_l = \frac{\alpha}{2}$, $\alpha_u = 1 - \frac{\alpha}{2}$, and α denotes the miscoverage level. The corresponding prediction set $\hat{R}_{\text{M-CP}}(x) = \times_{i=1}^{d} [\hat{l}_i(x) - \hat{q}, \hat{u}_i(x) + \hat{q}]$ is a hyperrectangle.

CopulaCPTS. CopulaCPTS (S. H. Sun and Yu, 2024) is originally designed for time-series but the calibration procedure is valid for any multi-dimensional variable. It models the joint probability of uncertainty for each output with a copula. The calibration set is divided into two subsets $\mathcal{D}_{\text{cal}-1}$ and $\mathcal{D}_{\text{cal}-2}$. $\mathcal{D}_{\text{cal}-1}$ serves for the estimation of a CDF on the conformity score of each output and $\mathcal{D}_{\text{cal}-2}$ allows to calibrate the copula. CopulaCPTS can combine any univariate or multivariate conformity scores. In this paper, we use the CQR score s_i (E.2) for each dimension $i \in [d]$.

Denote $\hat{F}_i$ the empirical CDF of the conformity scores $\{s_i(x, y_i)\}_{(x,y)\in\mathcal{D}_{\text{cal}-1}}$ for $i \in [d]$, and $\hat{F}_i^{-1}$ the corresponding empirical quantile function. In practice, to minimize set sizes while achieving marginal coverage, CopulaCPTS computes the optimal $s_1^*, \ldots, s_d^*$ that minimize the following loss using stochastic gradient descent:

$$\mathcal{L}(\hat{s}_1, \ldots, \hat{s}_d) = \frac{1}{|\mathcal{D}_{\text{cal}-2}|} \sum_{(x,y)\in\mathcal{D}_{\text{cal}-2}} \prod_{i=1}^{d} \mathbb{1}\left(\hat{F}_i(s_i(x, y_i)) < \hat{F}_i^{-1}(\hat{s}_i)\right) - (1-\alpha). \tag{E.3}$$

Then, the prediction set is defined as:

$$\hat{R}_{\text{CopulaCPTS}}(x) = \{y \in \mathcal{Y} : \forall i \in [d], s_i(x, y_i) < s_i^*\} \tag{E.4}$$

S. H. Sun and Yu (2024) proved that this prediction set achieves marginal coverage. However, since CopulaCPTS does not follow the SCP algorithm, it does not achieve properties on the marginal coverage from Section E.5.1.

E.3. Relationship between conformity scores and regions

Section 5.2 and Section 5.4 in the main text presented several multi-output conformal methods through their conformity scores s. As explained in Section 2.5.1, their corresponding prediction set $\hat{R}$ can be computed as follows:

$$\hat{R}(x) = \{y \in \mathcal{Y} : s(x, y) \leqslant \hat{q}\}.$$

In this section, we explicitly derive the prediction set associated with these methods.

M-CP. Following Y. Zhou et al. (2024), the prediction set $\hat{R}_{\text{M-CP}}$ can be derived from $s_{\text{M-CP}}$ as follows:

$$
\begin{aligned}
s_{\text{M-CP}}(x, y) \leqslant \hat{q} &\iff \max_{i \in [d]} s_i(x, y_i) \leqslant \hat{q} && \text{(E.5)}\\
&\iff \forall i \in [d], s_i(x, y_i) \leqslant \hat{q} && \text{(E.6)}\\
&\iff \forall i \in [d], \max\{\hat{l}_i(x) - y_i, y_i - \hat{u}_i(x)\} \leqslant \hat{q} && \text{(E.7)}\\
&\iff \forall i \in [d], \hat{l}_i(x) - y_i \leqslant \hat{q} \wedge y_i - \hat{u}_i(x) \leqslant \hat{q} && \text{(E.8)}\\
&\iff \forall i \in [d], \hat{l}_i(x) - \hat{q} \leqslant y_i \wedge y_i \leqslant \hat{u}_i(x) + \hat{q} && \text{(E.9)}\\
&\iff \forall i \in [d], y_i \in [\hat{l}_i(x) - \hat{q}, \hat{u}_i(x) + \hat{q}] && \text{(E.10)}\\
&\iff y \in \bigtimes_{i=1}^{d} [\hat{l}_i(x) - \hat{q}, \hat{u}_i(x) + \hat{q}] && \text{(E.11)}\\
&\iff y \in \hat{R}_{\text{M-CP}}(x). && \text{(E.12)}
\end{aligned}
$$

DR-CP The equivalence is straightforward.

C-HDR. Given $\hat{Y} \sim \hat{P}_{Y|X=x}$ and $U = \hat{f}_{Y|X=x}(\hat{Y})$, for any $y \in \mathcal{Y}$, we can write

$$
\begin{aligned}
& s_{\text{C-HDR}}(x, y) && \text{(E.13)}\\
&= \text{HPD}_{\hat{f}_{Y|X=x}}(y) && \text{(E.14)}\\
&= \mathbb{P}(\hat{f}_{Y|X=x}(\hat{Y}) \geqslant \hat{f}_{Y|X=x}(y) \mid X = x) && \text{(E.15)}\\
&= \mathbb{P}(U \geqslant \hat{f}_{Y|X=x}(y) \mid X = x) && \text{(E.16)}\\
&= 1 - \mathbb{P}(U \leqslant \hat{f}_{Y|X=x}(y) \mid X = x) && \text{(E.17)}\\
&= 1 - F_{U|X=x}(\hat{f}_{Y|X=x}(y)), && \text{(E.18)}
\end{aligned}
$$

where $F_{U|X=x}$ is the conditional CDF of U given $X = x$.

Recall that the prediction set for `C-HDR` is given by

$$
\hat{R}_{\text{C-HDR}}(x) = \{y \in \mathcal{Y} : \hat{f}_{Y|X=x}(y) \geqslant t_{\hat{q}}\}, \quad \text{where } t_{\hat{q}} = \sup\{t : \mathbb{P}(\hat{f}_{Y|X=x}(\hat{Y}) \geqslant t \mid X = x) \geqslant \hat{q}\}. \tag{E.19}
$$

The threshold $t_{\hat{q}}$ in (E.19) can be equivalently written as follows:

$$
\begin{aligned}
t_{\hat{q}} &= \sup\{t : \mathbb{P}(\hat{f}_{Y|X=x}(\hat{Y}) \geqslant t \mid X = x) \geqslant \hat{q}\} && \text{(E.20)}\\
&= \sup\{t : \mathbb{P}(U \geqslant t \mid X = x) \geqslant \hat{q}\} && \text{(E.21)}\\
&= \sup\{t : 1 - \mathbb{P}(U \leqslant t \mid X = x) \geqslant \hat{q}\} && \text{(E.22)}\\
&= \sup\{t : 1 - \hat{q} \geqslant F_{U|X=x}(t)\} && \text{(E.23)}\\
&= F^{-1}_{U|X=x}(1 - \hat{q}), && \text{(E.24)}
\end{aligned}
$$

where we use the definition of the upper quantile function in the last step.

Using (E.13), (E.19), and (E.24), we can write

$$
\begin{aligned}
s_{\text{C-HDR}}(x, y) \leqslant \hat{q} &\iff \text{HPD}_{\hat{f}_{Y|X=x}}(y) \leqslant \hat{q} && \text{(E.25)}\\
&\iff 1 - F_{U|X=x}(\hat{f}_{Y|X=x}(y)) \leqslant \hat{q} && \text{(E.26)}\\
&\iff F_{U|X=x}(\hat{f}_{Y|X=x}(y)) \geqslant 1 - \hat{q} && \text{(E.27)}\\
&\iff \hat{f}_{Y|X=x}(y) \geqslant F^{-1}_{U|X=x}(1 - \hat{q}) && \text{(E.28)}\\
&\iff \hat{f}_{Y|X=x}(y) \geqslant t_{\hat{q}} && \text{(E.29)}\\
&\iff y \in \hat{R}_{\text{C-HDR}}(x). && \text{(E.30)}
\end{aligned}
$$

PCP. Let $B(\mu, r)$ represent a ball with center μ and radius r. Following Z. Wang et al. (2023), we show that, for any $x \in \mathcal{X}$, $\hat{R}_{\text{PCP}}(x)$ corresponds to a union of balls:

$$
\begin{aligned}
s_{\text{PCP}}(x, y) \leqslant \hat{q} &\iff \min_{l \in [L]} \|y - \tilde{Y}^{(l)}\| \leqslant \hat{q} && \text{(E.31)}\\
&\iff \exists l \in [L], \|y - \tilde{Y}^{(l)}\| \leqslant \hat{q} && \text{(E.32)}\\
&\iff \exists l \in [L], y \in B(\tilde{Y}^{(l)}, \hat{q}) && \text{(E.33)}\\
&\iff y \in \bigcup_{l \in [L]} B(\tilde{Y}^{(l)}, \hat{q}) && \text{(E.34)}\\
&\iff y \in \hat{R}_{\text{PCP}}(x), && \text{(E.35)}
\end{aligned}
$$

where $\tilde{Y}^{(l)} \sim \hat{P}_{Y|X=x}, l \in [L]$.

HD-PCP. For `HD-PCP`, the reasoning is similar to `PCP` with the difference that only the $\lfloor (1 - \alpha)L \rfloor$ samples with the highest density are kept.

CDF-based conformity scores. We note that the region $\hat{R}_{\text{CDF}}(x)$ has a similar form to $\hat{R}_W(x) = \{y \in \mathcal{Y} : s_W(x, y) \leqslant \hat{q}\}$, except that the threshold on $s_W(x, y)$ is different and depends on x. In fact, we can write

$$
\begin{aligned}
\hat{R}_{\text{CDF}}(x) &= \{y \in \mathcal{Y} : s_{\text{CDF}}(x, y) \leqslant \hat{q}\} && \text{(E.36)}\\
&= \{y \in \mathcal{Y} : F_{W|X=x}(s_W(x, y)) \leqslant \hat{q}\} && \text{(E.37)}\\
&= \{y \in \mathcal{Y} : s_W(x, y) \leqslant F^{-1}_{W|X=x}(\hat{q})\}. && \text{(E.38)}
\end{aligned}
$$

In the special case where $s_W = s_{\text{PCP}}$, since `PCP` always generates predictions as a union of balls, we can conclude that `C-PCP` will do the same.

Latent-based conformity scores. Since $\hat{T}(\cdot; x)$ is bijective, for every $y \in \mathcal{Y}$, there exists a unique $z \in \mathcal{Z}$ such that $y = \hat{T}(z; x)$. Therefore, the condition $\rho_{\mathcal{Z}}(\hat{T}^{-1}(y; x)) \leqslant \hat{q}$ is equivalent to $\rho_{\mathcal{Z}}(z) \leqslant \hat{q}$, where $z = \hat{T}^{-1}(y; x)$. This gives the prediction set:

$$
\begin{aligned}
\hat{R}_{\text{L-CP}}(x) &= \{y \in \mathcal{Y} : \rho_{\mathcal{Z}}(\hat{T}^{-1}(y; x)) \leqslant \hat{q}\} && \text{(E.39)}\\
&= \{\hat{T}(z; x) : z \in \mathcal{Z} \text{ and } \rho_{\mathcal{Z}}(z) \leqslant \hat{q}\}. && \text{(E.40)}
\end{aligned}
$$

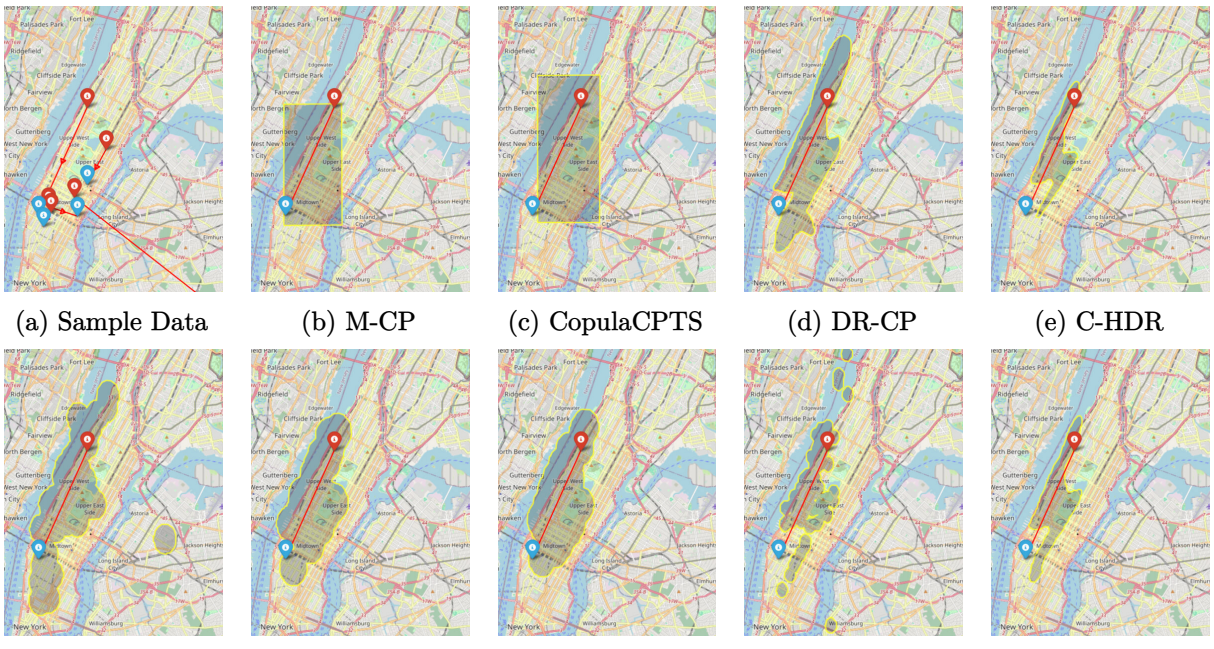
E.4. Additional illustrative examples

E.4.1 A real-world application

Following J. Wang et al. (2023), we apply the multi-output conformal methods to the taxi dataset, where the goal is to predict the drop-off location of a New York taxi passenger based on the passenger's information.

Figures E.1a and E.2a display five randomly selected samples from the dataset, showing the pick-up (red pin) and drop-off (blue pin) locations of taxi passengers. The remaining panels show a specific input-output pair (x, y) and the corresponding prediction sets generated by the conformal methods discussed in this paper. The coverage level $1 - \alpha$ for these regions is set to 0.8, with MQF^2 as the base predictor, as introduced in Section E.6.2.

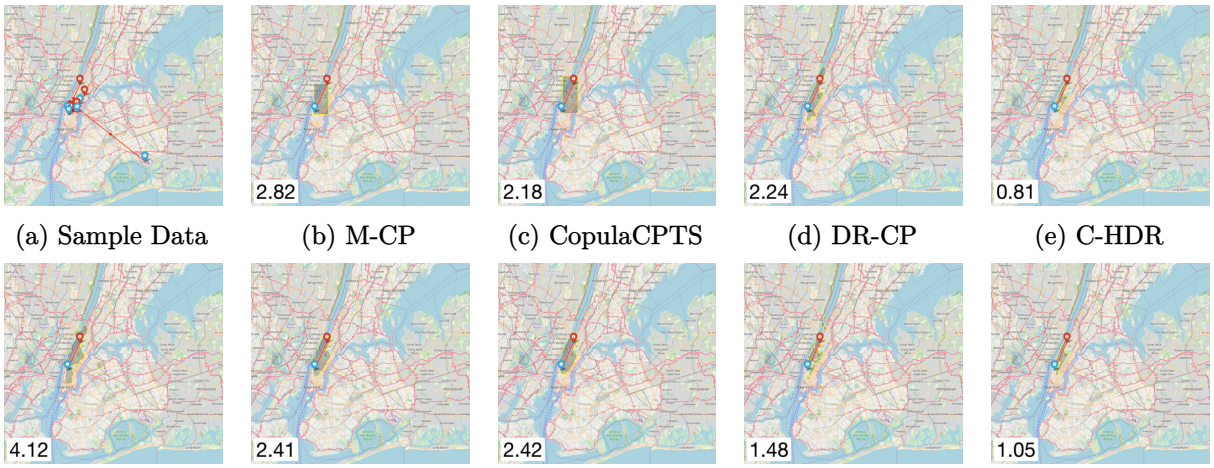
Figure E.2 corresponds to the same data and prediction sets as Figure E.1 except that it is zoomed out for better comparison with Figure E.3. Each region is labeled with its size, calculated using the estimator from Section E.6.4, displayed in the bottom left corner. Notably, C-PCP generates regions similar in shape to PCP but with an input-adaptive radius, resulting in smaller set sizes (8.2 compared to 8.67) in this case. Additionally, HD-PCP produces more clustered regions, while PCP and C-PCP show more dispersed regions.



(a) Sample Data (b) M-CP (c) CopulaCPTS (d) DR-CP (e) C-HDR

(f) PCP (g) HD-PCP (h) STDQR (i) C-PCP (j) L-CP

Figure E.1: Conformal methods applied on the NYC Taxi dataset for an input with low uncertainty.

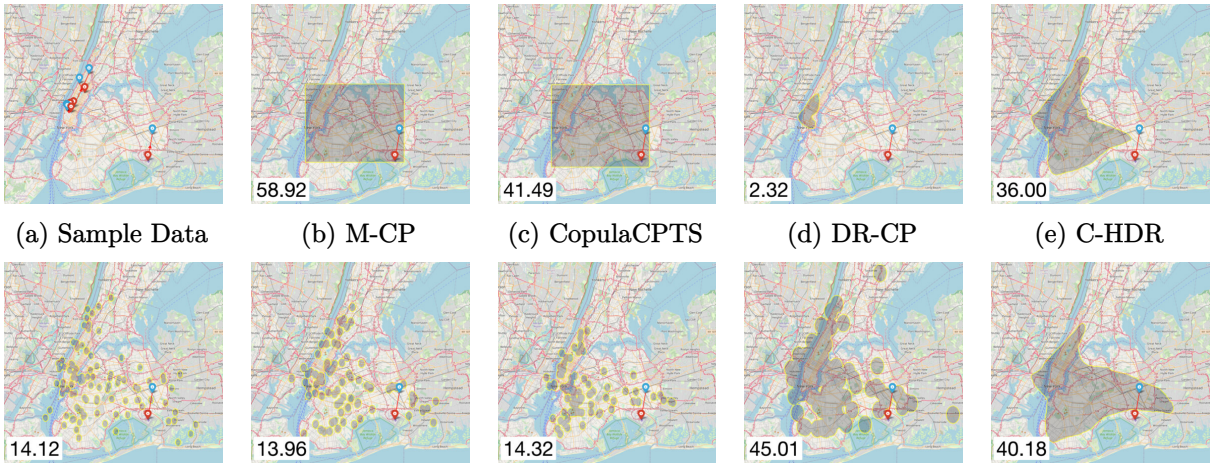


(a) Sample Data (b) M-CP (c) CopulaCPTS (d) DR-CP (e) C-HDR

(f) PCP (g) HD-PCP (h) STDQR (i) C-PCP (j) L-CP

Figure E.2: Zoomed out version of Figure E.1.

Figure E.3 presents the same example for an input-output pair where the input is associated with higher uncertainty, resulting in larger set sizes. As in the first figure, the shapes of the regions (e.g., unions of hyperrectangles, quantile regions, etc.) remain consistent but expand to cover a larger area. Conformal methods with the best set sizes differ between the two figures, with C-HDR producing the smallest region in the first figure and DR-CP in the second. In this case, C-PCP selects a larger radius than PCP, resulting in larger regions than PCP. The observation that PCP and C-PCP produce more dispersed regions, while HD-PCP generates more clustered regions, also holds true for this higher uncertainty case.



(a) Sample Data (b) M-CP (c) CopulaCPTS (d) DR-CP (e) C-HDR

(f) PCP (g) HD-PCP (h) STDQR (i) C-PCP (j) L-CP

Figure E.3: Conformal methods applied on the NYC Taxi dataset for an input with high uncertainty.

E.4.2 Toy examples

We define two data-generating processes to evaluate the models compared to a known distribution: a unimodal heteroscedastic distribution and a bimodal heteroscedastic distribution. The input variable $X \in \mathbb{R}$ is univariate ($p = 1$) and the output variable $Y \in \mathbb{R}^2$ is bidimensional ($d = 2$). The variables X and Y are scaled linearly such that the mean and variances on each dimension are 0 and 1. The figures are inspired by Barrio et al. (2024).

Unimodal heteroscedastic process. The first process is illustrated in Figure 5.2 in the main text. The data generating process is as follows:

$$X \sim \mathcal{U}(0, 1), \tag{E.41}$$

$$Y \mid X = x \sim \frac{1}{k} \sum_{j=1}^{k} \mathcal{N}\big((1.3 - x)\boldsymbol{\mu}^{(j)}(x),\, \sigma^2 I_2\big), \tag{E.42}$$

$$\tag{E.43}$$

where $k = 200$, $\sigma = 0.2$, I_2 is the 2×2 identity, and, for $j = 1, \ldots, k$,

$$\boldsymbol{\mu}_1^{(j)} = \cos \alpha_j \tag{E.44}$$

$$\boldsymbol{\mu}_2^{(j)} = (0.5 - \sin \alpha_j) \tag{E.45}$$

$$\alpha_j = \frac{(j-1)\,\pi}{k-1} \tag{E.46}$$

Detailed metrics for this dataset, supporting Section 5.5.1, are provided in Table E.1.

Table E.1: Detailed metrics for the unimodal heteroscedastic process from Figure 5.2.

Method	MC	Median Size	CEC-X ($\times 100$)	CEC-Z ($\times 100$)	WSC	Test time
M-CP	$\mathbf{0.805_{0.0039}}$	$8.47_{0.14}$	$0.110_{0.023}$	$0.0812_{0.018}$	$\mathbf{0.803_{0.011}}$	$\mathbf{0.0336_{0.022}}$
CopulaCPTS	$\mathbf{0.815_{0.011}}$	$8.78_{0.25}$	$0.191_{0.049}$	$0.163_{0.047}$	$\mathbf{0.814_{0.015}}$	$1.04_{0.021}$
DR-CP	$\mathbf{0.808_{0.0042}}$	$7.03_{0.071}$	$0.613_{0.045}$	$0.560_{0.042}$	$0.710_{0.016}$	$\mathbf{0.0209_{0.00047}}$
C-HDR	$\mathbf{0.810_{0.0038}}$	$\mathbf{6.80_{0.059}}$	$0.0637_{0.016}$	$0.0825_{0.012}$	$\mathbf{0.798_{0.0070}}$	$3.52_{0.085}$
PCP	$\mathbf{0.805_{0.0039}}$	$9.16_{0.089}$	$0.668_{0.052}$	$0.587_{0.046}$	$0.713_{0.0080}$	$1.69_{0.021}$
HD-PCP	$\mathbf{0.804_{0.0037}}$	$7.44_{0.056}$	$0.287_{0.031}$	$0.256_{0.034}$	$0.758_{0.013}$	$3.38_{0.043}$
STDQR	$\mathbf{0.806_{0.0027}}$	$7.87_{0.070}$	$0.343_{0.025}$	$0.305_{0.027}$	$0.746_{0.011}$	$1.77_{0.022}$
C-PCP	$\mathbf{0.808_{0.0049}}$	$9.14_{0.12}$	$\mathbf{0.0464_{0.013}}$	$\mathbf{0.0484_{0.013}}$	$0.822_{0.0085}$	$3.44_{0.056}$
L-CP	$\mathbf{0.803_{0.0039}}$	$8.24_{0.11}$	$\mathbf{0.0544_{0.0073}}$	$0.0654_{0.012}$	$\mathbf{0.811_{0.011}}$	$0.0217_{0.00052}$

Bimodal heteroscedastic process. Figure E.4, similar to Figure 5.2 but with a bimodal distribution for the output, is introduced in Section 5.5.1.

The data generating process is as follows:

$$X \sim \mathcal{U}(0.5, 2), \tag{E.47}$$

$$Y \mid X = x \sim 0.5 \cdot \mathcal{N}(4, xI_d) + 0.5 \cdot \mathcal{N}(-4, I_d/x). \tag{E.48}$$

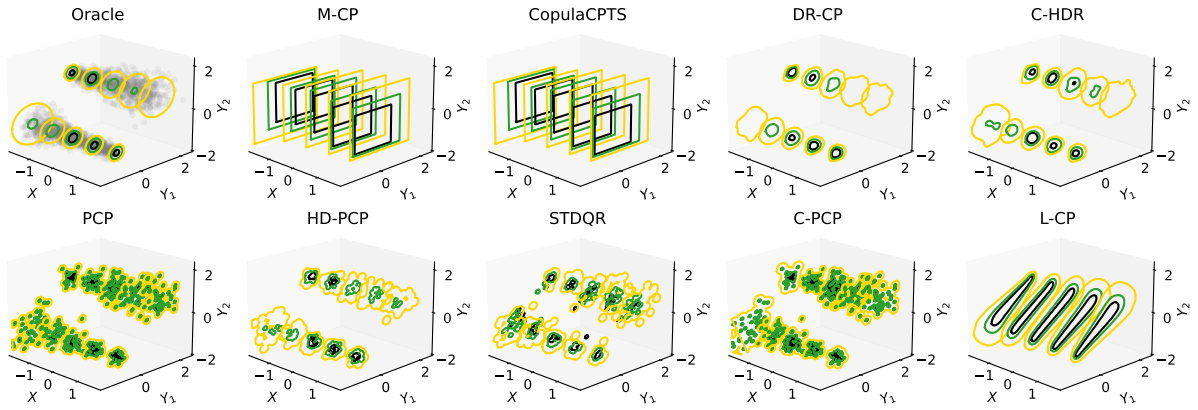


Figure E.4: Examples of prediction sets on a bivariate bimodal dataset, conditional on a univariate input.

E.5. Proofs

E.5.1 Distribution of the marginal coverage conditional on calibration data

In contrast to `M-CP`, `L-CP`, and `DR-CP`, the methods `C-HDR`, `PCP`, `HD-PCP`, and `C-PCP` rely on a non-deterministic conformity score. For each calibration and test point, `C-HDR`, `PCP`, `HD-PCP`, and `C-PCP` require sampling K, L, L, and $L + K$ points, respectively.

Let $\mathcal{D}_{\text{cal}} = \{(X^{(j)}, Y^{(j)})\}_{j \in [|\mathcal{D}_{\text{cal}}|]}$ represent the calibration dataset and (X, Y) be the test instance. Let $\hat{\boldsymbol{Y}}_{\text{cal}} = \{\hat{\boldsymbol{Y}}_{\text{cal}}^{(j)}\}_{j \in [|\mathcal{D}_{\text{cal}}|]}$ represent samples from the calibration dataset where $\hat{\boldsymbol{Y}}_{\text{cal}}^{(j)}$ is generated based on input $X^{(j)}$ and $\hat{\boldsymbol{Y}}_{\text{test}}$ the samples generated based on input X. Despite the added sampling uncertainty, these methods still provide a marginal coverage guarantee:

$$\mathbb{P}_{X,Y,\hat{\boldsymbol{Y}}_{\text{test}},\mathcal{D}_{\text{cal}},\hat{\boldsymbol{Y}}_{\text{cal}}}(Y \in \hat{R}(X)) \geqslant 1 - \alpha. \tag{E.49}$$

Compared to (2.55), the probability is additionally on $\hat{\boldsymbol{Y}}_{\text{cal}}$ and $\hat{\boldsymbol{Y}}_{\text{test}}$. This result, specifically for `PCP` and `HD-PCP`, was demonstrated by Z. Wang et al. (2023).

In Lemma 7, we further show that the marginal coverage conditional on the calibration dataset $\mathcal{D}_{\text{cal}}$ and the samples $\hat{\boldsymbol{Y}}_{\text{cal}}$ follows a beta distribution, using standard arguments. Assuming no ties among the scores, this lemma applies to any conformity score s.

Lemma 7. Assuming no ties among the scores and i.i.d. inputs, outputs and samples, the distribution of the coverage, conditional on the calibration dataset and its samples, is given by:

$$\mathbb{P}(Y \in \hat{R}(X) \mid \mathcal{D}_{\text{cal}}, \hat{\boldsymbol{Y}}_{\text{cal}}) \sim \text{Beta}(k_\alpha, |\mathcal{D}_{\text{cal}}| + 1 - k_\alpha), \tag{E.50}$$

where $k_\alpha = \lceil (1 - \alpha)(|\mathcal{D}_{\text{cal}}| + 1) \rceil$. Moreover, $\mathbb{P}(Y \in \hat{R}(X)) = \frac{k_\alpha}{|\mathcal{D}_{\text{cal}}|+1}$, which implies

$$1 - \alpha \leqslant \mathbb{P}(Y \in \hat{R}(X)) < 1 - \alpha + \frac{1}{|\mathcal{D}_{\text{cal}}| + 1}.$$

Proof. For the methods `C-HDR`, `PCP`, `HD-PCP`, and `C-PCP`, the conformity score s is non-deterministic due to sampling uncertainty. To clarify, we define a deterministic conformity score $\bar{s} : \mathcal{X} \times \mathcal{Y} \times \mathbb{S}$, where $\mathbb{S}$ represents the space of samples for a given method.

For $j = 1, \ldots, |\mathcal{D}_{\text{cal}}|$, let $S_j = \bar{s}(X^{(j)}, Y^{(j)}, \hat{\boldsymbol{Y}}_{\text{cal}}^{(j)})$ denote the conformity score on the calibration dataset, and let $S = \bar{s}(X, Y, \hat{\boldsymbol{Y}}_{\text{test}})$ represent the conformity score for the test instance. Since $\bar{s}$ is deterministic and the tuples $(X^{(1)}, Y^{(1)}, \hat{\boldsymbol{Y}}_{\text{cal}}^{(1)}), \ldots, (X^{(|\mathcal{D}_{\text{cal}}|)}, Y^{(|\mathcal{D}_{\text{cal}}|)}, \hat{\boldsymbol{Y}}_{\text{cal}}^{(|\mathcal{D}_{\text{cal}}|)}), (X, Y, \hat{\boldsymbol{Y}}_{\text{test}})$ are i.i.d. random variables, $S_1, \ldots, S_{|\mathcal{D}_{\text{cal}}|}, S$ are also i.i.d. random variables.

Since $S_1, \ldots, S_{|\mathcal{D}_{\text{cal}}|}, S$ are identically distributed, they share the same CDF. Using the probability integral transform, $F_S(S) \sim \mathcal{U}(0, 1)$. Thus, $F_S(S_1), \ldots, F_S(S_{|\mathcal{D}_{\text{cal}}|})$ correspond to uniform variates $U_1, \ldots, U_{|\mathcal{D}_{\text{cal}}|}$. Since there are no ties among the scores, F_S is strictly increasing, and $F_S(S_{(j)}) = U_{(j)}$ for $j = 1, \ldots, |\mathcal{D}_{\text{cal}}|$, where $S_{(j)}$ and $U_{(j)}$ are the j-th order statistics. Hence:

$$\begin{aligned}
\mathbb{P}(Y \in \hat{R}(X) \mid \mathcal{D}_{\text{cal}}, \hat{\boldsymbol{Y}}_{\text{cal}}) &= \mathbb{P}(S \leqslant S_{(k_\alpha)} \mid S_1, \ldots, S_{|\mathcal{D}_{\text{cal}}|}) && \text{(E.51)} \\
&= F_S(S_{(k_\alpha)}) && \text{(E.52)} \\
&= U_{(k_\alpha)} && \text{(E.53)} \\
&\sim \text{Beta}(k_\alpha, |\mathcal{D}_{\text{cal}}| + 1 - k_\alpha). && \text{(E.54)}
\end{aligned}$$

The final step results from the distribution of uniform order statistics. Taking the expectation of the Beta distribution gives:

$$\mathbb{P}(Y \in \hat{R}(X)) = \mathbb{E}\Big[\mathbb{P}(Y \in \hat{R}(X) \mid \mathcal{D}_{\text{cal}}, \hat{\boldsymbol{Y}}_{\text{cal}})\Big] = \frac{k_\alpha}{|\mathcal{D}_{\text{cal}}| + 1}, \tag{E.55}$$

which implies

$$1 - \alpha \leqslant \mathbb{P}(Y \in \hat{R}(X)) < 1 - \alpha + \frac{1}{|\mathcal{D}_{\text{cal}}| + 1}.$$

□

E.5.2 Proofs of asymptotic conditional coverage

L-CP

Proposition 7. Assuming $|\mathcal{D}_{\text{cal}}| \to \infty$ and $\hat{T}(Z; X) \overset{\text{d.}}{=} Y|X$, `L-CP` achieves conditional coverage.

Proof. We first show that the conditional coverage of `L-CP` is equal to the CDF of the random

variable $\rho_{\mathcal{Z}}(Z)$ in $\hat{q}$, i.e., $F_{\rho_{\mathcal{Z}}(Z)}(\hat{q})$. Given $x \in \mathcal{X}$, we have:

$$\begin{aligned}
&\mathbb{P}(Y \in \hat{R}_{\text{L-CP}}(X) \mid X = x) && && \text{(E.56)}\\
=&\mathbb{P}(Y \in \{\hat{T}(z; x) : z \in R_{\mathcal{Z}}(\hat{q})\} \mid X = x) && && \text{(E.57)}\\
=&\mathbb{P}(\hat{T}^{-1}(Y; X) \in R_{\mathcal{Z}}(\hat{q}) \mid X = x) && \text{(Invertibility of } \hat{T}(\cdot; X)) && \text{(E.58)}\\
=&\mathbb{P}(Z \in R_{\mathcal{Z}}(\hat{q})) && (\hat{T}(Z; X) \overset{\text{d.}}{=} Y|X) && \text{(E.59)}\\
=&\mathbb{P}(\rho_{\mathcal{Z}}(Z) \leqslant \hat{q}) && && \text{(E.60)}\\
=&F_{\rho_{\mathcal{Z}}(Z)}(\hat{q}). && && \text{(E.61)}
\end{aligned}$$

Marginalizing over X, we obtain that the marginal coverage is also equal to $F_{\rho_{\mathcal{Z}}(Z)}(\hat{q})$:

$$\begin{aligned}
&\mathbb{P}(Y \in \hat{R}_{\text{L-CP}}(X)) && \text{(E.62)}\\
=&\mathbb{E}_X\Big[\mathbb{P}(Y \in \hat{R}_{\text{L-CP}}(X) \mid X)\Big] && \text{(E.63)}\\
=&\mathbb{E}_X\big[F_{\rho_{\mathcal{Z}}(Z)}(\hat{q})\big] && \text{(E.64)}\\
=&F_{\rho_{\mathcal{Z}}(Z)}(\hat{q}) && \text{(E.65)}
\end{aligned}$$

In the limit of $|\mathcal{D}_{\text{cal}}| \to \infty$, thanks to the Glivenko-Cantelli theorem, $\mathbb{P}(Y \in \hat{R}_{\text{L-CP}}(X)) = 1 - \alpha$ and the quantile $\hat{q}$ obtained by SCP is thus $F^{-1}_{\rho_{\mathcal{Z}}(Z)}(1 - \alpha)$.

Finally, we obtain that the conditional coverage is equal to $1 - \alpha$:

$$\begin{aligned}
&\mathbb{P}(Y \in \hat{R}_{\text{L-CP}}(X) \mid X = x) && \text{(E.66)}\\
=&F_{\rho_{\mathcal{Z}}(Z)}(F^{-1}_{\rho_{\mathcal{Z}}(Z)}(1 - \alpha)) && \text{(E.67)}\\
=&1 - \alpha. && \text{(E.68)}
\end{aligned}$$

□

C-HDR and C-PCP

Lemma 8. Assuming $|\mathcal{D}_{\text{cal}}| \to \infty$, any conformal method with conformity score s_{CDF} (5.9) achieves conditional coverage, independently from the conformity score s_W of the base method. With the additional assumption that $K \to \infty$ and $\hat{f} = f$, s_{ECDF} (5.10) achieves conditional coverage.

Proof. Let $W = s_W(X, Y)$ and consider $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. By the probability integral transform, $s_{\text{CDF}}(x, Y) = F_{W|X=x}(W \mid X = x) \sim \mathcal{U}(0, 1)$.

Marginalizing over X, we obtain:

$$\begin{aligned}
\mathbb{P}(Y \in \hat{R}_{\text{CDF}}(X)) &= \mathbb{P}(s_{\text{CDF}}(X, Y) \leqslant \hat{q}) && \text{(E.69)}\\
&= \mathbb{E}_X[\mathbb{P}(s_{\text{CDF}}(X, Y) \leqslant \hat{q} \mid X)] && \text{(E.70)}\\
&= \mathbb{E}_X[\mathbb{P}(U \leqslant \hat{q})] && \text{(E.71)}\\
&= \mathbb{E}_X[\hat{q}] && \text{(E.72)}\\
&= \hat{q}, && \text{(E.73)}
\end{aligned}$$

where $U \sim \mathcal{U}(0,1)$. In the limit of $|\mathcal{D}_{\text{cal}}| \to \infty$, thanks to the Glivenko-Cantelli theorem, $\mathbb{P}(Y \in \hat{R}_{\text{CDF}}(X)) = 1 - \alpha$ and the quantile $\hat{q}$ obtained by SCP is thus $1 - \alpha$.

Finally, we note that:

$$\begin{aligned}
\mathbb{P}(Y \in \hat{R}_{\text{CDF}}(X) \mid X = x) &= \mathbb{P}(s_{\text{CDF}}(X, Y) \leqslant \hat{q} \mid X = x) && \text{(E.74)} \\
&= \mathbb{P}(U \leqslant 1 - \alpha) && \text{(E.75)} \\
&= 1 - \alpha. && \text{(E.76)}
\end{aligned}$$

Assuming $\hat{f} = f$, observe that, for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, $s_{\text{ECDF}}(x, y) \to s_{\text{CDF}}(x, y)$ as $K \to \infty$ by the law of large numbers. Thus, under these conditions, any conformal method with conformity score s_{ECDF} achieves conditional coverage.

□

Proposition 8. Assuming $|\mathcal{D}_{\text{cal}}| \to \infty$ and $K \to \infty$, both `C-HDR` and `C-PCP` with the oracle base predictor $\hat{f} = f$ achieve conditional coverage.

Proof. The proof is direct by Lemma 8 with $s_W(x, y) = s_{\text{DR-CP}}(x, y)$ for C-HDR and $s_W(x, y) = s_{\text{PCP}}(x, y)$ for C-PCP. □

M-CP

Consider `M-CP` with exact quantile estimates $\hat{l}_i(x) = Q_{Y_i|X=x}(\alpha_l)$ and $\hat{u}_i(x) = Q_{Y_i|X=x}(\alpha_u)$, where $Q_{Y_i|X=x}(\alpha)$ is the quantile function of Y_i conditional on $X = x$ evaluated in α. This section introduces two propositions where `M-CP` requests two different nominal coverage levels $\alpha_u - \alpha_l$, namely $\sqrt[d]{1-\alpha}$ and $1 - \alpha$. The propositions show that `M-CP` can achieve conditional coverage under two contrasting scenarios: independence or total dependence between the dimensions of the output.

Proposition 9. Assuming $Y_1, \ldots, Y_d$ are conditionally independent given X, `M-CP` achieves conditional coverage if $\alpha_u - \alpha_l = \sqrt[d]{1-\alpha}$.

Proof. For any $x \in \mathcal{X}$ and $i \in [d]$, we first establish that the $\sqrt[d]{1-\alpha}$th quantile of the distribution of $s_i(X, Y_i)$ given $X = x$ equals 0:

$$\begin{aligned}
\mathbb{P}(s_i(X, Y_i) \leqslant 0 \mid X = x) &= \mathbb{P}(\max\{\hat{l}_i(X) - Y_i, Y_i - \hat{u}_i(X)\} \leqslant 0 \mid X = x) && \text{(E.77)} \\
&= \mathbb{P}(\hat{l}_i(X) \leqslant Y_i \wedge Y_i \leqslant \hat{u}_i(X) \mid X = x) && \text{(E.78)} \\
&= 1 - \mathbb{P}(\hat{l}_i(X) > Y_i \vee Y_i > \hat{u}_i(X) \mid X = x) && \text{(E.79)} \\
&= 1 - \mathbb{P}(\hat{l}_i(X) > Y_i \mid X = x) - \mathbb{P}(Y_i > \hat{u}_i(X) \mid X = x) && \text{(E.80)} \\
&= 1 - \alpha_l - (1 - \alpha_u) && \text{(E.81)} \\
&= \alpha_u - \alpha_l && \text{(E.82)} \\
&= \sqrt[d]{1-\alpha}. && \text{(E.83)}
\end{aligned}$$

Using (E.83), we show that the $1-\alpha$th quantile of the distribution of $s_{\text{M-CP}}(X,Y)$ given $X=x$ is 0:

$$\begin{aligned}
\mathbb{P}(s_{\text{M-CP}}(X,Y) \leqslant 0 \mid X=x) &= \mathbb{P}(s_i(X,Y_i) \leqslant 0, \forall i \in [d] \mid X=x) && \text{(E.84)}\\
&= \mathbb{P}(s_1(X,Y_1) \leqslant 0 \wedge \cdots \wedge s_d(X,Y_d) \leqslant 0 \mid X=x) && \text{(E.85)}\\
&= \mathbb{P}(s_1(X,Y_1) \leqslant 0 \mid X=x) \ldots \mathbb{P}(s_d(X,Y_d) \leqslant 0 \mid X=x) && \text{(E.86)}\\
&= \sqrt[d]{1-\alpha}^{d} && \text{(E.87)}\\
&= 1-\alpha, && \text{(E.88)}
\end{aligned}$$

where (E.86) is obtained by conditional independence of $Y_1,\ldots,Y_d$ given X. Marginalizing over X, we obtain that the $1-\alpha$th quantile of $s_{\text{M-CP}}(X,Y)$ is 0:

$$\begin{aligned}
\mathbb{P}(s_{\text{M-CP}}(X,Y) \leqslant 0) &= \mathbb{E}_X[\mathbb{P}(s_{\text{M-CP}}(X,Y) \leqslant 0 \mid X)] && \text{(E.89)}\\
&= \mathbb{E}_X[1-\alpha] && \text{(E.90)}\\
&= 1-\alpha. && \text{(E.91)}
\end{aligned}$$

In the limit of $|\mathcal{D}_{\text{cal}}| \to \infty$, thanks to the Glivenko-Cantelli theorem, $\mathbb{P}(Y \in \hat{R}_{\text{M-CP}}(X)) = 1-\alpha$ and the quantile $\hat{q}$ obtained by SCP is thus 0.

Finally, using (E.88) and $\hat{q}=0$, we obtain that `M-CP` achieves conditional coverage:

$$\mathbb{P}(Y \in \hat{R}_{\text{M-CP}}(X) \mid X=x) = \mathbb{P}(s_{\text{M-CP}}(X,Y) \leqslant \hat{q} \mid X=x) = 1-\alpha. \tag{E.92}$$

□

Proposition 10. Assuming $Y_1|X \overset{\text{a.s.}}{=} \ldots \overset{\text{a.s.}}{=} Y_d|X$, `M-CP` achieves conditional coverage if $\alpha_u - \alpha_l = 1-\alpha$.

Proof. Let $x \in \mathcal{X}$. Using (E.82), we first show that the $1-\alpha$th conditional quantile of the distribution of $s_i(X,Y_i)$, for any $i \in [d]$, is 0:

$$\begin{aligned}
\mathbb{P}(s_i(X,Y_i) \leqslant 0 \mid X=x) &= \alpha_u - \alpha_l && \text{(E.93)}\\
&= 1-\alpha. && \text{(E.94)}
\end{aligned}$$

Using (E.94), we show that the $1-\alpha$th quantile of the distribution of $s_{\text{M-CP}}(X,Y)$ given X is 0:

$$\begin{aligned}
\mathbb{P}(s_{\text{M-CP}}(X,Y) \leqslant 0 \mid X=x) &= \mathbb{P}(s_i(X,Y_i) \leqslant 0, \forall i \in [d] \mid X=x) && \text{(E.95)}\\
&= \mathbb{P}(s_1(X,Y_1) \leqslant 0 \wedge \cdots \wedge s_d(X,Y_d) \leqslant 0 \mid X=x) && \text{(E.96)}\\
&= \mathbb{P}(s_1(X,Y_1) \leqslant 0 \mid X=x) && \text{(E.97)}\\
&= 1-\alpha, && \text{(E.98)}
\end{aligned}$$

where (E.97) is due to $Y_1|X \overset{\text{a.s.}}{=} \ldots \overset{\text{a.s.}}{=} Y_d|X$, which implies that, conditional on $X=x$, $\hat{l}_1(x) = \cdots = \hat{l}_d(x)$ and $\hat{u}_1(x) = \cdots = \hat{u}_d(x)$, and thus $s_1(X,Y_1) = \cdots = s_d(X,Y_d)$. Using (E.91), we obtain that $\hat{q}=0$. Finally, using (E.98), we obtain that `M-CP` achieves conditional coverage:

$$\mathbb{P}(Y \in \hat{R}_{\text{M-CP}}(X) \mid X=x) = \mathbb{P}(s_{\text{M-CP}}(X,Y) \leqslant 0 \mid X=x) = 1-\alpha. \tag{E.99}$$

□

E.5.3 Connection between sample-based and density-based methods

This section proves the connections between sample-based and density-based methods as introduced in Section 5.5.3. We start by restating a known lemma of conformal prediction.

Lemma 9. Consider a conformal prediction method with conformity score s. If $g : \mathbb{R} \to \mathbb{R}$ is a strictly increasing function, then the method with conformity score $g \circ s$ will produce the same prediction sets.

Proof. For any $x \in \mathcal{X}$, consider the prediction set created with s as in Section 2.5.1:

$$\hat{R}(x) = \left\{y \in \mathcal{Y} : s(x,y) \leqslant \text{Quantile}\left(\{s_i\}_{i \in [|\mathcal{D}_{\text{cal}}|]} \cup \{\infty\}; k_\alpha\right)\right\}. \tag{E.100}$$

Since g is strictly increasing,

$$\begin{aligned}
\hat{R}(x) &= \left\{y \in \mathcal{Y} : g(s(x,y)) \leqslant g\left(\text{Quantile}\left(\{s_i\}_{i \in [|\mathcal{D}_{\text{cal}}|]} \cup \{\infty\}; k_\alpha\right)\right)\right\} && \text{(E.101)}\\
&= \left\{y \in \mathcal{Y} : g(s(x,y)) \leqslant \text{Quantile}\left(\{g(s_i)\}_{i \in [|\mathcal{D}_{\text{cal}}|]} \cup \{\infty\}; k_\alpha\right)\right\}. && \text{(E.102)}
\end{aligned}$$

Since (E.102) corresponds to the prediction set with conformity score $g \circ s$, this shows that the two methods create the same regions. □

Proposition 3. `PCP` is equivalent to `DR-CP` with $\hat{f}_{Y|X=x} = \hat{f}^{\max}_{Y|X=x}$. Similarly, `HD-PCP` is equivalent to `DR-CP` with $\hat{f}_{Y|X=x} = \hat{f}^{\max}_{Y|X=x}$ where only $\lfloor (1-\alpha)L \rfloor$ samples with the highest density among $\{\tilde{Y}^{(l)}\}_{l \in [L]}$ are kept. Finally, `C-PCP` is equivalent to `C-HDR` with $\hat{f}_{Y|X=x} = \hat{f}^{\max}_{Y|X=x}$.

Proof. In the following proof, we note $a \uparrow b$ to signify that there exists a strictly increasing function g such that $a = g(b)$. Consider `DR-CP` with $\hat{f}_{Y|X} = \hat{f}^{\max}_{Y|X}$. We have:

$$\begin{aligned}
s_{\text{DR-CP}}(x,y) &= -\hat{f}^{\max}_{Y|X=x}(y) && && \text{(E.103)}\\
&\uparrow -\max_{l \in [L]} f_{\mathbb{S}}(y; \tilde{Y}^{(l)}) && (\hat{f}^{\max}_{Y|X=x}(y) = \max_{l \in [L]} f_{\mathbb{S}}(y; \tilde{Y}^{(l)})/C) && \text{(E.104)}\\
&= \min_{l \in [L]} -f_{\mathbb{S}}(y; \tilde{Y}^{(l)}) && && \text{(E.105)}\\
&\uparrow \min_{l \in [L]} \|y - \tilde{Y}^{(l)}\| && (f_{\mathbb{S}}(y; \tilde{Y}^{(l)}) \text{ has spherical level sets}) && \text{(E.106)}\\
&= s_{\text{PCP}}(x,y). && && \text{(E.107)}
\end{aligned}$$

We obtain the equivalence between the two methods by Lemma 9. The proof for HD-PCP follows the same arguments.

We now consider `C-HDR` with $\hat{f}_{Y|X} = \hat{f}^{\max}_{Y|X}$. We have:

$$s_{\text{C-HDR}}(x,y) = \frac{1}{K} \sum_{k \in [K]} \mathbb{1}(\hat{f}^{\max}_{Y|X=x}(\hat{Y}^{(k)}) \geqslant \hat{f}^{\max}_{Y|X=x}(y)) \quad \text{where } \hat{Y}^{(k)} \sim \hat{P}_{Y|X=x}, k \in [K]. \tag{E.108}$$

Developing the inequality for $k \in [K]$, we obtain:

$$\hat{f}^{\max}_{Y|X=x}(\hat{Y}^{(k)}) \geqslant \hat{f}^{\max}_{Y|X=x}(y) \tag{E.109}$$

$$\iff \max_{l\in[L]} f_{\mathbb{S}}(\hat{Y}^{(k)}; \tilde{Y}^{(l)}) \geqslant \max_{l\in[L]} f_{\mathbb{S}}(y; \tilde{Y}^{(l)}) \qquad (\hat{f}^{\max}_{Y|X=x}(y) = \max_{l\in[L]} f_{\mathbb{S}}(y; \tilde{Y}^{(l)})/C) \tag{E.110}$$

$$\iff \min_{l\in[L]} -f_{\mathbb{S}}(\hat{Y}^{(k)}; \tilde{Y}^{(l)}) \leqslant \min_{l\in[L]} -f_{\mathbb{S}}(y; \tilde{Y}^{(l)}) \tag{E.111}$$

$$\iff \min_{l\in[L]} \|\hat{Y}^{(k)} - \tilde{Y}^{(l)}\| \leqslant \min_{l\in[L]} \|y - \tilde{Y}^{(l)}\|. \qquad (f_{\mathbb{S}}(y; \tilde{Y}^{(l)}) \text{ has spherical level sets}) \tag{E.112}$$

Noting that (E.108) with (E.112) corresponds to the conformity score of C-PCP, we obtain the equivalence. □

E.6. Experimental setup

This section describes our experimental setup in more details. Computations were performed based on 2 workstations, one with with 2 A6000 GPUs and 64 CPU threads, and one with 2 A5000 GPUs and 64 CPU threads, running for 48 hours.

E.6.1 Datasets

We consider a total of 13 datasets that have been used in previous studies. Since our focus is on multivariate prediction sets, we select only datasets with an output that is at least two-dimensional. Specifically, we include 6 datasets from Feldman et al. (2023), 4 datasets from Tsoumakas et al. (2011) (MULAN benchmark), 1 dataset from Z. Wang et al. (2023), 1 datasets from Barrio et al. (2024), and 1 dataset from Camehl et al. (2024).

Each dataset is split into training, validation, calibration, and test sets with 2048 points reserved for calibration. The remaining data is split into 55% for training, 15% for validation and 30% for testing. The preprocessing follows the setup described in Grinsztajn et al. (2022). Table A.2 provides the detailed characteristics of each dataset.

E.6.2 Base predictors

We consider multiple base predictors that were presented in Section 2.2.3 and focus on MQF^2 for our main experiments (Section 5.6). In this section, we give further details on hyperparameters.

MQF^2. The underlying model of MQF^2 is a partially input-convex neural network (PINN, Amos et al., 2017) with two hidden layers, each containing 30 units. Increasing the number of parameters did not significantly improve performance, which is partly due to the efficiency of Convex Potential Flows compared to other normalizing flows (C.-W. Huang et al., 2021). While hyperparameter tuning for each dataset could enhance performance, it is not the primary focus of this paper.

MQF^2 is trained using maximum likelihood estimation with early stopping, with a patience of 15 epochs, where validation loss is measured every two epochs.

Distributional random forests. The minimum node size is set to 15, the forest consists of 2000 trees, and the splitting criterion is the maximum mean discrepancy (MMD).

Since this method does not operate in a latent space, we do not consider `L-CP` in combination with this base predictor. CD diagrams for this predictor are presented in Section E.7.2.

Cholesky-based mixture density network. The model is trained using maximum likelihood estimation and produces Gaussian mixtures with $M = 10$ components.

Similarly to DRF, this method does not operate in a latent space, and thus we do not consider `L-CP`. CD diagrams for this predictor are presented in Section E.7.3.

E.6.3 Adaptation of conformal methods into a common framework.

To ensure a fair comparison among conformal methods, we apply the calibration step using the same base predictors. Only `M-CP`, `CopulaCPTS`, and `STDQR` require slight modifications from their original formulations.

For `M-CP` and `CopulaCPTS`, direct estimation of marginal distributions for each output $Y_i, i \in [d]$ is infeasible with MQF^2. Instead, we estimate the lower and upper quantiles by first sampling $\{\tilde{Y}^{(l)}\}_{l \in [L]}$ from $\hat{P}_{Y|X=x}$ given $x \in \mathcal{X}$, and then computing the empirical quantiles $\tilde{Y}_i^{(\lfloor L\frac{\alpha}{2} \rfloor)}$ and $\tilde{Y}_i^{(\lfloor L(1-\frac{\alpha}{2}) \rfloor)}$. Sampling time is not accounted for in time computations for these methods. While a more computationally efficient base predictor could be used, this approach ensures a direct comparison with other conformal methods by maintaining consistency in the base predictor.

For `STDQR`, we modify the original method by replacing the conditional variational autoencoder (CVAE) with a normalizing flow. Following recommendations for future work from Feldman et al. (2023), we exploit the property that the output is normally distributed in the latent space and replace the base predictor by a normalizing flow. This adaptation leverages the assumption that the output is normally distributed in the latent space, allowing for an exact inverse transformation and eliminating a potential source of noise. To construct a region $R_{\mathcal{Z}}$ with coverage $1 - \alpha$ in the latent space, we select the $1 - \alpha$ proportion of samples closest to the origin, ensuring correct coverage without the need for directional quantile regression. The calibration procedure remains unchanged.

E.6.4 Metrics

Marginal coverage. Marginal coverage is measured using

$$\text{MC} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(x,y) \in \mathcal{D}_{\text{test}}} \mathbb{1}(y \in \hat{R}(x)). \tag{E.113}$$

Region size. We report the mean set size

$$\text{Mean Size} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(x,y) \in \mathcal{D}_{\text{test}}} |\hat{R}(x)|. \tag{E.114}$$

To avoid large regions disproportionately affecting the result, we also report the median of the set sizes

$$\text{Median Size} = \text{Quantile}(\{|\hat{R}(x)|\}_{(x,y)\in\mathcal{D}_{\text{test}}}; 0.5) \tag{E.115}$$

Computing the size of the region is challenging in high dimensions. Hence, we propose an unbiased estimator of the set size using importance sampling:

$$|\hat{R}(x)| = \int_{\mathcal{Y}} \mathbb{1}(y \in \hat{R}(x))dy = \mathbb{E}_{\hat{Y}\sim\hat{P}_{Y|X=x}}\left[\frac{\mathbb{1}(\hat{Y} \in \hat{R}(x))}{\hat{f}_{Y|X=x}(\hat{Y})}\right] \approx \frac{1}{K}\sum_{k=1}^{K}\frac{\mathbb{1}(\hat{Y}^{(k)} \in \hat{R}(x))}{\hat{f}_{Y|X=x}(\hat{Y}^{(k)})}, \tag{E.116}$$

where $\hat{Y}^{(k)} \sim \hat{P}_{Y|X=x}$, $k \in [K]$. This estimator is compatible with all base predictors in Section E.6.2 since it is both possible to sample from their predictive distribution and evaluate the PDF. In Section E.6.7, we discuss the efficiency of this estimator.

Conditional coverage. To ensure a robust evaluation of conditional coverage, we consider three different conditional coverage metrics, detailed in Section E.6.6. The Worst Slab Coverage (WSC, Cauchois et al., 2021) groups inputs into "slabs" and evaluates the worst obtained coverage. The coverage error conditional on X (CEC-X) partitions the input space $\mathcal{X}$ and evaluates coverage on each subset. The coverage error conditional on $V = \hat{f}_{Y|X}(\hat{Y})$, where $\hat{Y} \sim \hat{P}_{Y|X}$, (CEC-V, Izbicki et al., 2022; Dheur et al., 2024), creates a partition based on the distribution of V, which is more robust to high-dimensional inputs.

Computing time. We report the total time required for calibration and testing the marginal coverage. Specifically, this requires evaluating conformity scores on $\mathcal{D}_{\text{cal}}$ followed by evaluating conformity scores on $\mathcal{D}_{\text{test}}$.

E.6.5 Multi-Model, Multi-Dataset Comparison

In order to determine whether there are significant differences in model performance, we construct CD diagrams using the procedure described in Section 3.6.1. For MC and WSC, the CD diagrams report $|\text{MC} - (1-\alpha)|$ and $|\text{WSC} - (1-\alpha)|$, both of which should be minimized.

E.6.6 Metrics of Conditional Coverage

In this section, $\hat{\mathbb{P}}_{\mathcal{D}_{\text{test}}}(E)$ denotes the empirical probability of an event E over the dataset $\mathcal{D}_{\text{test}}$, i.e., the fraction of points in $\mathcal{D}_{\text{test}}$ for which E is true.

Worst Slab Coverage. Introduced in Cauchois et al. (2021), the *Worst Slab Coverage* (WSC) metric quantifies the minimal coverage over all possible slabs in $\mathbb{R}^d$, where each slab contains at least a fraction δ of the total mass, with $0 < \delta \leqslant 1$. For a given vector $v \in \mathbb{R}^d$, the WSC associated with v, denoted as WSC_v, is defined by:

$$\text{WSC}_v = \inf_{a<b}\left\{\hat{\mathbb{P}}_{\mathcal{D}_{\text{test}}}\left(y \in \hat{R}(x) \mid a \leqslant v^\top x \leqslant b\right) \text{ s.t. } \hat{\mathbb{P}}_{\mathcal{D}_{\text{test}}}(a \leqslant v^\top x \leqslant b) \geqslant \delta\right\}, \tag{E.117}$$

where $a, b \in \mathbb{R}$. This metric assesses conditional coverage by focusing on inputs x that lie within a slab defined by v, using the inner product $v^\top x$ to measure similarity.

To estimate the worst-case slab, we follow the method from Cauchois et al. (2021), uniformly sampling 1,000 vectors v_j from the unit sphere $\mathbb{S}^{d-1}$ and calculating:

$$\text{WSC} = \min_{v_j \in \mathbb{S}^{d-1}} \text{WSC}_{v_j}. \tag{E.118}$$

To mitigate overfitting on the test dataset, we partition the test set into two subsets, $\mathcal{D}_{\text{test}} = \mathcal{D}_{\text{test}}^{(1)} \cup \mathcal{D}_{\text{test}}^{(2)}$, as in Romano et al. (2020) and Sesia and Romano (2021). We identify the worst combination of a, b, and v on $\mathcal{D}_{\text{test}}^{(1)}$ by minimizing the WSC metric with $\delta = 0.2$, and then evaluate conditional coverage on the separate subset $\mathcal{D}_{\text{test}}^{(2)}$.

As noted by Cauchois et al. (2021), the worst-slab coverage for a given projection can be computed in linear time ($O(n)$) using an algorithm for the maximum density segment problem (K.-M. Chung and H.-I. Lu, 2005). However, existing Python implementations are inefficient, typically relying on a simpler, non-vectorized $O(n^2)$ approach. To address this gap, we introduce a Python package[1] featuring an optimized $O(n)$ C++ backend, making this metric practical for large-scale use.

CEC-X. CEC-X approximates conditional coverage by partitioning the input space $X \in \mathcal{X} \subseteq \mathbb{R}^p$. We apply the k-means++ clustering algorithm on the inputs $X^{(i)}$ in the validation dataset $\mathcal{D}_{\text{val}}$, creating a partition $\mathcal{A} = A_1 \cup \cdots \cup A_J$ over $\mathcal{X}$. The *Coverage Error Conditional on X* is defined as:

$$\text{CEC-X} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{i=1}^{|\mathcal{D}_{\text{test}}|} \frac{1}{J} \sum_{j=1}^{J} \left(\hat{\mathbb{P}}_{\mathcal{D}_{\text{test}}} \left(y \in \hat{R}(x) \;\middle|\; x \in A_j \right) - (1-\alpha) \right)^2. \tag{E.119}$$

CEC-V. CEC-V is similar to CEC-X, but the conditioning is on the distribution of $\log(V) = \log(\hat{f}_{Y|X}(\hat{Y}))$, where $\hat{Y} \sim \hat{P}_{Y|X}$. Unlike CEC-X, CEC-V is more robust to high-dimensional inputs. This approach originates from the CD-split$^+$ method (Izbicki et al., 2022) and has been adapted to multivariate outputs in Dheur et al. (2024).

In practice, given an input x, a new feature v_x is created. First, samples v_i from $V \mid X = x$ are generated by sampling $y_1, \ldots, y_m \sim \hat{P}_{Y|X=x}$ and evaluating $v_i = \hat{f}_{Y|X=x}(y_i)$. The resulting vector $v_x = (v_{(1)}, \ldots, v_{(m)})$ consists of the order statistics $v_{(i)}$ from $v_1, \ldots, v_m$.

The k-means++ clustering algorithm is applied on the vectors $\log(v_{X^{(i)}})$ in the validation dataset $\mathcal{D}_{\text{val}}$, and a partition $\mathcal{A}_V = A_1 \cup \cdots \cup A_J$ over $\mathbb{R}^m$ is obtained. The *Coverage Error Conditional on the distribution of V* is then computed according to (E.119), using the partition $\mathcal{A}_V$.

Dheur et al. (2024) notes that the distance function corresponding to this partitioning approach is the 2-Wasserstein distance w.r.t. the distribution of V.

[1] `https://github.com/Vekteur/max-density-segment`

E.6.7 Estimator for the set size

In this section, we discuss the efficiency of the set size estimator introduced in Section E.6.4. This estimator is based on a density estimator $\hat{f}_{Y|X=x}$ and a sample $\hat{Y}^{(k)}, k \in [K]$, drawn i.i.d. from the conditional distribution $Y \mid X = x$ for any $x \in \mathcal{X}$. Specifically, the estimator is given by:

$$\hat{V}(x) = \frac{1}{K} \sum_{k=1}^{K} \frac{\mathbb{1}(\hat{Y}^{(k)} \in \hat{R}(x))}{\hat{f}_{Y|X=x}(\hat{Y}^{(k)})}.$$

While the estimator is unbiased, i.e., $\mathbb{E}\left[\hat{V}(x)\right] = |\hat{R}(x)|$, we want to study its variance. Let $I = \mathbb{1}(\hat{Y} \in \hat{R}(x))$ represent the indicator that a sample $\hat{Y}$ lies within the prediction set $\hat{R}(x)$, and let $\rho = \mathbb{P}(\hat{Y} \in \hat{R}(x))$ denote the coverage probability obtained from the samples based on our density estimator. Using the law of total variance, we obtain the following expression for the variance of $\hat{V}(x)$:

$$\begin{aligned}
\mathbb{V}\left[\hat{V}(x)\right] &= \frac{1}{K}\mathbb{V}\left[\frac{I}{\hat{f}_{Y|X=x}(\hat{Y})}\right] \\
&= \frac{1}{K}\left(\mathbb{E}\left[\mathbb{V}\left[\frac{I}{\hat{f}_{Y|X=x}(\hat{Y})} \mid I\right]\right] + \mathbb{V}\left[\mathbb{E}\left[\frac{I}{\hat{f}_{Y|X=x}(\hat{Y})} \mid I\right]\right]\right) \\
&= \frac{1}{K}\left(\rho\mathbb{V}\left[\frac{1}{\hat{f}_{Y|X=x}(\hat{Y})}\right] + \rho(1-\rho)\mathbb{E}\left[\frac{1}{\hat{f}_{Y|X=x}(\hat{Y})}\right]^2\right).
\end{aligned}$$

Assuming that the density estimate corresponds to the true density, i.e. $\hat{f}_{Y|X=x} = f_{Y|X=x}$ and that $\hat{R}$ achieves conditional coverage, then $\rho = 1 - \alpha$, and we obtain:

$$\mathbb{V}\left[\hat{V}(x)\right] = \frac{1}{K}\left((1-\alpha)\mathbb{V}\left[\frac{1}{f_{Y|X=x}(Y)}\right] + \alpha(1-\alpha)\mathbb{E}\left[\frac{1}{f_{Y|X=x}(Y)}\right]^2\right).$$

This indicates that the variance of our estimator only depends on the variance and expectation of the random variable $\frac{1}{f_{Y|X=x}(Y)}$. In this case, the variance does not directly depend on the output dimension d.

Figure E.5 shows how the estimator behaves in a scenario with a specific density estimator and prediction set with varying output dimension d and an 80% coverage level. Since there is no dependence on X, we abbreviate the notation as follows: $\hat{R} = \hat{R}(x)$, $\hat{f}(y) = \hat{f}_{Y|X=x}(y)$, and $\hat{V} = \hat{V}(x)$ for any $x \in \mathcal{X}$. The density estimator is a standard normal distribution $\hat{f}(y) = \mathcal{N}(y; 0, I_d)$ and the prediction set is a ball $\hat{R} = \left\{y \in \mathcal{Y} : \|y\| \leqslant Q_{\chi^2_d}(1-\alpha)\right\}$, where χ^2_d is the chi-squared distribution with d degrees of freedom and $Q_{\chi^2_d}$ is its quantile function. It can be shown that $\mathbb{P}(\hat{Y} \in \hat{R}) = 1 - \alpha$. In this case, the volume V of $\hat{R}$ can be computed exactly.

Each of the first four panels in Figure E.5 shows five trajectories for $\log \hat{V}$ as K increases from 1 to 100. The true volume, $\log V$, of the prediction set is indicated by a dashed line. We observe

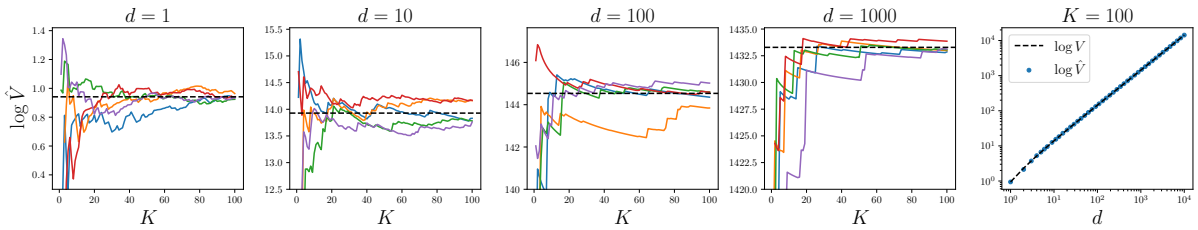


Figure E.5: Panels 1 to 4: Trajectories of the log volume estimator with increasing K compared to the true log volume (dashed line) for different output dimensions d. Panel 5: Log volume estimator with $K = 100$ compared to the true log volume (dashed line).

that the estimator converges within a reasonable range of the true volume for varying output dimensions d. The last panel illustrates the value of $\log \hat{V}$ as a function of d, with $\log V$ again marked by a dashed line. From this, we observe that the estimator remains close to the true volume across different output dimensions d.

E.7. Additional results

This section presents additional results for the models MQF^2 (Section E.7.1), DRF (Section E.7.2) and MDN (Section E.7.3). The experimental setup is described in Section E.6.

E.7.1 MQF²

Figure E.6 presents the marginal coverage and median set size across datasets of increasing size for MQF^2. In Panel 1, all methods except CopulaCPTS attain precise marginal coverage. This is expected since these methods follow the SCP algorithm (Section 2.5.1) and their marginal coverage conditional on the calibration dataset and samples from the calibration dataset follows a Beta distribution whose parameters only depend on the size of the calibration dataset (Section E.5.1). While CopulaCPTS attains marginal coverage, the larger variance in its marginal coverage arises because it does not follow the SCP algorithm.

In Panel 2, the median set size is normalized between 0 and 1 for each dataset in order to facilitate comparison. We observe that `C-HDR` often obtains the smallest median set size. The performance of the other methods can vary highly across datasets for the median set size and is better visualized in a CD diagram (see Figure E.7).

Table E.2: Median set size with the base predictor MQF^2.

Dataset	M-CP	CopulaCPTS	DR-CP	C-HDR	PCP	HD-PCP	STDQR	C-PCP	L-CP
households	$14.2_{0.48}$	$12.3_{0.87}$	$13.2_{0.29}$	$\mathbf{10.6_{0.33}}$	$20.5_{0.38}$	$15.6_{0.39}$	$17.8_{0.41}$	$15.5_{0.74}$	$18.6_{0.80}$
scm20d	$67.6_{8.5}$	$1.12e+02_{2.1e+01}$	$2.33e+02_{2.2e+01}$	$42.0_{7.9}$	$1.05e+02_{1.1e+01}$	$94.4_{9.4}$	$99.4_{1.0e+01}$	$\mathbf{26.0_{3.1}}$	$72.0_{1.0e+01}$
rf2	$0.00547_{0.0027}$	$0.00555_{0.0033}$	$0.00215_{0.0010}$	$\mathbf{0.000690_{0.00032}}$	$0.00700_{0.0036}$	$0.00617_{0.0030}$	$0.00624_{0.0031}$	$0.00262_{0.0012}$	$\mathbf{0.00104_{0.00048}}$
rf1	$0.00547_{0.0027}$	$0.00555_{0.0033}$	$0.00215_{0.0010}$	$\mathbf{0.000690_{0.00032}}$	$0.00700_{0.0036}$	$0.00617_{0.0030}$	$0.00624_{0.0031}$	$0.00262_{0.0012}$	$\mathbf{0.00104_{0.00048}}$
scm1d	$0.528_{0.046}$	$0.323_{0.050}$	$0.867_{0.078}$	$0.239_{0.026}$	$0.698_{0.065}$	$0.684_{0.062}$	$0.671_{0.069}$	$\mathbf{0.216_{0.024}}$	$\mathbf{0.197_{0.020}}$
meps_21	$0.185_{0.013}$	$0.171_{0.014}$	$0.227_{0.013}$	$\mathbf{0.132_{0.024}}$	$0.359_{0.021}$	$0.246_{0.015}$	$0.283_{0.015}$	$0.220_{0.021}$	$0.244_{0.052}$
meps_19	$0.214_{0.022}$	$0.595_{0.42}$	$0.175_{0.011}$	$\mathbf{0.119_{0.019}}$	$0.396_{0.059}$	$0.266_{0.033}$	$0.307_{0.043}$	$0.238_{0.026}$	$0.232_{0.043}$
meps_20	$0.371_{0.061}$	$0.362_{0.059}$	$0.223_{0.020}$	$\mathbf{0.114_{0.012}}$	$0.535_{0.050}$	$0.436_{0.066}$	$0.472_{0.052}$	$0.341_{0.039}$	$0.280_{0.028}$
house	$1.17_{0.023}$	$1.22_{0.043}$	$\mathbf{0.664_{0.021}}$	$\mathbf{0.651_{0.016}}$	$0.882_{0.023}$	$0.680_{0.018}$	$0.799_{0.023}$	$0.858_{0.018}$	$1.19_{0.017}$
bio	$0.303_{0.0066}$	$0.296_{0.0092}$	$0.257_{0.0067}$	$\mathbf{0.218_{0.0053}}$	$0.343_{0.0076}$	$0.259_{0.0065}$	$0.269_{0.0067}$	$0.302_{0.0074}$	$0.267_{0.0061}$
blog_data	$0.170_{0.039}$	$0.0948_{0.015}$	$0.0374_{0.0056}$	$\mathbf{0.0155_{0.0031}}$	$0.141_{0.023}$	$0.125_{0.023}$	$0.163_{0.036}$	$0.106_{0.021}$	$0.0676_{0.017}$
calcofi	$2.13_{0.024}$	$2.38_{0.12}$	$\mathbf{1.67_{0.022}}$	$1.99_{0.026}$	$2.33_{0.029}$	$1.89_{0.029}$	$1.97_{0.021}$	$2.81_{0.042}$	$2.70_{0.024}$
taxi	$4.26_{0.068}$	$4.72_{0.11}$	$\mathbf{2.62_{0.029}}$	$\mathbf{2.62_{0.033}}$	$4.03_{0.040}$	$3.18_{0.030}$	$3.63_{0.058}$	$4.02_{0.064}$	$4.94_{0.12}$

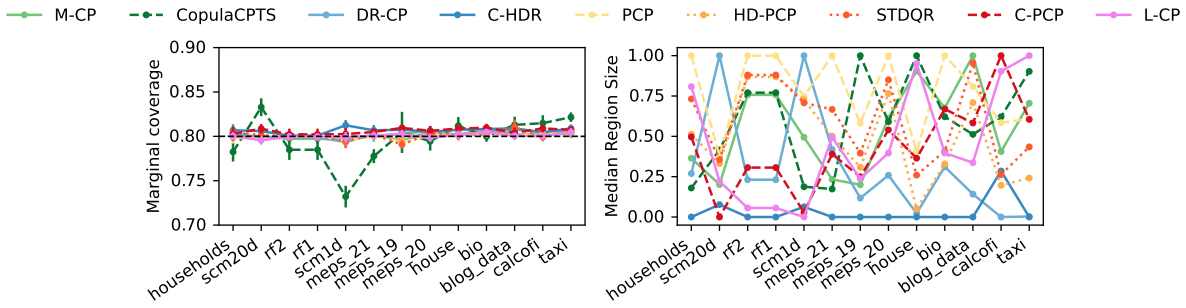


Figure E.6: Marginal coverage and median set size with the base predictor MQF^2 across datasets sorted by size.

Figure E.7 presents critical difference diagrams for three conditional coverage metrics (CEC-X, CEC-Z and WSC), the mean set size and median set size, the total calibration and test time. Results are consistent with the results from the main text.

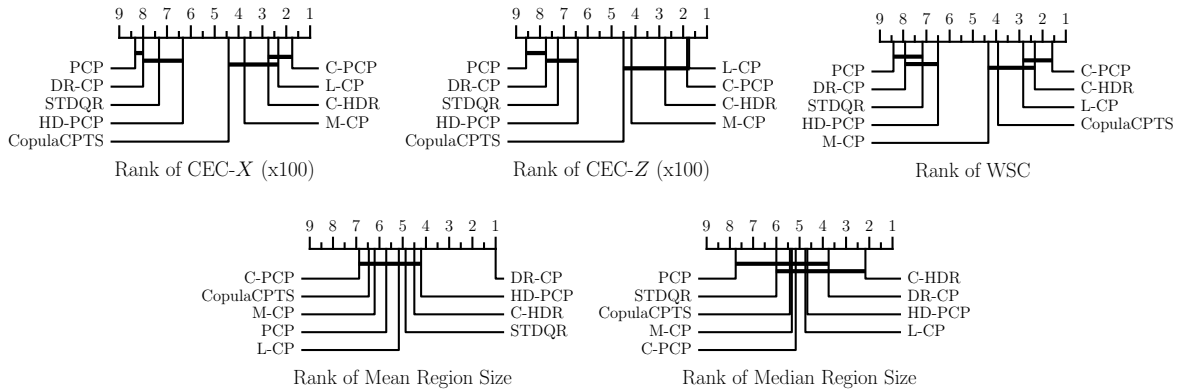


Figure E.7: CD diagrams with the base predictor MQF^2 with 10 runs per dataset and method.

Table E.3: Mean set size with the base predictor MQF^2.

Dataset	M-CP	CopulaCPTS	DR-CP	C-HDR	PCP	HD-PCP	STDQR	C-PCP	L-CP
households	$36.9_{0.86}$	$35.1_{1.4}$	**$15.7_{0.63}$**	$40.1_{1.2}$	$33.8_{2.1}$	$30.1_{2.1}$	$28.8_{2.1}$	$62.6_{2.5}$	$50.6_{1.4}$
scm20d	$7.03e+06_{2.5e+06}$	$3.82e+07_{1.6e+07}$	**$6.40e+03_{1.9e+03}$**	$5.30e+09_{2.1e+09}$	$1.61e+04_{5.1e+03}$	$1.56e+04_{5.1e+03}$	$1.59e+04_{5.1e+03}$	$1.37e+09_{1.0e+09}$	$2.20e+10_{9.1e+09}$
rf1	$1.86e+02_{1.0e+02}$	$1.83e+02_{9.6e+01}$	**$15.1_{9.9}$**	$3.50e+02_{1.7e+02}$	$1.29e+02_{8.1e+01}$	$1.09e+02_{6.7e+01}$	$4.05e+05_{2.1e+05}$	$9.85e+02_{4.8e+02}$	$4.01e+02_{1.9e+02}$
rf2	$1.86e+02_{1.0e+02}$	$1.83e+02_{9.6e+01}$	**$15.1_{9.9}$**	$3.51e+02_{1.7e+02}$	$1.29e+02_{8.1e+01}$	$1.09e+02_{6.7e+01}$	$4.05e+05_{2.1e+05}$	$9.87e+02_{4.8e+02}$	$4.02e+02_{1.9e+02}$
scm1d	$2.37e+05_{5.7e+04}$	$1.81e+05_{5.1e+04}$	$78.4_{1.6e+01}$	$2.73e+08_{5.3e+07}$	**$57.3_{1.8e+01}$**	**$43.0_{1.1e+01}$**	**$43.6_{1.1e+01}$**	$1.48e+08_{4.9e+07}$	$1.52e+08_{2.6e+07}$
meps_21	$1.21_{0.045}$	$1.17_{0.046}$	**$0.315_{0.020}$**	$1.44_{0.14}$	$0.617_{0.029}$	$0.558_{0.026}$	$0.553_{0.025}$	$2.07_{0.10}$	$1.96_{0.34}$
meps_19	$1.14_{0.027}$	$1.11_{0.031}$	**$0.293_{0.018}$**	$1.29_{0.056}$	$0.581_{0.027}$	$0.559_{0.021}$	$0.537_{0.021}$	$1.96_{0.072}$	$1.52_{0.053}$
meps_20	$1.20_{0.045}$	$1.17_{0.038}$	**$0.309_{0.014}$**	$1.30_{0.047}$	$0.606_{0.020}$	$0.562_{0.020}$	$0.546_{0.018}$	$2.01_{0.11}$	$1.59_{0.064}$
house	$1.83_{0.027}$	$1.81_{0.038}$	**$0.887_{0.033}$**	$1.09_{0.033}$	$1.23_{0.034}$	$0.964_{0.030}$	$1.14_{0.030}$	$1.44_{0.040}$	$1.71_{0.013}$
bio	$1.40_{0.39}$	$1.42_{0.39}$	**$0.269_{0.010}$**	$0.486_{0.037}$	$0.396_{0.014}$	$0.297_{0.0090}$	$0.311_{0.010}$	$1.41_{0.31}$	$2.57_{0.77}$
calcofi	$2.04_{0.023}$	$2.22_{0.071}$	**$1.42_{0.018}$**	$1.95_{0.040}$	$2.22_{0.031}$	$1.70_{0.022}$	$1.78_{0.026}$	$2.83_{0.036}$	$2.34_{0.044}$
blog_data	$0.390_{0.017}$	$0.390_{0.016}$	**$0.0852_{0.0050}$**	$0.375_{0.018}$	$0.173_{0.0097}$	$0.163_{0.0078}$	$0.188_{0.0067}$	$0.613_{0.028}$	$0.520_{0.027}$
taxi	$5.68_{0.084}$	$6.35_{0.16}$	**$2.67_{0.047}$**	$3.21_{0.049}$	$4.55_{0.090}$	$3.54_{0.075}$	$3.99_{0.071}$	$5.36_{0.090}$	$6.51_{0.12}$

E.7.2 Distributional random forests

Figure E.8 presents additional results for the base predictor DRF. Since this model does not rely on a latent space, results for `STDQR` and `L-CP` are not included.

In terms of conditional coverage, the results align with those of MQF^2, with `C-PCP` and `C-HDR` outperforming `DR-CP`, `PCP`, and `HD-PCP`. Notably, `M-CP` achieves competitive conditional coverage, suggesting it pairs well with `DRF-KDE`. Similar to MQF^2, all methods except for CopulaCPTS attain precise marginal coverage.

The median set size is normalized to a [0,1] range for each dataset to facilitate comparison. We observe that `C-HDR` generally achieves the smallest median set size, followed by `DR-CP`. The test time is the lowest for `M-CP` and `CopulaCPTS` while `C-PCP` and `C-HDR` obtain the highest computation times.

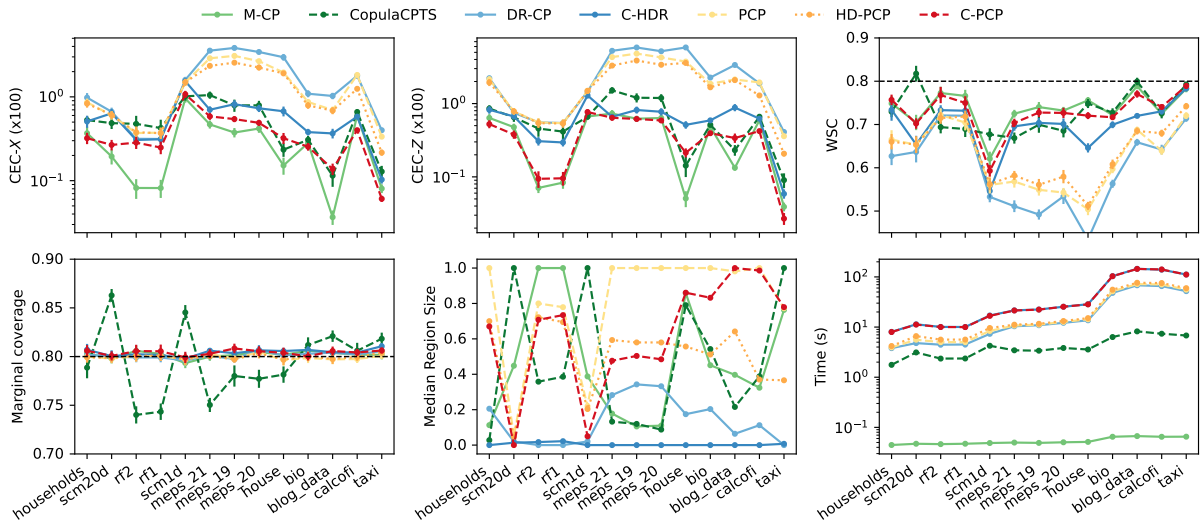


Figure E.8: Metrics across datasets sorted by size with the base predictor DRF.

Figure E.9 shows CD diagrams obtained with DRF as the base predictor. The results are consistent with Figure E.8.

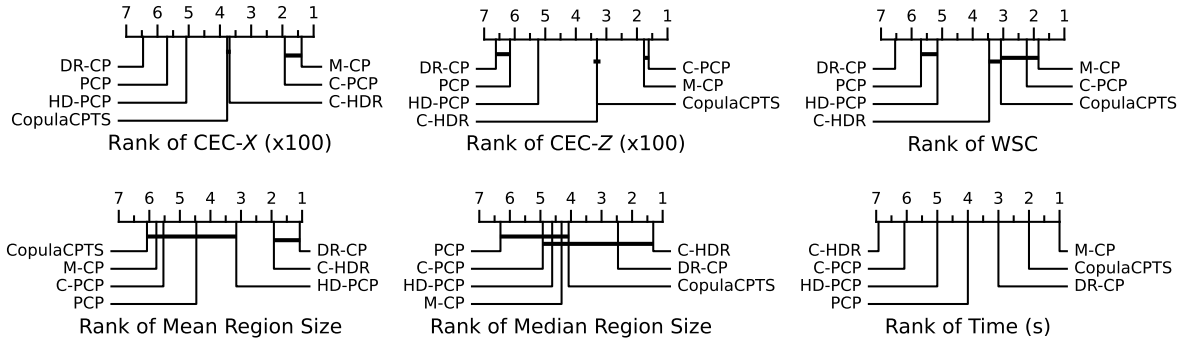


Figure E.9: CD diagrams with the base predictor DRF based on 10 runs per dataset and method.

E.7.3 Cholesky-based mixture density network

Figure E.10 presents additional results for the Cholesky-based mixture density network. Similarly to DRF, this model does not rely on a latent space and thus results for `STDQR` and `L-CP` are not included.

The conditional coverage also aligns with MQF^2, `C-PCP` and `C-HDR` outperforming `DR-CP`, `PCP`, and `HD-PCP`. `M-CP` and `CopulaCPTS` achieving intermediate conditional coverage. As expected, marginal coverage is precise for all methods except CopulaCPTS.

`C-HDR` often obtains the smallest median set size, while `DR-CP` consistently attains the best mean set size.

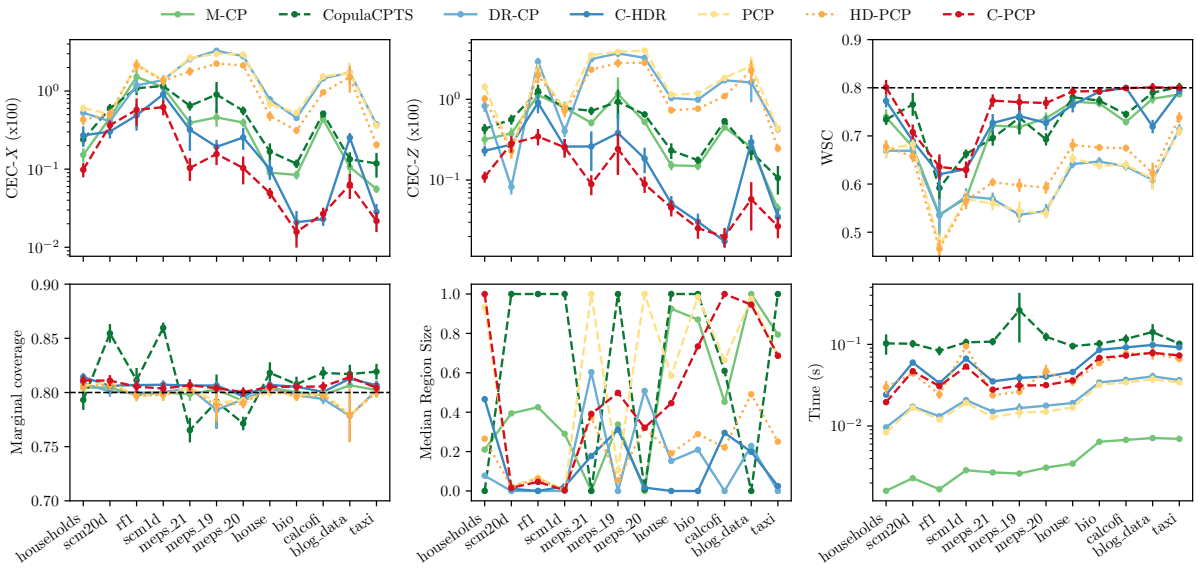


Figure E.10: Metrics across datasets sorted by size with the multivariate Gaussian mixture model base predictor.

CD diagrams in Figure E.11 are consistent with Figure E.10.

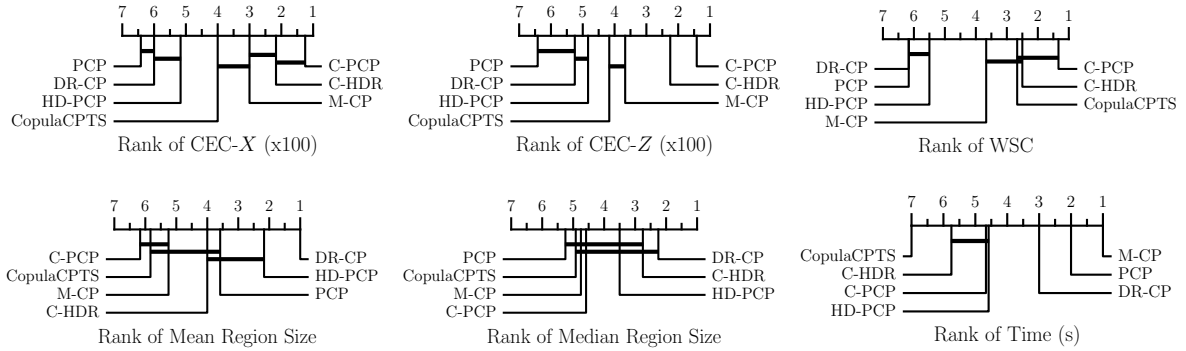


Figure E.11: CD diagrams based on Multivariate Gaussian Mixture Model parameterized by a hypernetwork with $M = 10$ and 10 runs per dataset and method.

E.7.4 Impact of the number of samples K

Figures E.12 and E.13 illustrate how conditional coverage, marginal coverage and prediction set size change as a function of$K$ on all datasets. For a better comparison among datasets, the metrics CEC-X, CEC-Z, the median set size and the mean set size are normalized between 0 and 1, with results averaged over 10 runs. Furthermore, the red line indicates a linear regression fit, allowing to see the trend.

Conditional coverage metrics decreasing with K indicate that conditional coverage tends to improve with an increasing number of samples. This is expected since an increasing number of Monte-Carlo samples allows a better estimation of the CDF of the scores in (5.10). Marginal coverage is obtained with any K. However, small sizes of K will lead to more duplicated conformity scores and thus a possibility of overcoverage. Median set sizes and mean set sizes also tend to decrease with K as the CDF approximation improves.

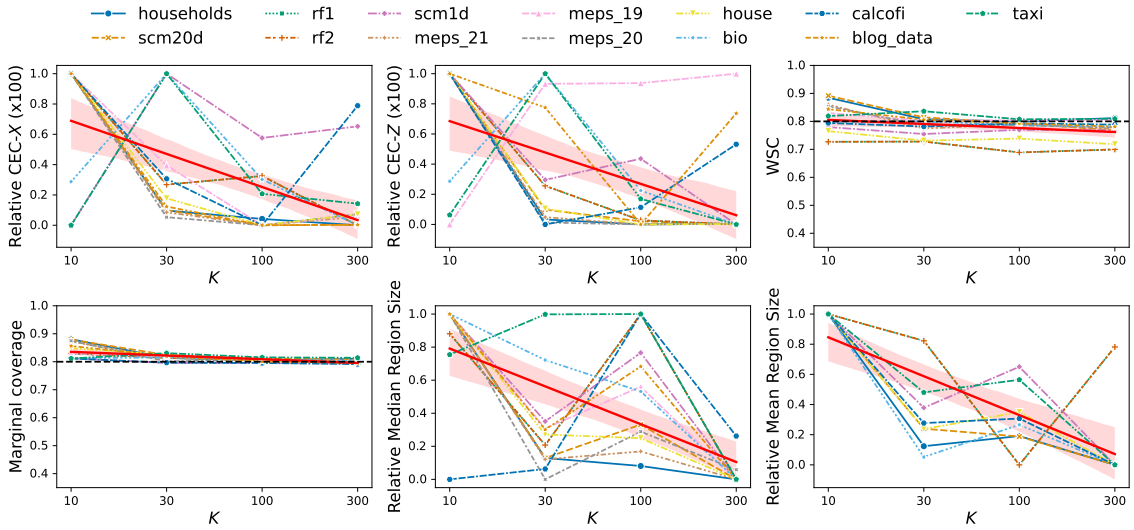


Figure E.12: Evolution of conditional coverage, marginal coverage and set sizes of `C-PCP` as a function of the number of samples K using the base predictor MQF^2. The metrics CEC-X, and CEC-Z should be minimized, while the marginal coverage and WSC should approach $1-\alpha$ (indicated by the dashed black line). The red line, obtained by linear regression, indicates the general trend.

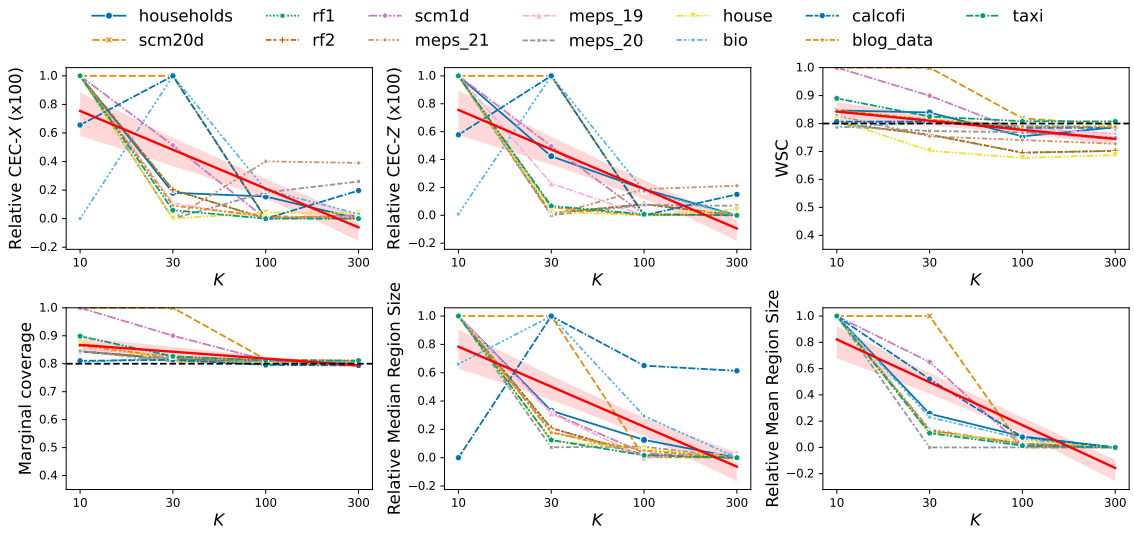


Figure E.13: Reproduction of Figure E.12 for `C-HDR`.

E.7.5 Comparison with Bonferroni correction

To better understand the prediction sets produced by FWER control methods, we provide a qualitative and quantitative comparison with Bonferroni correction. We consider Bonferroni correction applied to the scores of CQR (see (E.2)), similarly to M-CP. Figure E.14 provides

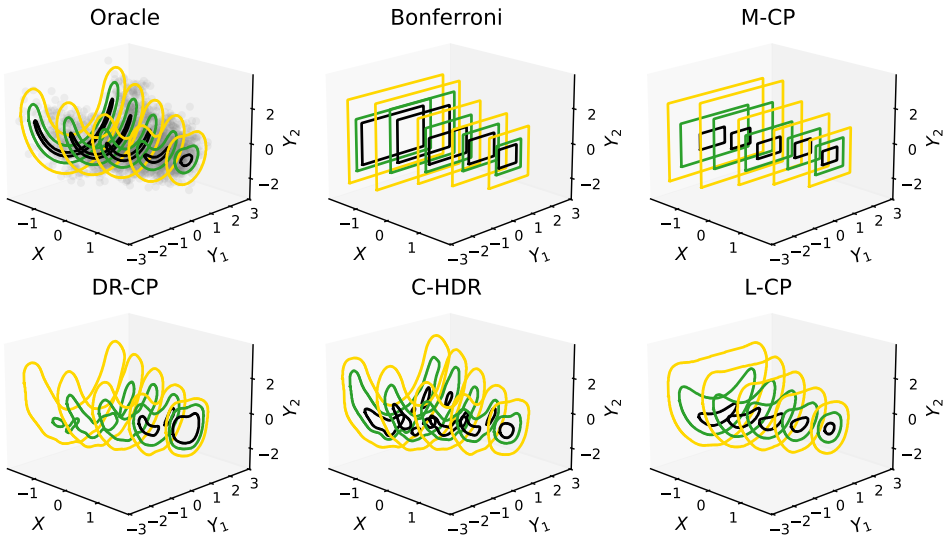


Figure E.14: Prediction sets for a bivariate unimodal dataset, conditional on a univariate input. The black, green, and yellow contours represent regions with nominal coverage levels of 20%, 40%, and 80%, respectively. The figure is similar to Figure 5.2 in the main text, with Bonferroni added as a comparison. Both Bonferroni and M-CP are based on Conformal Quantile Regression (CQR) applied separately for each dimension.

an illustrative example, and Table E.4 provides results on this same dataset. This shows that Bonferroni is computationally fast but produces larger regions due to the rectangular shape.

Table E.4: Detailed metrics for the unimodal heteroscedastic process from Figure E.14, with $1 - \alpha$ fixed to 0.8.

Method	MC	Median Size	CEC-X ($\times 100$)	CEC-Z ($\times 100$)	WSC	Test time
Bonferroni	$0.813_{0.0036}$	$9.07_{0.15}$	$0.0241_{0.012}$	$0.0249_{0.0098}$	$0.815_{0.0063}$	$\mathbf{0.00339_{5.9e\text{-}05}}$
M-CP	$\mathbf{0.801_{0.0037}}$	$8.62_{0.074}$	$0.0240_{0.0031}$	$0.0157_{0.0031}$	$\mathbf{0.796_{0.012}}$	$0.0959_{0.058}$
DR-CP	$\mathbf{0.796_{0.0019}}$	$\mathbf{6.83_{0.042}}$	$0.432_{0.019}$	$0.403_{0.015}$	$0.697_{0.0093}$	$0.0557_{0.00075}$
C-HDR	$0.809_{0.0025}$	$6.97_{0.039}$	$0.0129_{0.0059}$	$0.0155_{0.0037}$	$0.815_{0.0030}$	$14.2_{0.11}$
L-CP	$\mathbf{0.798_{0.0024}}$	$8.06_{0.035}$	$\mathbf{0.00586_{0.00095}}$	$\mathbf{0.00549_{0.0014}}$	$\mathbf{0.794_{0.0039}}$	$0.0584_{0.0012}$

E.8. Comparison between C-PCP and CP²-PCP

In this section, we compare our proposed method, `C-PCP`, with the CP^2-PCP method recently proposed by Plassier et al. (2025a). More generally, we also compare the methods from the CP^2 framework of Plassier et al. (2025a) with our class of CDF-based conformity scores (Section 5.4.1 in the main text). In Section E.8.1, we present the more general CP^2 framework using our own notation for clarity, with CP^2-PCP as a particular case of CP^2. In Section E.8.2, we discuss the asymptotic properties of CP^2 and show the asymptotic equivalence with CDF-based methods. In Section E.8.3, we discuss the relationship between CDF-based and CP^2-based methods.

E.8.1 The CP² framework

Let us define a family of non-decreasing nested regions $\{\mathcal{R}(x;t)\}_{t\in\mathbb{R}}$ such that $\bigcap_{t\in\mathbb{R}}\mathcal{R}(x;t)=\varnothing$ and $\bigcup_{t\in\mathbb{R}}\mathcal{R}(x;t)=\mathcal{Y}$, and $\bigcap_{t'<t}\mathcal{R}(x;t')=\mathcal{R}(x;t)$. Without loss of generality, these nested regions are expressed in terms of a conformity score $s_W(x,y)\in\mathbb{R}$ as follows:

$$\mathcal{R}(x;t)=\{y\in\mathcal{Y}: s_W(x,y)\leqslant t\}, \tag{E.120}$$

where $s_W(x,y)$ is continuous in y.

As the next step, we introduce a family of transformation functions $f_\tau(\lambda):\mathbb{R}\to\mathbb{R}$ parameterized by $\tau\in\mathbb{R}$. It is assumed that for any τ, the function $\lambda\mapsto f_\tau(\lambda)$ is increasing and bijective. Let $\varphi\in\mathbb{R}$ be a constant (e.g., $\varphi=1$). We also define the function $\tilde{f}_\varphi(\tau)=f_\tau(\varphi)$ and assume that $\tau\mapsto\tilde{f}_\varphi(\tau)$ is increasing and bijective.

As a first step towards defining CP^2, we construct a prediction set assuming knowledge of the conditional distribution $F_{Y|X}$. For a given input $x\in\mathcal{X}$, the prediction set is defined as:

$$\bar{R}_{\mathrm{CP}^2}(x)=\mathcal{R}(x;f_{\tau_\star(x)}(\varphi)), \tag{E.121}$$

where

$$\tau_\star(x)=\inf\{\tau:\mathbb{P}\left(Y\in\mathcal{R}(X;f_\tau(\varphi))\mid X=x\right)\geqslant 1-\alpha\} \tag{E.122}$$

implies that $\bar{R}_{\mathrm{CP}^2}(x)$ guarantees conditional coverage given x. Furthermore, using (E.120) and defining the random variable $W=s_W(X,Y)$, we can equivalently express (E.122) as

$$\begin{aligned}
\tau_\star(x) &= \inf\{\tau:\mathbb{P}\left(s_W(X,Y)\leqslant f_\tau(\varphi)\mid X=x\right)\geqslant 1-\alpha\} && \text{(E.123)}\\
&= \inf\left\{\tau:\mathbb{P}\left(\tilde{f}_\varphi^{-1}(s_W(X,Y))\leqslant\tau\mid X=x\right)\geqslant 1-\alpha\right\} && \text{(E.124)}\\
&= Q_{\tilde{f}_\varphi^{-1}(W)|X=x}(1-\alpha) && \text{(E.125)}\\
&= \tilde{f}_\varphi^{-1}(Q_{W|X=x}(1-\alpha)), && \text{(E.126)}
\end{aligned}$$

where we used that $\tilde{f}_\varphi$ is increasing and bijective, with $\tilde{f}_\varphi^{-1}(f_\tau(\varphi))=\tau$. In other words, $\tau_\star(x)$ is the $1-\alpha$ quantile of $\tilde{f}_\varphi^{-1}(W)$.

However, in practice, $\tau_\star(x)$ cannot be computed directly because the true conditional distribution $F_{Y|X=x}$ is unknown. Instead, it can be estimated using a sample $\hat{Y}^{(k)},k\in[K]$, drawn from the estimated conditional distribution $\hat{P}_{Y|X=x}$. If $\hat{Q}_{W|X=x}(1-\alpha)$ is the $1-\alpha$ quantile of the empirical distribution $\frac{1}{K}\sum_{k\in[K]}\delta_{s_W(x,\hat{Y}^{(k)})}$, we can compute

$$\hat{\tau}(x)=\tilde{f}_\varphi^{-1}(\hat{Q}_{W|X=x}(1-\alpha)). \tag{E.127}$$

It should be noted that this estimated prediction set loses the exact conditional and marginal coverage properties due to the reliance on the estimated conditional distribution. The following shows how conformal prediction can restore some coverage properties.

From (E.120), using (E.121), we can write

$$\begin{aligned}
\bar{R}_{\mathrm{CP}^2}(x) &= \left\{y\in\mathcal{Y}: s_W(x,y)\leqslant f_{\tau_\star(x)}(\varphi)\right\} && \text{(E.128)}\\
&= \left\{y\in\mathcal{Y}: f_{\tau_\star(x)}^{-1}(s_W(x,y))\leqslant\varphi\right\}, && \text{(E.129)}
\end{aligned}$$

where we used the invertibility of f_τ for any $\tau \in \mathbb{R}$.

Based on (E.129), Plassier et al. (2025a) defined the following conformity score:

$$s_{\mathrm{CP}^2}(x, y) = f^{-1}_{\hat{\tau}(x)}(s_W(x, y)), \tag{E.130}$$

for which the corresponding prediction set $\hat{R}_{\mathrm{CP}^2}$ is given by

$$\hat{R}_{\mathrm{CP}^2}(x) = \{y \in \mathcal{Y} : s_{\mathrm{CP}^2}(x, y) \leqslant \hat{q}\}, \tag{E.131}$$

where we used (2.53) from the main text.

As an example, taking $f_\tau(\lambda) = \tau\lambda$ and $\varphi = 1$, the conformity score becomes:

$$s_{\mathrm{CP}^2}(x, y) = s_W(x, y)/\hat{\tau}(x), \tag{E.132}$$

where $\hat{\tau}(x)$ is defined in (E.127). Finally, we obtain CP^2-PCP simply by replacing s_W with s_{PCP} in (E.132).

E.8.2 Asymptotic properties

Asymptotic equivalence of prediction sets

In the following, we prove that the prediction sets generated by CP^2 (for any f_τ and φ) and CDF-based methods are identical in the oracle setting, asymptotically, as $|\mathcal{D}_{\mathrm{cal}}| \to \infty$. Specifically, for any $x \in \mathcal{X}$, both methods select the same threshold $t_{1-\alpha} = Q_{W|X=x}(1-\alpha)$ for the prediction set $\mathcal{R}(x; t_{1-\alpha})$, which ensures a coverage level of $1 - \alpha$.

Proposition 11. Provided that the assumptions in Section E.8.1 hold, for any $x \in \mathcal{X}$, the prediction sets $\bar{R}_{\mathrm{CP}^2}(x)$ (for any choice of f_τ and φ) and $\bar{R}_{\mathrm{CDF}}(x)$ are equivalent.

Proof. Using the fact that $\tilde{f}_\varphi^{-1}(f_\tau(\varphi)) = \tau$ for any $\tau \in \mathbb{R}$ and that $\tilde{f}_\varphi$ is increasing and bijective, we can write:

$$\begin{aligned}
\bar{R}_{\mathrm{CP}^2}(x) &= \{y \in \mathcal{Y} : s_W(x, y) \leqslant f_{\tau_\star(x)}(\varphi)\} && \text{(E.133)}\\
&= \{y \in \mathcal{Y} : \tilde{f}_\varphi^{-1}(s_W(x, y)) \leqslant \tau_\star(x)\} && \text{(E.134)}\\
&= \{y \in \mathcal{Y} : \tilde{f}_\varphi^{-1}(s_W(x, y)) \leqslant \tilde{f}_\varphi^{-1}(Q_{W|X=x}(1-\alpha))\} && \text{(E.135)}\\
&= \{y \in \mathcal{Y} : s_W(x, y) \leqslant Q_{W|X=x}(1-\alpha)\}. && \text{(E.136)}
\end{aligned}$$

Let $\bar{R}_{\mathrm{CDF}}(x)$ denote the prediction set obtained using the conformity score s_{CDF} as $|\mathcal{D}_{\mathrm{cal}}| \to \infty$. As shown in Section 5.4.1, $s_{\mathrm{CDF}}(X, Y) \sim \mathcal{U}(0, 1)$, which implies $\hat{q} = 1 - \alpha$. Therefore:

$$\begin{aligned}
\bar{R}_{\mathrm{CDF}}(x) &= \{y \in \mathcal{Y} : s_{\mathrm{CDF}}(x, y) \leqslant 1 - \alpha\} && \text{(E.137)}\\
&= \{y \in \mathcal{Y} : F_{W|X=x}(s_W(x, y)) \leqslant 1 - \alpha\} && \text{(E.138)}\\
&= \{y \in \mathcal{Y} : s_W(x, y) \leqslant Q_{W|X=x}(1-\alpha)\}. && \text{(E.139)}
\end{aligned}$$

This shows that $\bar{R}_{\mathrm{CP}^2}(x) = \bar{R}_{\mathrm{CDF}}(x)$ and that the threshold $t_{1-\alpha} = Q_{W|X=x}(1-\alpha)$ is identical for both methods.

□

Asymptotic conditional coverage

Proposition 12. Provided that the assumptions in Section 5.5.2 of the main text hold, specifically that $\hat{F}_{Y|X=x} = F_{Y|X=x}$ for all $x \in \mathcal{X}$, and $|\mathcal{D}_{\text{cal}}| \to \infty$, CP^2 achieves ACC as $K \to \infty$.

Proof. Under these assumptions, we have $\hat{Q}_{W|X=x} = Q_{W|X=x}$, which implies $\hat{\tau}(x) = \tau_\star(x)$ for all $x \in \mathcal{X}$. Hence, the prediction set for CP^2 is given by:

$$\bar{R}_{\text{CP}^2}(x) = \{y \in \mathcal{Y} : s_{\text{CP}^2}(x, y) \leqslant \varphi\}.$$

Since this prediction set provides conditional coverage, it also ensures marginal coverage:

$$\begin{aligned}
\mathbb{P}(Y \in \bar{R}_{\text{CP}^2}(X)) &= \mathbb{P}(s_{\text{CP}^2}(X, Y) \leqslant \varphi) && \text{(E.140)}\\
&= \mathbb{E}_X[\mathbb{P}(s_{\text{CP}^2}(X, Y) \leqslant \varphi \mid X)] && \text{(E.141)}\\
&= \mathbb{E}_X[1 - \alpha] && \text{(E.142)}\\
&= 1 - \alpha. && \text{(E.143)}
\end{aligned}$$

Since $\hat{q}$ is the $1 - \alpha$ quantile of $s_{\text{CP}^2}(X, Y)$, and as $|\mathcal{D}_{\text{cal}}| \to \infty$, we have $\hat{q} = \varphi$ by definition. Therefore, since $\bar{R}_{\text{CP}^2}(x)$ achieves conditional coverage (see (E.122)), the region $\hat{R}_{\text{CP}^2}(x)$ also achieves ACC:

$$\begin{aligned}
\mathbb{P}(Y \in \hat{R}_{\text{CP}^2}(X) \mid X = x) &= \mathbb{P}(s_{\text{CP}^2}(X, Y) \leqslant \hat{q} \mid X = x) && \text{(E.144)}\\
&= \mathbb{P}(s_{\text{CP}^2}(X, Y) \leqslant \varphi \mid X = x) && \text{(E.145)}\\
&\geqslant 1 - \alpha. && \text{(E.146)}
\end{aligned}$$

□

E.8.3 Relationship between CDF-based and CP^2-based methods

A natural question is whether there exists $\{f_\tau\}_{\tau \in \mathbb{R}}$ and $\varphi \in \mathbb{R}$ (with the assumptions introduced in Section E.8.1) such that CDF-based and CP^2-based methods produce the same regions. In the simple case where the distribution of the base conformity score is in a location family, Proposition 13 shows that the two methods are equivalent for a simple choice of f_τ and φ. However, the proof is not easily generalizable to a location-scale family. Further development of existing classes of conformal methods with ACC and their intersections is a promising avenue for future research. Interestingly, we discuss below that answering this question would also draw links between established univariate conformal methods.

Analogy to univariate conformal prediction. To further clarify the distinction between CDF- and CP^2-based methods, we can draw an analogy to the established univariate methods dist-split (DS, Izbicki et al. (2020)) and conformalized quantile regression (CQR, Romano et al. (2019)). Since CDF- and CP^2-based methods calibrate one quantile instead of an interval, we only consider the right-tail version of DS and CQR:

- s_{ECDF} is analogous to DS but operates in the space of conformity instead of the output space $\mathcal{Y}$. DS uses the estimated conditional CDF of the output variable, $s_{\text{DS}}(x, y) = \hat{F}_{Y|X=x}(y)$, transforming y based on its rank.

- s_{CP^2} with difference adjustment is analogous to CQR, and also operates in the space of conformity instead of the output space $\mathcal{Y}$. Note that CP^2 with difference adjustment can be simplified to $s_{\text{CP}^2}(x, y) = s_W(x, y) - \hat{Q}_{W|X=x}(1-\alpha)$. Similarly, CQR uses a score based on the difference from a single estimated conditional quantile, $s_{\text{CQR}}(x, y) = y - \hat{Q}_{Y|X=x}(1-\alpha)$.

Both CDF-based and CP^2-based methods rely on a sample $\{\hat{Y}^{(k)}\}_{k=1}^{K}$ where $\hat{Y}^{(k)} \sim \hat{P}_{Y|X=x}$. The difference lies in the way they transform $s_W(x, y)$ to obtain ACC. Recall that the conformity scores s_{ECDF} and s_{CP^2} are given by

$$s_{\text{ECDF}}(x, y) = \frac{1}{K} \sum_{k \in [K]} \mathbb{1}\left(s_W(x, \hat{Y}^{(k)}) \leqslant s_W(x, y)\right) = \hat{F}_{W|X=x}(s_W(x, y)), \tag{E.147}$$

$$s_{\text{CP}^2}(x, y) = f^{-1}_{\hat{\tau}(x)}(s_W(x, y)) \text{ where } \hat{\tau}(x) = \tilde{f}^{-1}_{\varphi}(\hat{Q}_{W|X=x}(1-\alpha)). \tag{E.148}$$

It is known that two conformal methods produce equal regions if and only if their conformity scores are equal after applying a strictly increasing function $\phi : \mathbb{R} \to \mathbb{R}$, i.e.:

$$s_{\text{ECDF}}(x, y) = \phi(s_{\text{CP}^2}(x, y)) \quad \forall x \in \mathcal{X}, y \in \mathcal{Y}. \tag{E.149}$$

Given $x \in \mathcal{X}$, when K is finite, the conformity score $s_{\text{ECDF}}(x, \cdot)$ is discontinuous and is thus necessarily different from the conformity score $s_{\text{CP}^2}(x, \cdot)$, which is continuous. A more interesting setting is the case where $K \to \infty$ and $s_{\text{ECDF}}(x, \cdot)$ becomes continuous. We define the random variable $\hat{W} = s_W(X, \hat{Y})$, with $\hat{Y} \sim \hat{P}_{Y|X}$. Let $F_{\hat{W}|X=x}$ and $Q_{\hat{W}|X=x}$ denote the conditional CDF and QF of $\hat{W}$ given $X = x$. The conformity scores are defined as follows:

$$\bar{s}_{\text{ECDF}}(x, y) = F_{\hat{W}|X=x}(s_W(x, y)), \tag{E.150}$$

$$\bar{s}_{\text{CP}^2}(x, y) = f^{-1}_{\hat{\tau}(x)}(s_W(x, y)) \text{ where } \hat{\tau}(x) = \tilde{f}^{-1}_{\varphi}(Q_{\hat{W}|X=x}(1-\alpha)). \tag{E.151}$$

Thus, we require that

$$f^{-1}_{\hat{\tau}(x)}(s_W(x, y)) = \phi(F_{\hat{W}|X=x}(s_W(x, y))) \quad \forall x \in \mathcal{X}, y \in \mathcal{Y} \tag{E.152}$$

or equivalently

$$f^{-1}_{\hat{\tau}(x)}(w) = \phi(F_{\hat{W}|X=x}(w)) \quad \forall x \in \mathcal{X}, w \in \mathbb{R}. \tag{E.153}$$

In Proposition 13, we show that, in the particular case where the conditional distributions $\{F_{\hat{W}|X=x}\}_{x \in \mathcal{X}}$ belong to a location family, there exists a simple choice of $\{f_\tau\}_{\tau \in \mathbb{R}}$, $\varphi \in \mathbb{R}$ and strictly increasing $\phi : \mathbb{R} \to \mathbb{R}$ such that the two methods are equivalent.

Proposition 13. Consider a scenario where all conditional distributions $\{F_{\hat{W}|X=x}\}_{x \in \mathcal{X}}$ belong to a location family, i.e.,

$$F_{\hat{W}|X=x}(w) = F(w - \hat{\mu}_x) \text{ and } Q_{\hat{W}|X=x}(\alpha) = F^{-1}(\alpha) + \hat{\mu}_x, \tag{E.154}$$

for some continuous and strictly increasing base CDF F and location parameter $\hat{\mu}_x$. The conformity scores $\bar{s}_{\text{ECDF}}$ and $\bar{s}_{\text{CP}^2}$ lead to the same prediction sets.

Proof. We will show that there is a family of transformations $\{f_\tau\}_{\tau\in\mathbb{R}}$, $\varphi \in \mathbb{R}$ and strictly increasing $\phi : \mathbb{R} \to \mathbb{R}$ with the assumptions above such that, for any $x \in \mathcal{X}$ and $w \in \mathbb{R}$,

$$f^{-1}_{\hat{\tau}(x)}(w) = \phi(F_{\hat{W}|X=x}(w)) \tag{E.155}$$

Define the transformation function f_τ as:

$$f_\tau(\lambda) = F^{-1}(\lambda) + \tau, \tag{E.156}$$

where $\tau > 0$, and define $\varphi = 1 - \alpha$ and $\phi(\lambda) = \lambda$.

The inverse transformations are:

$$f^{-1}_\tau(\lambda) = F(\lambda - \tau), \tag{E.157}$$

and

$$\tilde{f}^{-1}_\varphi(w) = w - F^{-1}(\varphi). \tag{E.158}$$

Now, for $x \in \mathcal{X}$, compute

$$\hat{\tau}(x) = F^{-1}(1-\alpha) + \hat{\mu}_x - F^{-1}(\varphi) = \hat{\mu}_x. \tag{E.159}$$

Finally, we obtain the required equality

$$f^{-1}_{\hat{\tau}(x)}(w) = F(w - \hat{\tau}(x)) = F(w - \hat{\mu}_x) = F_{\hat{W}|X=x}(w). \tag{E.160}$$

□

E.8.4 Empirical comparison

We perform a direct empirical comparison between CDF-based methods (C-PCP, C-HDR) and the corresponding CP^2 methods (CP^2-PCP, CP^2-HPD using both linear (-L) and difference (-D) adjustments from Plassier et al. (2025a)). Figure E.15 shows that:

- C-PCP performs comparably to CP^2-PCP-L (best CP^2 variant for PCP).
- C-HDR performs comparably to CP^2-HPD-D (best CP^2 variant for HPD).
- Other CP^2 variants (CP^2-PCP-D, CP^2-HPD-L) are generally outperformed by their CDF-based version.

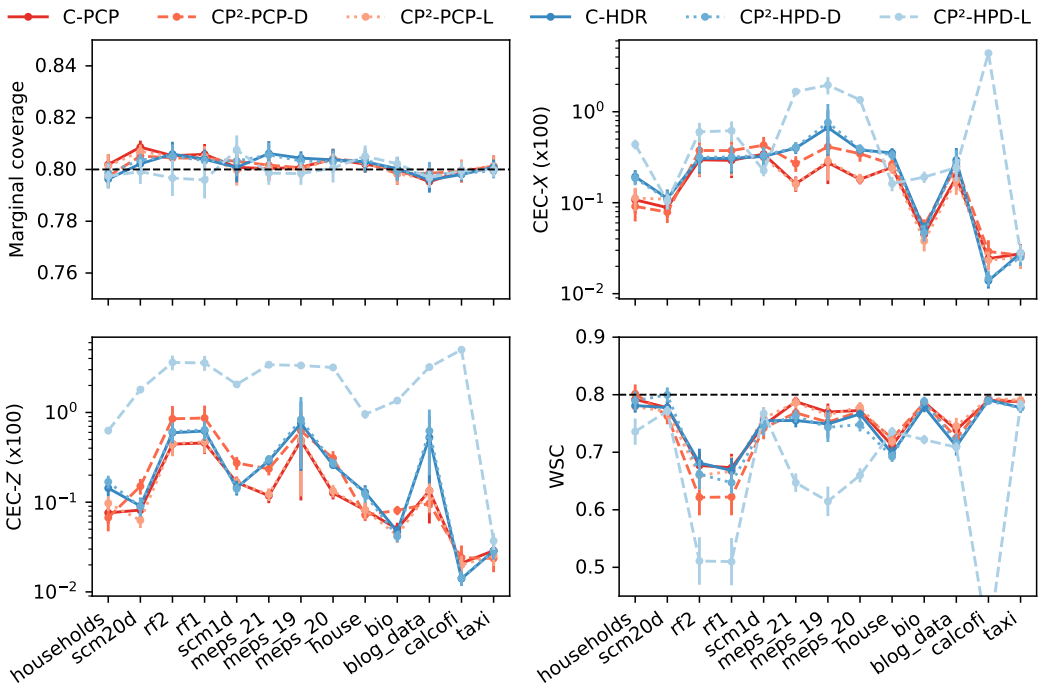


Figure E.15: Comparison of CDF-based methods and CP^2-based methods.

The worsened conditional coverage of CP^2-PCP-D is an interesting observation that was not observed in the smaller scale study of Plassier et al. (2025a). In the case of CP^2-HPD-L, the poor conditional coverage is due to an incompatibility of the linear adjustment function with the (log-scaled) conformity score $s_{\text{DR-CP}}(x, y) = -\log \hat{f}(y|x)$, which can present negative values and thus a decreasing (instead of increasing) adjustment function $f_\tau(\lambda) = \tau\lambda$.

This shows that our simpler s_{ECDF} formulation achieves the same practical benefits as s_{CP^2} without the sensibility of choosing an adjustment function f_τ.

E.9. Full results

Tables E.5 and E.6 show the full results obtained with the setup described in Section 5.6. Each metric is the mean over 10 independent runs. The standard error of the mean is indicated as an index. For each dataset and metric, bold values indicate results statistically similar to the best performer ($\alpha = 0.05$) according to a Z-test.

Table E.5: Full results obtained with the setup described in Section 5.6 (Part 1).

Dataset	Method	MC	Median Size	CEC-X ($\times 100$)	CEC-Z ($\times 100$)	WSC	Test time
households	M-CP	$\mathbf{0.801_{0.0051}}$	$14.2_{0.48}$	$0.340_{0.068}$	$0.364_{0.032}$	$0.779_{0.010}$	$5.69_{0.49}$
	CopulaCPTS	$0.782_{0.0094}$	$12.3_{0.87}$	$0.524_{0.057}$	$0.651_{0.058}$	$0.745_{0.016}$	$8.86_{0.77}$
	DR-CP	$\mathbf{0.802_{0.0046}}$	$13.2_{0.29}$	$0.987_{0.10}$	$1.88_{0.14}$	$0.656_{0.018}$	$0.225_{0.0092}$
	C-HDR	$\mathbf{0.807_{0.0054}}$	$\mathbf{10.6_{0.33}}$	$\mathbf{0.209_{0.039}}$	$0.149_{0.020}$	$\mathbf{0.795_{0.010}}$	$6.12_{0.50}$
	PCP	$\mathbf{0.798_{0.0048}}$	$20.5_{0.38}$	$1.07_{0.085}$	$2.35_{0.15}$	$0.632_{0.015}$	$5.48_{0.46}$
	HD-PCP	$\mathbf{0.800_{0.0043}}$	$15.6_{0.39}$	$0.776_{0.091}$	$1.38_{0.10}$	$0.707_{0.014}$	$5.76_{0.47}$
	STDQR	$\mathbf{0.804_{0.0050}}$	$17.8_{0.41}$	$0.918_{0.073}$	$1.97_{0.098}$	$0.677_{0.019}$	$8.45_{0.79}$
	C-PCP	$\mathbf{0.803_{0.0066}}$	$15.5_{0.74}$	$\mathbf{0.179_{0.045}}$	$\mathbf{0.120_{0.026}}$	$\mathbf{0.800_{0.0061}}$	$11.2_{0.95}$
	L-CP	$\mathbf{0.800_{0.0034}}$	$18.6_{0.80}$	$\mathbf{0.204_{0.040}}$	$\mathbf{0.118_{0.018}}$	$\mathbf{0.788_{0.014}}$	$\mathbf{0.101_{0.0043}}$
scm20d	M-CP	$\mathbf{0.800_{0.0039}}$	$67.6_{8.5}$	$\mathbf{0.0682_{0.011}}$	$0.914_{0.061}$	$0.777_{0.0090}$	$8.08_{0.17}$
	CopulaCPTS	$0.833_{0.0086}$	$1.12\text{e+}02_{2.1\text{e+}01}$	$0.221_{0.063}$	$0.878_{0.043}$	$\mathbf{0.802_{0.012}}$	$10.9_{0.21}$
	DR-CP	$\mathbf{0.799_{0.0048}}$	$2.33\text{e+}02_{2.2\text{e+}01}$	$0.429_{0.044}$	$2.72_{0.16}$	$0.691_{0.018}$	$0.560_{0.025}$
	C-HDR	$\mathbf{0.806_{0.0055}}$	$42.0_{7.9}$	$0.159_{0.024}$	$\mathbf{0.102_{0.017}}$	$\mathbf{0.796_{0.0065}}$	$9.42_{0.18}$
	PCP	$\mathbf{0.798_{0.0051}}$	$1.05\text{e+}02_{1.1\text{e+}01}$	$0.581_{0.045}$	$5.28_{0.23}$	$0.621_{0.016}$	$6.27_{0.39}$
	HD-PCP	$\mathbf{0.799_{0.0049}}$	$94.4_{9.4}$	$0.504_{0.047}$	$4.78_{0.23}$	$0.671_{0.011}$	$7.11_{0.42}$
	STDQR	$\mathbf{0.801_{0.0047}}$	$99.4_{1.0\text{e+}01}$	$0.540_{0.052}$	$4.86_{0.17}$	$0.620_{0.016}$	$8.15_{0.29}$
	C-PCP	$0.809_{0.0038}$	$\mathbf{26.0_{3.1}}$	$0.105_{0.020}$	$\mathbf{0.0896_{0.014}}$	$\mathbf{0.789_{0.0066}}$	$14.3_{0.44}$
	L-CP	$\mathbf{0.796_{0.0035}}$	$72.0_{1.0\text{e+}01}$	$0.166_{0.033}$	$\mathbf{0.0873_{0.018}}$	$\mathbf{0.786_{0.0063}}$	$\mathbf{0.0987_{0.0059}}$
rf1	M-CP	$\mathbf{0.797_{0.0046}}$	$0.00547_{0.0027}$	$0.202_{0.030}$	$0.968_{0.15}$	$0.667_{0.018}$	$20.9_{2.5}$
	CopulaCPTS	$0.785_{0.010}$	$0.00555_{0.0033}$	$0.299_{0.045}$	$1.18_{0.17}$	$0.635_{0.019}$	$29.8_{4.1}$
	DR-CP	$\mathbf{0.799_{0.0028}}$	$0.00215_{0.0010}$	$0.949_{0.21}$	$5.42_{0.70}$	$0.549_{0.038}$	$0.335_{0.016}$
	C-HDR	$\mathbf{0.801_{0.0033}}$	$\mathbf{0.000690_{0.00032}}$	$\mathbf{0.111_{0.036}}$	$0.230_{0.047}$	$\mathbf{0.732_{0.018}}$	$21.7_{2.5}$
	PCP	$\mathbf{0.801_{0.0022}}$	$0.00700_{0.0036}$	$0.863_{0.20}$	$5.95_{0.48}$	$0.538_{0.030}$	$17.9_{2.4}$
	HD-PCP	$\mathbf{0.800_{0.0024}}$	$0.00617_{0.0030}$	$0.776_{0.19}$	$5.58_{0.49}$	$0.563_{0.029}$	$18.4_{2.4}$
	STDQR	$\mathbf{0.800_{0.0032}}$	$0.00624_{0.0031}$	$0.788_{0.19}$	$5.67_{0.50}$	$0.566_{0.025}$	$25.7_{4.1}$
	C-PCP	$\mathbf{0.802_{0.0051}}$	$0.00262_{0.0012}$	$\mathbf{0.0925_{0.016}}$	$\mathbf{0.169_{0.027}}$	$\mathbf{0.732_{0.017}}$	$38.8_{4.9}$
	L-CP	$\mathbf{0.800_{0.0026}}$	$\mathbf{0.00104_{0.00048}}$	$\mathbf{0.107_{0.032}}$	$0.236_{0.042}$	$\mathbf{0.730_{0.0093}}$	$\mathbf{0.0976_{0.0057}}$
scm1d	M-CP	$\mathbf{0.796_{0.0027}}$	$0.528_{0.046}$	$1.02_{0.060}$	$2.42_{0.094}$	$0.636_{0.017}$	$85.6_{2.4\text{e+}01}$
	CopulaCPTS	$0.732_{0.011}$	$0.323_{0.050}$	$1.72_{0.20}$	$3.49_{0.27}$	$0.582_{0.017}$	$87.8_{2.4\text{e+}01}$
	DR-CP	$\mathbf{0.793_{0.0036}}$	$0.867_{0.078}$	$1.50_{0.087}$	$5.17_{0.20}$	$0.559_{0.0097}$	$0.584_{0.025}$
	C-HDR	$0.812_{0.0046}$	$0.239_{0.026}$	$\mathbf{0.452_{0.062}}$	$\mathbf{0.114_{0.015}}$	$\mathbf{0.761_{0.010}}$	$87.0_{2.4\text{e+}01}$
	PCP	$\mathbf{0.795_{0.0054}}$	$0.698_{0.065}$	$1.77_{0.12}$	$8.11_{0.23}$	$0.516_{0.013}$	$5.53_{0.34}$
	HD-PCP	$\mathbf{0.795_{0.0053}}$	$0.684_{0.062}$	$1.75_{0.11}$	$7.96_{0.22}$	$0.530_{0.017}$	$6.41_{0.38}$
	STDQR	$\mathbf{0.795_{0.0064}}$	$0.671_{0.069}$	$1.78_{0.13}$	$8.07_{0.23}$	$0.502_{0.017}$	$6.87_{0.23}$
	C-PCP	$\mathbf{0.803_{0.0053}}$	$\mathbf{0.216_{0.024}}$	$\mathbf{0.456_{0.066}}$	$0.154_{0.028}$	$\mathbf{0.751_{0.0044}}$	$91.1_{2.4\text{e+}01}$
	L-CP	$\mathbf{0.799_{0.0045}}$	$\mathbf{0.197_{0.020}}$	$\mathbf{0.463_{0.059}}$	$\mathbf{0.108_{0.017}}$	$0.731_{0.014}$	$\mathbf{0.102_{0.0056}}$
meps_21	M-CP	$\mathbf{0.800_{0.0051}}$	$0.185_{0.013}$	$0.926_{0.096}$	$0.775_{0.099}$	$0.701_{0.010}$	$1.35\text{e+}02_{1.7\text{e+}01}$
	CopulaCPTS	$0.778_{0.0064}$	$0.171_{0.014}$	$0.957_{0.13}$	$0.693_{0.099}$	$0.684_{0.011}$	$1.61\text{e+}02_{1.9\text{e+}01}$
	DR-CP	$\mathbf{0.803_{0.0023}}$	$0.227_{0.013}$	$3.75_{0.16}$	$4.38_{0.52}$	$0.531_{0.012}$	$0.228_{0.011}$
	C-HDR	$\mathbf{0.807_{0.0046}}$	$\mathbf{0.132_{0.024}}$	$0.437_{0.045}$	$0.260_{0.041}$	$0.745_{0.013}$	$1.35\text{e+}02_{1.7\text{e+}01}$
	PCP	$\mathbf{0.801_{0.0031}}$	$0.359_{0.021}$	$3.17_{0.13}$	$3.75_{0.44}$	$0.550_{0.0078}$	$78.4_{8.3}$
	HD-PCP	$\mathbf{0.802_{0.0024}}$	$0.246_{0.015}$	$2.09_{0.15}$	$2.18_{0.28}$	$0.601_{0.010}$	$78.7_{8.3}$
	STDQR	$\mathbf{0.802_{0.0022}}$	$0.283_{0.015}$	$2.60_{0.12}$	$2.97_{0.36}$	$0.582_{0.011}$	$96.0_{9.9}$
	C-PCP	$\mathbf{0.805_{0.0026}}$	$0.220_{0.021}$	$\mathbf{0.165_{0.044}}$	$\mathbf{0.0851_{0.025}}$	$\mathbf{0.775_{0.0065}}$	$2.13\text{e+}02_{2.3\text{e+}01}$
	L-CP	$\mathbf{0.801_{0.0034}}$	$0.244_{0.052}$	$0.770_{0.13}$	$0.422_{0.11}$	$0.685_{0.026}$	$\mathbf{0.125_{0.0073}}$
meps_19	M-CP	$\mathbf{0.803_{0.0027}}$	$0.214_{0.022}$	$0.702_{0.049}$	$0.622_{0.086}$	$0.709_{0.0095}$	$1.44\text{e+}02_{2.3\text{e+}01}$
	CopulaCPTS	$\mathbf{0.804_{0.022}}$	$0.595_{0.42}$	$1.13_{0.26}$	$0.926_{0.29}$	$0.721_{0.030}$	$1.77\text{e+}02_{2.8\text{e+}01}$
	DR-CP	$0.795_{0.0028}$	$0.175_{0.011}$	$3.91_{0.18}$	$3.98_{0.73}$	$0.501_{0.013}$	$0.224_{0.011}$
	C-HDR	$\mathbf{0.807_{0.0039}}$	$\mathbf{0.119_{0.019}}$	$0.380_{0.036}$	$0.245_{0.039}$	$0.753_{0.013}$	$1.44\text{e+}02_{2.3\text{e+}01}$
	PCP	$0.794_{0.0033}$	$0.396_{0.059}$	$2.95_{0.23}$	$3.51_{0.53}$	$0.542_{0.013}$	$99.9_{1.7\text{e+}01}$
	HD-PCP	$0.796_{0.0032}$	$0.266_{0.033}$	$1.98_{0.14}$	$2.05_{0.35}$	$0.583_{0.0090}$	$1.00\text{e+}02_{1.7\text{e+}01}$
	STDQR	$0.791_{0.0032}$	$0.307_{0.043}$	$2.63_{0.23}$	$2.95_{0.49}$	$0.557_{0.014}$	$1.17\text{e+}02_{1.8\text{e+}01}$
	C-PCP	$0.810_{0.0021}$	$0.238_{0.026}$	$\mathbf{0.128_{0.016}}$	$\mathbf{0.0757_{0.024}}$	$\mathbf{0.797_{0.0088}}$	$2.44\text{e+}02_{3.9\text{e+}01}$
	L-CP	$\mathbf{0.803_{0.0033}}$	$0.232_{0.043}$	$0.679_{0.13}$	$0.415_{0.13}$	$0.702_{0.022}$	$\mathbf{0.123_{0.0069}}$

Table E.6: Full results obtained with the setup described in Section 5.6 (Part 2).

Dataset	Method	MC	Median Size	CEC-X ($\times 100$)	CEC-Z ($\times 100$)	WSC	Test time
meps_20	M-CP	$\mathbf{0.806_{0.0042}}$	$0.371_{0.061}$	$0.868_{0.10}$	$0.455_{0.12}$	$0.702_{0.014}$	$1.93e{+}02_{2.1e+01}$
	CopulaCPTS	$\mathbf{0.794_{0.0091}}$	$0.362_{0.059}$	$0.963_{0.12}$	$0.497_{0.12}$	$0.692_{0.016}$	$2.30e{+}02_{2.4e+01}$
	DR-CP	$\mathbf{0.805_{0.0036}}$	$0.223_{0.020}$	$3.52_{0.11}$	$2.79_{0.76}$	$0.530_{0.0098}$	$0.227_{0.010}$
	C-HDR	$\mathbf{0.805_{0.0044}}$	$\mathbf{0.114_{0.012}}$	$0.439_{0.10}$	$0.122_{0.036}$	$0.745_{0.010}$	$1.94e{+}02_{2.1e+01}$
	PCP	$\mathbf{0.801_{0.0036}}$	$0.535_{0.050}$	$2.83_{0.091}$	$2.42_{0.67}$	$0.544_{0.0088}$	$1.20e{+}02_{1.4e+01}$
	HD-PCP	$\mathbf{0.804_{0.0039}}$	$0.436_{0.066}$	$1.89_{0.13}$	$1.34_{0.37}$	$0.622_{0.012}$	$1.20e{+}02_{1.4e+01}$
	STDQR	$\mathbf{0.803_{0.0048}}$	$0.472_{0.052}$	$2.45_{0.15}$	$1.84_{0.50}$	$0.575_{0.016}$	$1.40e{+}02_{1.6e+01}$
	C-PCP	$\mathbf{0.806_{0.0041}}$	$0.341_{0.039}$	$\mathbf{0.186_{0.061}}$	$\mathbf{0.0484_{0.010}}$	$\mathbf{0.792_{0.013}}$	$3.13e{+}02_{3.1e+01}$
	L-CP	$\mathbf{0.799_{0.0033}}$	$0.280_{0.028}$	$0.662_{0.073}$	$0.259_{0.081}$	$0.703_{0.014}$	$\mathbf{0.127_{0.0062}}$
house	M-CP	$\mathbf{0.801_{0.0023}}$	$1.17_{0.023}$	$0.254_{0.023}$	$0.190_{0.019}$	$0.730_{0.0098}$	$56.0_{3.8e+01}$
	CopulaCPTS	$0.812_{0.0082}$	$1.22_{0.043}$	$0.316_{0.035}$	$0.276_{0.027}$	$\mathbf{0.750_{0.012}}$	$60.7_{3.8e+01}$
	DR-CP	$\mathbf{0.801_{0.0041}}$	$\mathbf{0.664_{0.021}}$	$0.895_{0.045}$	$1.20_{0.073}$	$0.627_{0.011}$	$0.283_{0.011}$
	C-HDR	$0.807_{0.0039}$	$\mathbf{0.651_{0.016}}$	$0.388_{0.026}$	$0.114_{0.013}$	$0.709_{0.010}$	$56.6_{3.8e+01}$
	PCP	$\mathbf{0.801_{0.0026}}$	$0.882_{0.023}$	$0.753_{0.030}$	$1.14_{0.038}$	$0.643_{0.0076}$	$17.6_{0.98}$
	HD-PCP	$\mathbf{0.803_{0.0034}}$	$0.680_{0.018}$	$0.694_{0.033}$	$0.789_{0.035}$	$0.649_{0.0089}$	$18.0_{0.99}$
	STDQR	$\mathbf{0.801_{0.0042}}$	$0.799_{0.023}$	$0.670_{0.022}$	$0.788_{0.038}$	$0.649_{0.0077}$	$19.5_{0.88}$
	C-PCP	$0.809_{0.0030}$	$0.858_{0.018}$	$0.275_{0.026}$	$0.0831_{0.011}$	$0.729_{0.0091}$	$73.7_{3.8e+01}$
	L-CP	$\mathbf{0.802_{0.0035}}$	$1.19_{0.017}$	$\mathbf{0.174_{0.020}}$	$\mathbf{0.0542_{0.0079}}$	$\mathbf{0.756_{0.0090}}$	$\mathbf{0.146_{0.0067}}$
bio	M-CP	$0.809_{0.0021}$	$0.303_{0.0066}$	$0.137_{0.0093}$	$0.253_{0.013}$	$0.764_{0.0055}$	$1.27e{+}02_{6.0}$
	CopulaCPTS	$\mathbf{0.800_{0.0045}}$	$0.296_{0.0092}$	$0.137_{0.0083}$	$0.260_{0.015}$	$0.751_{0.0068}$	$1.45e{+}02_{7.1}$
	DR-CP	$\mathbf{0.805_{0.0020}}$	$0.257_{0.0067}$	$0.507_{0.028}$	$1.14_{0.034}$	$0.646_{0.0066}$	$0.511_{0.020}$
	C-HDR	$0.808_{0.0015}$	$\mathbf{0.218_{0.0053}}$	$0.0372_{0.0073}$	$0.0360_{0.0056}$	$\mathbf{0.794_{0.0054}}$	$1.29e{+}02_{6.0}$
	PCP	$\mathbf{0.802_{0.0021}}$	$0.343_{0.0076}$	$0.567_{0.029}$	$1.32_{0.023}$	$0.628_{0.0052}$	$1.27e{+}02_{6.1}$
	HD-PCP	$\mathbf{0.804_{0.0016}}$	$0.259_{0.0065}$	$0.352_{0.020}$	$0.803_{0.020}$	$0.673_{0.0043}$	$1.27e{+}02_{6.1}$
	STDQR	$\mathbf{0.803_{0.0024}}$	$0.269_{0.0067}$	$0.389_{0.019}$	$0.912_{0.036}$	$0.667_{0.0058}$	$86.6_{6.5}$
	C-PCP	$0.810_{0.0029}$	$0.302_{0.0074}$	$0.0369_{0.0063}$	$0.0404_{0.0069}$	$\mathbf{0.798_{0.0052}}$	$2.54e{+}02_{1.2e+01}$
	L-CP	$\mathbf{0.805_{0.00093}}$	$0.267_{0.0061}$	$\mathbf{0.0203_{0.0045}}$	$\mathbf{0.0198_{0.0021}}$	$0.789_{0.0039}$	$\mathbf{0.251_{0.013}}$
blog_data	M-CP	$\mathbf{0.802_{0.0049}}$	$0.170_{0.039}$	$0.292_{0.051}$	$0.153_{0.072}$	$0.736_{0.012}$	$5.06e{+}03_{7.0e+02}$
	CopulaCPTS	$0.813_{0.0078}$	$0.0948_{0.015}$	$0.313_{0.050}$	$0.231_{0.063}$	$0.742_{0.010}$	$5.13e{+}03_{7.1e+02}$
	DR-CP	$0.808_{0.0014}$	$0.0374_{0.0056}$	$1.06_{0.098}$	$1.50_{0.43}$	$0.644_{0.0059}$	$0.515_{0.026}$
	C-HDR	$0.809_{0.0030}$	$\mathbf{0.0155_{0.0031}}$	$0.237_{0.068}$	$\mathbf{0.0611_{0.019}}$	$\mathbf{0.751_{0.013}}$	$5.06e{+}03_{7.0e+02}$
	PCP	$\mathbf{0.801_{0.0033}}$	$0.141_{0.023}$	$0.938_{0.081}$	$1.52_{0.36}$	$0.643_{0.0052}$	$5.74e{+}02_{7.9e+01}$
	HD-PCP	$\mathbf{0.803_{0.0038}}$	$0.125_{0.023}$	$0.794_{0.075}$	$0.945_{0.22}$	$0.660_{0.0080}$	$5.75e{+}02_{7.9e+01}$
	STDQR	$\mathbf{0.810_{0.0072}}$	$0.163_{0.036}$	$0.805_{0.074}$	$0.881_{0.18}$	$0.678_{0.012}$	$5.84e{+}02_{8.0e+01}$
	C-PCP	$\mathbf{0.804_{0.0045}}$	$0.106_{0.021}$	$\mathbf{0.163_{0.049}}$	$0.113_{0.056}$	$\mathbf{0.764_{0.012}}$	$5.63e{+}03_{7.3e+02}$
	L-CP	$\mathbf{0.801_{0.0023}}$	$0.0676_{0.017}$	$0.327_{0.088}$	$\mathbf{0.0624_{0.023}}$	$0.722_{0.012}$	$\mathbf{0.258_{0.012}}$
calcofi	M-CP	$\mathbf{0.803_{0.0023}}$	$2.13_{0.024}$	$0.433_{0.015}$	$0.446_{0.016}$	$0.734_{0.0069}$	$26.4_{1.1}$
	CopulaCPTS	$0.815_{0.0075}$	$2.38_{0.12}$	$0.480_{0.048}$	$0.492_{0.048}$	$0.746_{0.0096}$	$29.6_{1.2}$
	DR-CP	$0.805_{0.0027}$	$\mathbf{1.67_{0.022}}$	$1.44_{0.040}$	$1.56_{0.039}$	$0.654_{0.0061}$	$0.529_{0.023}$
	C-HDR	$0.805_{0.0018}$	$1.99_{0.026}$	$\mathbf{0.0294_{0.012}}$	$\mathbf{0.0187_{0.0037}}$	$0.794_{0.0053}$	$27.7_{1.2}$
	PCP	$\mathbf{0.802_{0.0026}}$	$2.33_{0.029}$	$1.64_{0.042}$	$1.79_{0.041}$	$0.638_{0.0034}$	$26.5_{1.2}$
	HD-PCP	$\mathbf{0.802_{0.0033}}$	$1.89_{0.029}$	$0.980_{0.033}$	$1.05_{0.030}$	$0.683_{0.0050}$	$27.3_{1.2}$
	STDQR	$\mathbf{0.799_{0.0034}}$	$1.97_{0.021}$	$1.13_{0.031}$	$1.23_{0.033}$	$0.676_{0.0080}$	$26.4_{0.99}$
	C-PCP	$0.809_{0.0030}$	$2.81_{0.042}$	$\mathbf{0.0332_{0.0093}}$	$0.0253_{0.0048}$	$\mathbf{0.806_{0.0050}}$	$52.9_{2.3}$
	L-CP	$\mathbf{0.800_{0.0020}}$	$2.70_{0.024}$	$\mathbf{0.0332_{0.019}}$	$\mathbf{0.0179_{0.0040}}$	$0.792_{0.0035}$	$\mathbf{0.264_{0.012}}$
taxi	M-CP	$\mathbf{0.802_{0.0032}}$	$4.26_{0.068}$	$0.0585_{0.0034}$	$0.0421_{0.0058}$	$0.784_{0.0052}$	$60.6_{8.6}$
	CopulaCPTS	$0.822_{0.0040}$	$4.72_{0.11}$	$0.114_{0.018}$	$0.0989_{0.019}$	$\mathbf{0.799_{0.0050}}$	$68.4_{9.7}$
	DR-CP	$\mathbf{0.805_{0.0024}}$	$\mathbf{2.62_{0.029}}$	$0.383_{0.016}$	$0.451_{0.024}$	$0.707_{0.0048}$	$0.539_{0.031}$
	C-HDR	$0.809_{0.0030}$	$\mathbf{2.62_{0.033}}$	$0.0388_{0.0049}$	$0.0441_{0.0053}$	$\mathbf{0.793_{0.0040}}$	$61.9_{8.6}$
	PCP	$\mathbf{0.804_{0.0016}}$	$4.03_{0.040}$	$0.341_{0.022}$	$0.399_{0.025}$	$0.715_{0.0042}$	$60.2_{8.5}$
	HD-PCP	$\mathbf{0.805_{0.0018}}$	$3.18_{0.030}$	$0.194_{0.012}$	$0.219_{0.012}$	$0.750_{0.0055}$	$61.1_{8.5}$
	STDQR	$\mathbf{0.805_{0.0035}}$	$3.63_{0.058}$	$0.203_{0.011}$	$0.224_{0.013}$	$0.748_{0.0080}$	$32.4_{1.1}$
	C-PCP	$\mathbf{0.807_{0.0026}}$	$4.02_{0.064}$	$\mathbf{0.0307_{0.0053}}$	$0.0338_{0.0048}$	$\mathbf{0.802_{0.0050}}$	$1.21e{+}02_{1.7e+01}$
	L-CP	$\mathbf{0.805_{0.0033}}$	$4.94_{0.12}$	$\mathbf{0.0264_{0.0030}}$	$\mathbf{0.0196_{0.0035}}$	$\mathbf{0.796_{0.0046}}$	$\mathbf{0.243_{0.012}}$

APPENDIX F

Supplementary Material for Section 5.7

F.1. Experimental setup

Each model mentioned is trained by minimizing the NLL across training sequences contained in $\mathcal{D}_{\text{train}}$. For optimization, we use mini-batch gradient descent with the Adam optimizer (Kingma and Ba, 2015) and a learning rate of $\eta = 10^{-3}$. The models are trained for at most 500 epochs, and training is interrupted through an early-stopping procedure if there is no improvement in NLL on the validation dataset $\mathcal{D}_{\text{val}}$ for 100 consecutive epochs. In such instances, the model's parameters revert to the state where the validation loss was lowest.

To compute individual prediction sets for the arrival time and the mark, we need to compute the predictive marginals, $\hat{f}_{\tau|X=\boldsymbol{h}}(\tau)$ and $\hat{p}_{k|X=\boldsymbol{h}}(k)$, respectively. To derive $\hat{f}_{\tau|X=\boldsymbol{h}}(\tau)$, we sum over the joint density for each mark, as follows: $\hat{f}_{\tau|X=\boldsymbol{h}}(\tau) = \sum_{k=1}^{K} \hat{f}(\tau, k|\boldsymbol{h})$. Meanwhile, $\hat{p}_{k|X=\boldsymbol{h}}(k)$ is approximated through integration over the positive real line:

$$\hat{p}_{k|X=\boldsymbol{h}}(k) = \int_{\mathbb{R}^+} \hat{f}(\tau, k|\boldsymbol{h}) d\tau = \mathbb{E}_{\tau}\big[\hat{p}_{k|\tau,X=\boldsymbol{h}}(k)\big] \simeq \frac{1}{L} \sum_{i=1}^{L} \hat{p}_{k|\tau_i,X=\boldsymbol{h}}(k), \tag{F.1}$$

where $L = 100$ samples τ_i are generated from $\hat{f}_{\tau|X=\boldsymbol{h}}$. This sampling is achieved with the inverse transform sampling method using a binary search algorithm.

F.2. Results on independent regions for the time and mark

F.2.1 Methods

In this section, we present methods to create univariate prediction sets for the event arrival time and the event mark of new test inputs. This is achieved through the application of conformal regression and classification methods. We also consider heuristic versions, which correspond to non-conformal versions of these methods, by simply replacing the model estimate in the corresponding oracle prediction set. We provide a summary of these methods below.

Prediction sets for the event arrival time. We explore methods to generate a prediction set for τ_{n+1} of a test input $\boldsymbol{h}_{n+1}$, targeting marginal coverage $1-\alpha$. We consider three main families of conformity scores based on quantile regression and density estimation:

$$s_{\text{C-QR}}(\boldsymbol{h}, \tau) = \max\{\hat{Q}_{\tau|X=\boldsymbol{h}}(\alpha/2) - \tau, \tau - \hat{Q}_{\tau|X=\boldsymbol{h}}(1-\alpha/2)\}, \tag{F.2}$$

$$s_{\text{C-QRL}}(\boldsymbol{h}, \tau) = \tau - \hat{Q}_{\tau|X=\boldsymbol{h}}(1-\alpha), \tag{F.3}$$

$$s_{\text{C-HDR}}(\boldsymbol{h}, \tau) = \text{HPD}_{\hat{f}_{\tau|X=\boldsymbol{h}}}(\tau). \tag{F.4}$$

Heuristic methods (H-QR, H-QRL, H-HDR) form prediction sets by setting a score-specific threshold designed to achieve $1-\alpha$ coverage under the estimated model. For H-QR and H-QRL, this yields the intervals $[\hat{Q}_{\tau|X=\boldsymbol{h}_{n+1}}(\alpha/2), \hat{Q}_{\tau|X=\boldsymbol{h}_{n+1}}(1-\alpha/2)]$ and $[0, \hat{Q}_{\tau|X=\boldsymbol{h}_{n+1}}(1-\alpha)]$, respectively. For H-HDR, the set is $\{\tau \in \mathbb{R}^+ : \hat{f}_{\tau|X=\boldsymbol{h}_{n+1}}(\tau) \geqslant z_{1-\alpha}\}$, where $z_{1-\alpha}$ is chosen such that the set has probability mass $1-\alpha$ under $\hat{f}_{\tau|X=\boldsymbol{h}_{n+1}}$.

Their conformal counterparts (C-QR, C-QRL, and C-HDR) apply the SCP algorithm from Section 2.5.1 to these same scores. The resulting prediction sets are of the form $\{\tau \in \mathbb{R}^+ : s(\boldsymbol{h}_{n+1}, \tau) \leqslant \hat{q}\}$, where $\hat{q}$ is the calibrated quantile. Additionally, we analyze *C-CONST*, a simple conformal baseline whose conformity score $s_{\text{C}}(\boldsymbol{h}, \tau) = \tau$ produces a model-agnostic prediction interval $[0, \hat{q}]$.

Prediction sets for the event mark. We explore various methods to generate a prediction set for k_{n+1} given a test input $\boldsymbol{h}_{n+1}$. We consider conformity scores based on cumulative probabilities, including the APS score and its regularized RAPS variant from (5.15):

$$s_{\text{APS}}(\boldsymbol{h}, k) = \sum_{k' : \hat{p}_{k'|X=\boldsymbol{h}}(k) \geqslant \hat{p}_{k|X=\boldsymbol{h}}(k)} \hat{p}_{k|X=\boldsymbol{h}}(k'). \tag{F.5}$$

Heuristic methods (H-APS and H-RAPS) form prediction sets by comparing these scores to a fixed threshold of $1-\alpha$, yielding sets of the form $\{k' \in [K] : s(\boldsymbol{h}_{n+1}, k') \leqslant 1-\alpha\}$.

Their conformal counterparts (C-APS and C-RAPS) apply the SCP algorithm to find a calibrated threshold $\hat{q}$ and form prediction sets $\{k' \in [K] : s(\boldsymbol{h}_{n+1}, k') \leqslant \hat{q}\}$. Additionally, we explore the *C-PROB* conformal baseline, which uses the score $s_{\text{C-PROB}}(\boldsymbol{h}, k) = 1 - \hat{p}_{k|X=\boldsymbol{h}}(k)$. For all mark prediction methods considered, the mark with the highest estimated probability is always included in the set to avoid generating empty predictions.

F.2.2 Experiments

We detail the results for individual prediction sets for the arrival time and the mark in Section F.2.2, respectively. Our primary focus is on a probability miscoverage level of $\alpha = 0.2$.

Prediction sets for the arrival time

In Figure F.1, the results are systematically organized in a table where each row represents a specific metric, and each column corresponds to one of the datasets. This figure gives the results for various methods that are used to generate prediction sets solely for the arrival time. Each heuristic method and its corresponding conformal counterpart are represented in matching colors. To differentiate them, the heuristic methods are marked with hatching patterns.

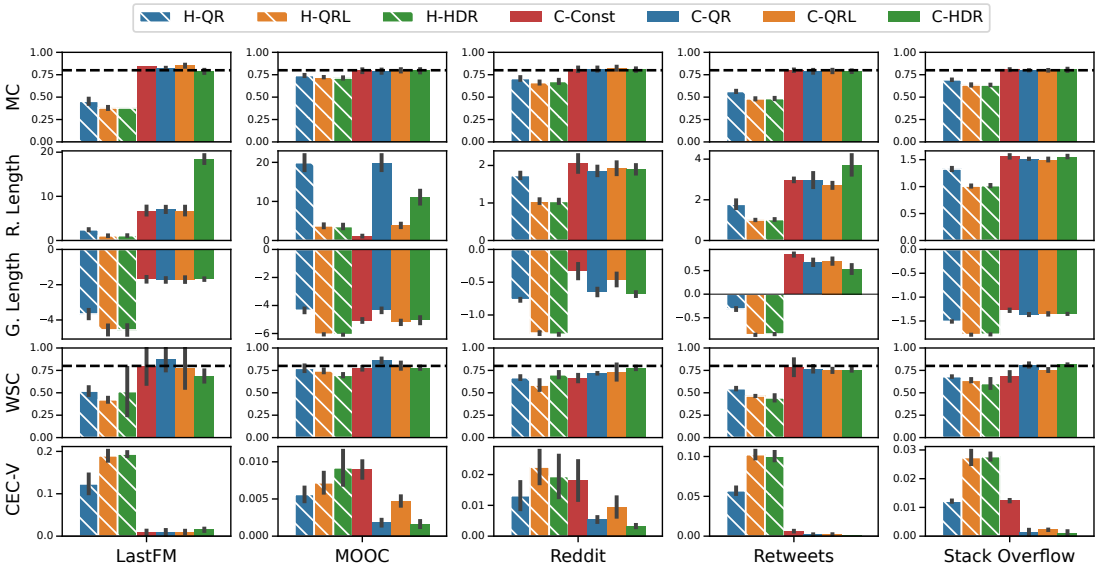


Figure F.1: Performance of different methods producing a region for the time on real-world datasets using the CLNM model. Heuristic methods are hatched.

The first row of the figure demonstrates that all CP methods attain the desired marginal coverage. In contrast, heuristic methods generally undercover, which aligns with expectations. The second row focuses on the average length of the prediction sets. Here, it is evident that heuristic methods generate smaller regions compared to their conformal counterparts. While this might seem beneficial, it is important to note that these smaller regions result from undercoverage, which diminishes their practical utility.

Among the heuristic methods, H-HDR consistently produces regions of smaller or equal lengths compared to H-QR and H-QRL for each prediction instance. Consequently, H-HDR emerges as the method with the smallest average region length. H-QR, not adjusting adequately to the right-skewed nature of the distributions, tends to yield larger regions.

Focusing now on the conformal methods, we exclude heuristic methods from this analysis due to their inability to achieve marginal coverage, which can lead to arbitrarily small regions. In the second row, the variations in average region length among CP methods differ across datasets. Notably, C-HDR, unlike its heuristic counterpart H-HDR, often yields larger average region lengths, especially in the LastFM, MOOC, and Retweets datasets. This difference arises because C-HDR adjusts the initial H-HDR prediction sets adaptively based on the individual predictive distributions. In contrast, C-QR and C-QRL modify their respective heuristic initial regions by a constant amount. While C-Const generates identical regions regardless of the history $\boldsymbol{h}$, it occasionally has the smallest average region length while still maintaining marginal coverage. This occurs because C-Const does not tailor its regions to account for extreme right-skewed distributions, leading to regions that are either slightly larger or significantly smaller compared to other conformal methods. These two phenomena are exemplified in a toy example shown in Figure F.2.

This figure demonstrates a scenario where the average region length of C-HDR is larger than that

of other conformal methods in inter-arrival time prediction. The first row shows predictive distributions in blue and their corresponding realizations as dashed lines, based on three observations from a calibration dataset. In the second row, the prediction sets for seven methods are depicted with $\alpha = 0.5$. All heuristic methods underperform, achieving a maximum coverage of only $1/3$, which is less than the desired coverage of 0.5. Conformal prediction methods, in response, adjust their prediction sets to achieve coverage in at least two out of three cases. Despite H-HDR always producing shorter or equivalent lengths compared to H-QR and H-QRL, C-HDR generates larger regions on average than other conformal methods. Again, C-Const, which does not adapt to individual predictive distributions, presents the smallest average regions among the conformal methods in this particular example. C-Const however does not achieve conditional coverage even asymptotically.

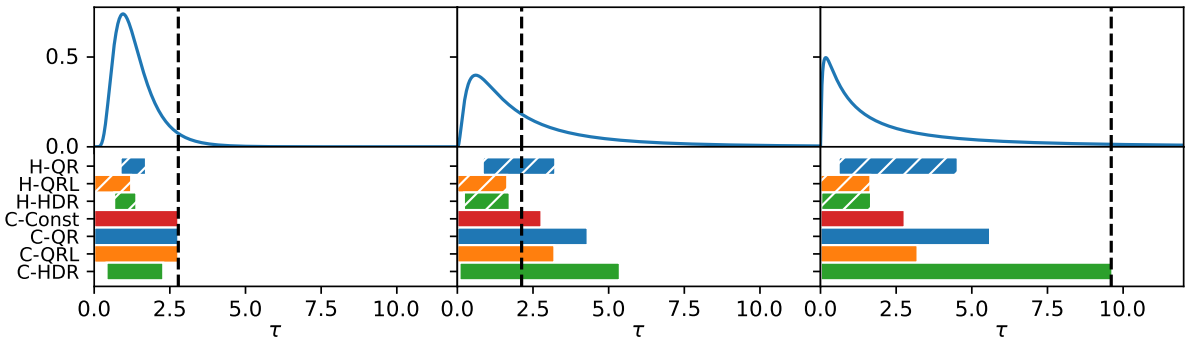


Figure F.2: The figure showcases predictive distributions (blue) and realizations (dashed lines) in the first row, based on a calibration dataset. The second row illustrates prediction sets for various methods with $\alpha = 0.5$. It highlights the undercoverage of heuristic methods, the adaptive adjustments of conformal methods, and the notable differences between C-HDR and other methods in terms of set size. We provide an additional example with $\alpha = 0.2$ in Section F.3.4.

Returning to Figure F.1, the third row introduces an alternative aggregation method for region lengths – the geometric mean. This method assigns less weight to larger regions and more to smaller ones. Here, C-HDR's performance is more in line with other conformal methods, indicating that average region length might not be a reliable metric, particularly in cases of high variability in conditional distributions.

The fourth and fifth rows of the figure assess conditional coverage. WSC denotes coverage over the worst slab, with methods closer to $1 - \alpha$ being preferable, whereas CCE represents a conditional coverage error, which should be minimized. Conformal methods, already proficient in achieving marginal coverage, exhibit a conditional coverage that is usually better than heuristic methods based on the evaluated metrics. Methods capable of tailoring prediction sets to specific instances are expected to exhibit enhanced conditional coverage. Although the WSC metric reveals no marked distinction among conformal methods, the CCE metric shows that C-HDR frequently attains one of the highest levels of conditional coverage. Moreover, C-QR often outperforms C-QRL in conditional coverage. As anticipated, the CCE metric reveals that C-Const generally exhibits the poorest conditional coverage, attributable to its lack of adaptability.

Prediction sets for the mark

Figure F.3 presents similar metrics to those in Figure F.1, but focuses on methods that generate prediction sets exclusively for the mark. Here, the heuristic methods H-APS and H-RAPS already meet the marginal coverage criteria, meaning that conformal prediction primarily offers theoretical backing rather than significant changes in predictions.

Turning our attention to the conformal methods, these methods show similar region lengths across all datasets, with the exception of Reddit, where C-PROB exhibits smaller region lengths. However, on this same dataset and on Stack Overflow, C-PROB has a poor conditional coverage compared to both other conformal methods and heuristic methods. This reflects similar findings discussed in Section Section F.2.2, where the method C-Const manages to attain short prediction sets, albeit with weak conditional coverage. This is explained due to the fact that, in contrast to C-APS, C-PROB does not achieve conditional coverage asymptotically.

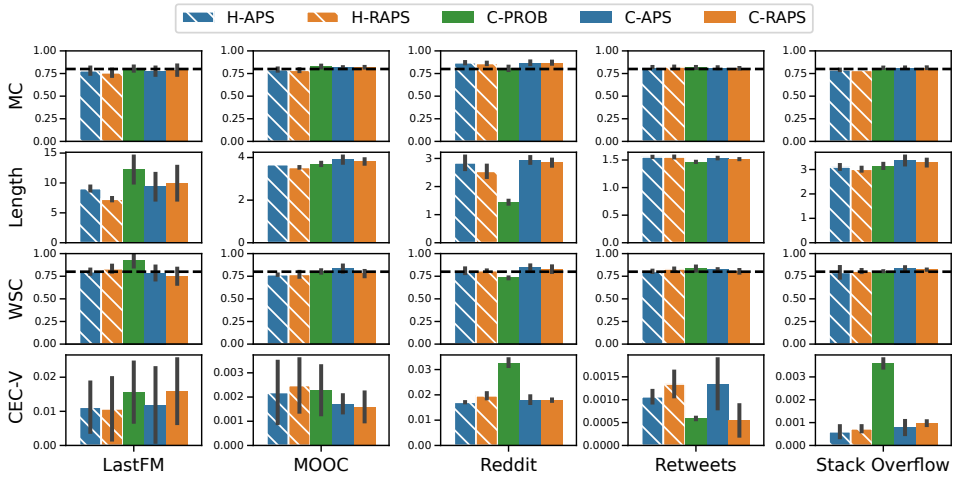


Figure F.3: Performance of different methods producing a region for the mark on real-world datasets using the CLNM model. Heuristic methods are hatched.

F.3. Additional Results

F.3.1 Results on Other Neural TPP Models

In Section 5.7.4, we provided results for the CLNM neural TPP model. In this section, we present additional findings for the FNN, RMTPP, and SAHP models, on the datasets discussed in the main text. The architecture of these neural TPP models is detailed in Dheur et al. (2024). Across all these models, our conclusions align with those outlined in Section 5.7.4, applicable to all scenarios considered, namely, predictions for the time, the mark, or joint predictions on the time and mark.

FNN

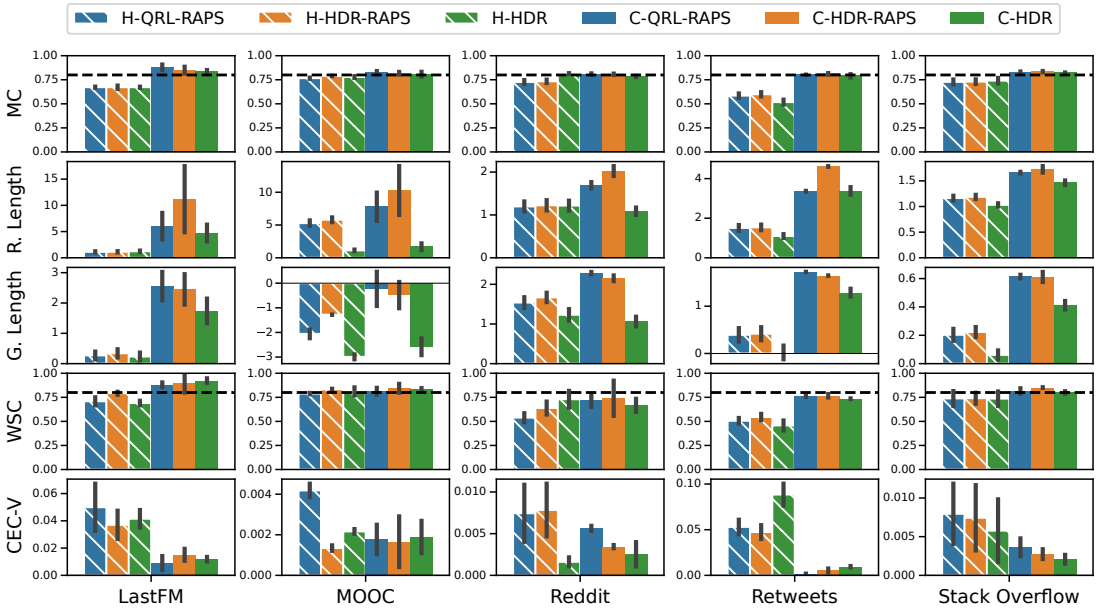


Figure F.4: Performance of different methods producing a joint region for the time and mark on real-world datasets using the FNN model.

RMTPP

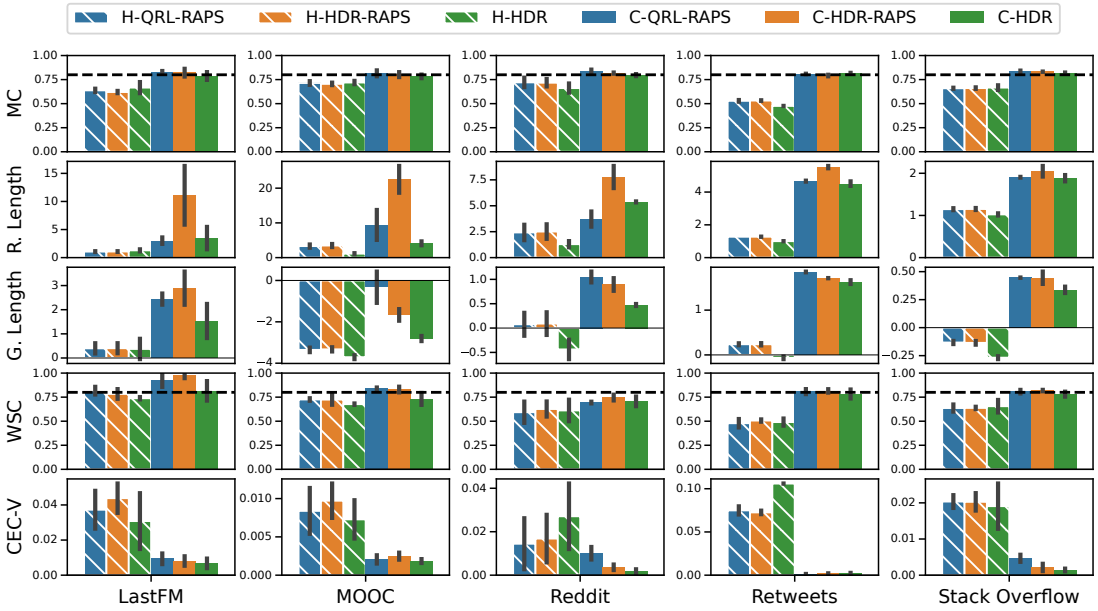


Figure F.5: Performance of different methods producing a joint region for the time and mark on real-world datasets using the RMTPP model.

SAHP

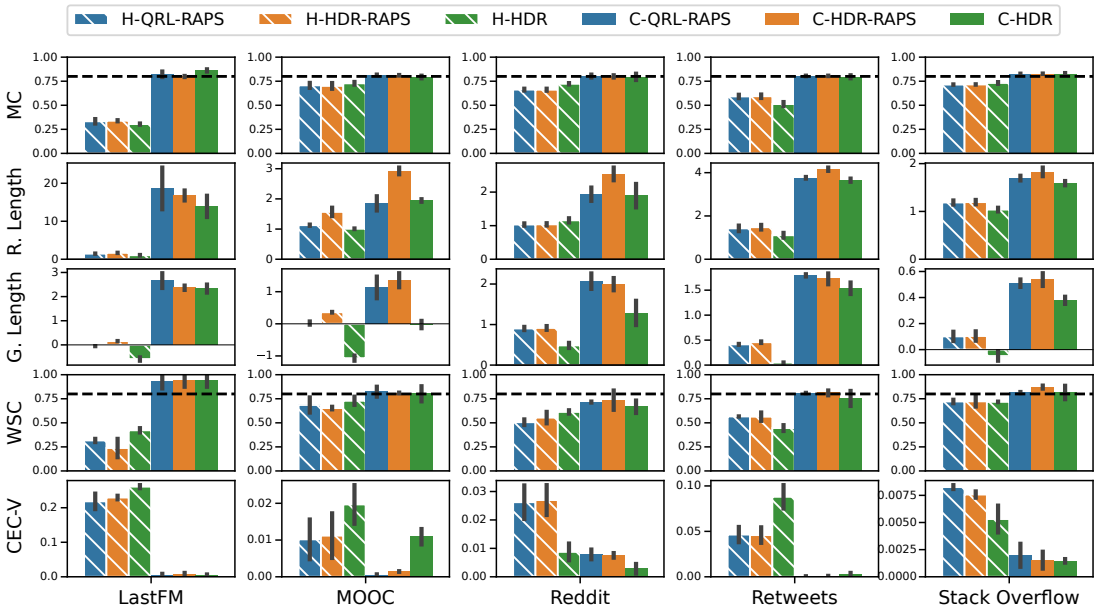


Figure F.6: Performance of different methods producing a joint region for the time and mark on real-world datasets using the SAHP model.

F.3.2 Results on other real-world and synthetic datasets

In this section, we report the results on the Github, MIMIC2, Wikipedia, and Hawkes datasets for all models and all scenarios. We note that the findings on these datasets are also generally consistent with our conclusions from Section 5.7.4. Nonetheless, we usually observe a large variability in the results for Github, MIMIC2, and Wikipedia, explained by the few number of observations in the calibration and test sequences. We therefore invite the reader to exercise caution when interpreting the findings on these real-world datasets.

Finally, for the Hawkes dataset, we observe that heuristic methods tend to already attain the desired coverage level. This finding may be explained by a too simplistic underlying generative Hawkes process, which is already well fitted by the MTPP models. We plan to investigate more complex simulated point processes as part of our future work.

CLNM

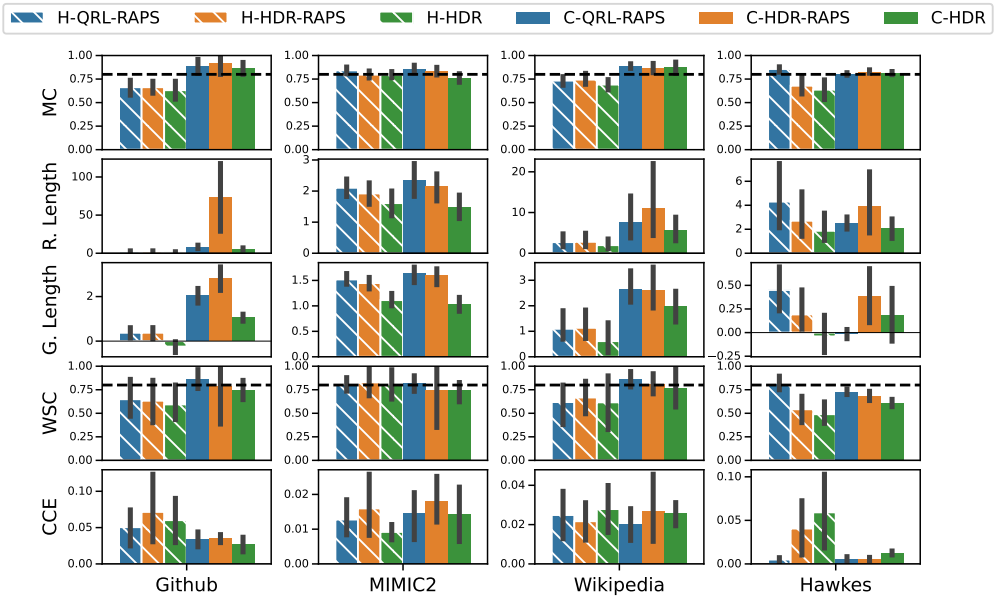


Figure F.7: Performance of different methods producing a joint region for the time and mark on the datasets not discussed in the main text using the CLNM model.

FNN

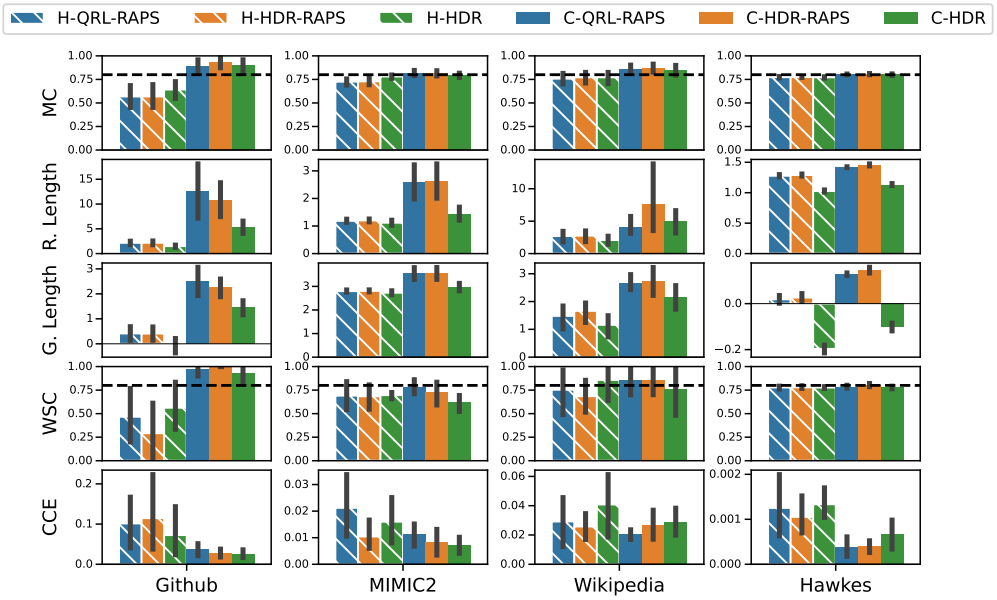


Figure F.8: Performance of different methods producing a joint region for the time and mark on the datasets not discussed in the main text using the FNN model.

RMTPP

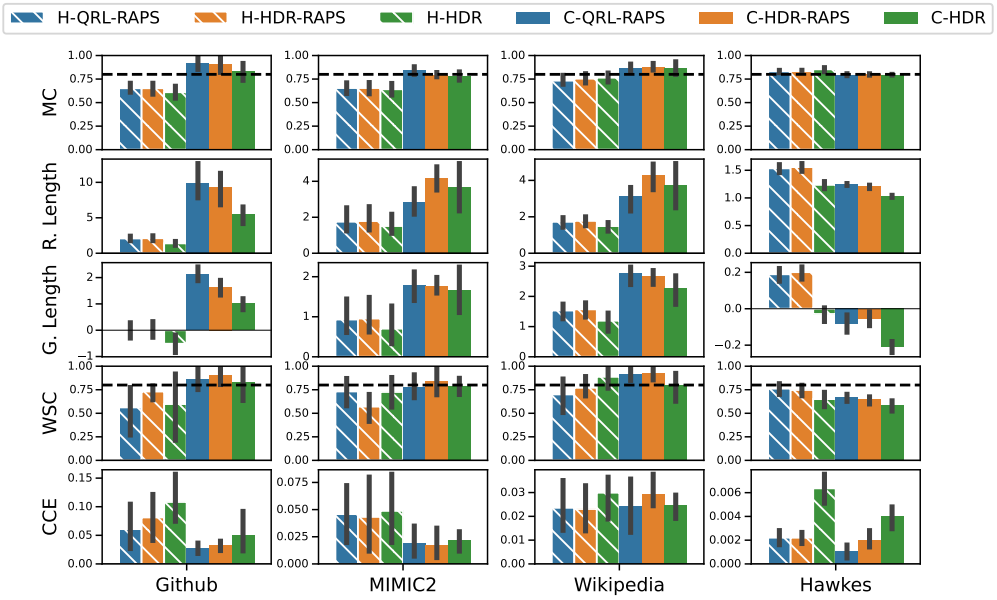


Figure F.9: Performance of different methods producing a joint region for the time and mark on the datasets not discussed in the main text using the RMTPP model.

SAHP

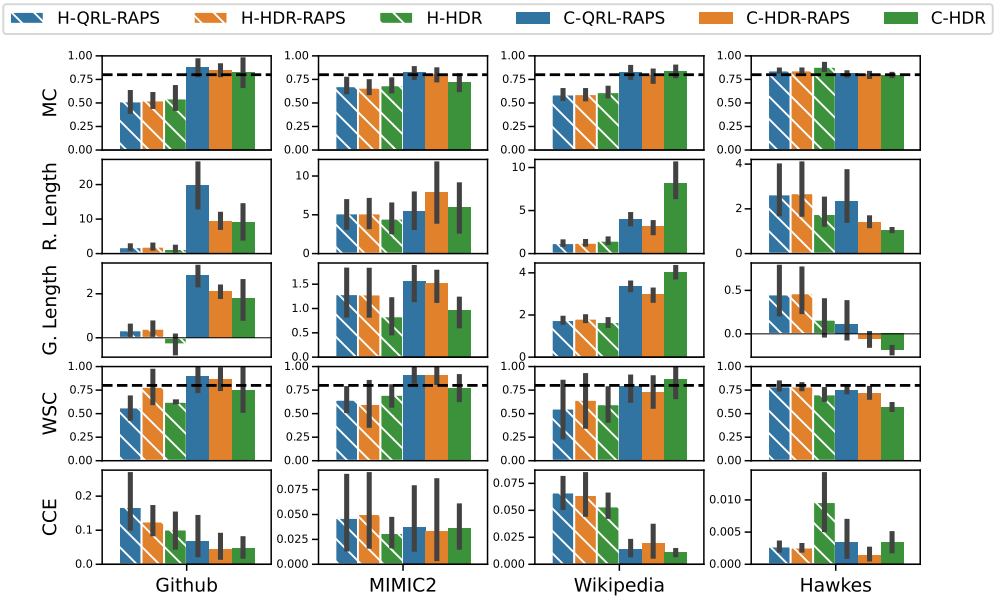


Figure F.10: Performance of different methods producing a joint region for the time and mark on the datasets not discussed in the main text using the SAHP model.

F.3.3 Coverage per level

The last paragraph of Section 5.7.4 discussed the empirical marginal coverage obtained at different coverage levels for methods that generate a joint prediction set on the arrival time and mark. In this section, we present additional results for methods that generate a prediction set individually for either the time or mark.

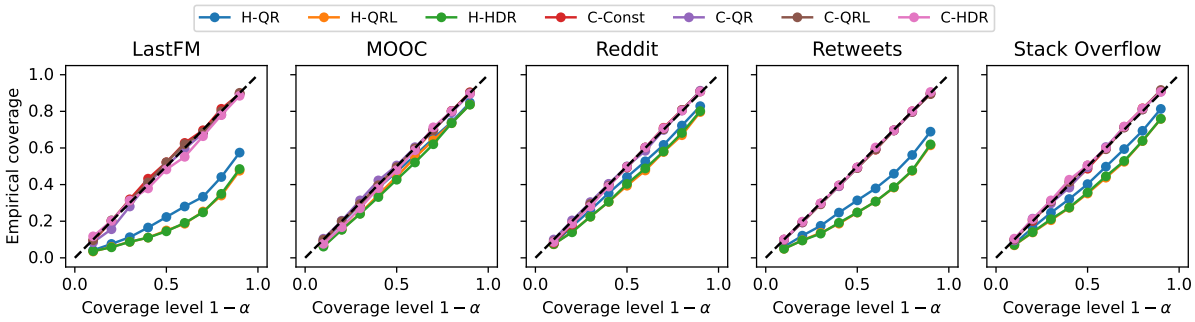


Figure F.11: Empirical marginal coverage for different coverage levels for methods that produce a prediction set for the time with the CLNM model.

Figure F.11 shows that conformal methods for the time attain the desired coverage at all levels while heuristic methods generally undercover. This is expected and mirrors the observations in Section 5.7.4.

Figure F.12 shows that all methods, either heuristic or conformal, overcover for small coverage levels, while coverage is attained for high coverage levels. The reason is that all methods that generate a prediction set for the mark guarantee that prediction sets are not empty by always adding the class with the highest probability. We do not observe overcoverage for high coverage levels because the class with highest probability will almost always be included. However, for low coverage levels, prediction sets that would normally be empty now include the mark with the highest probability, which leads to increased coverage.

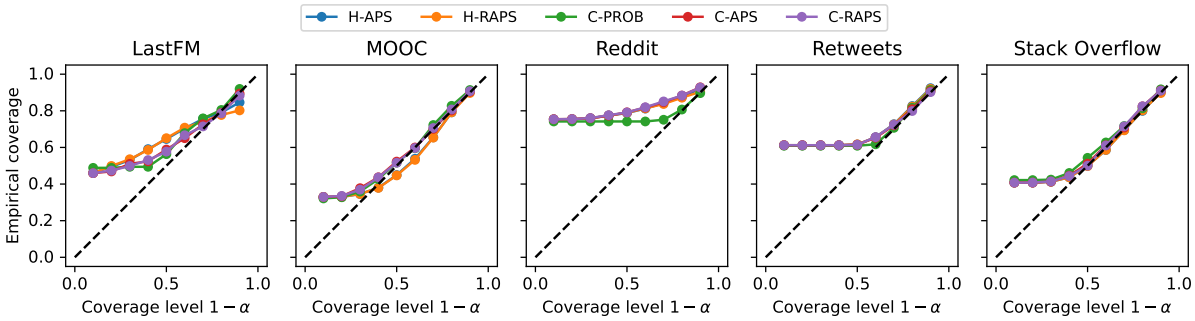


Figure F.12: Empirical marginal coverage for different coverage levels for methods that produce a prediction set for the mark with the CLNM model.

F.3.4 Additional example

In Figure F.2 in Section F.2.2, we presented an example illustrating prediction sets for the time for seven methods with $\alpha = 0.5$. For completeness, we provide an additional toy example with $\alpha = 0.2$ and a calibration dataset of 6 data points in Figure F.13. As in Figure F.2, the heuristic methods undercover, achieving a maximum coverage of $4/6$, which is less than the desired coverage of 0.8. Notably, H-QRL and H-HDR produce exactly the same prediction sets because the densities are decreasing in this case. Conformal methods adjust the prediction sets to achieve coverage in at least five out of six cases. Similarly to Figure F.2, C-HDR generates larger regions on average than other conformal methods despite H-HDR always producing shorter or equivalent lengths compared to H-QR and H-QRL.

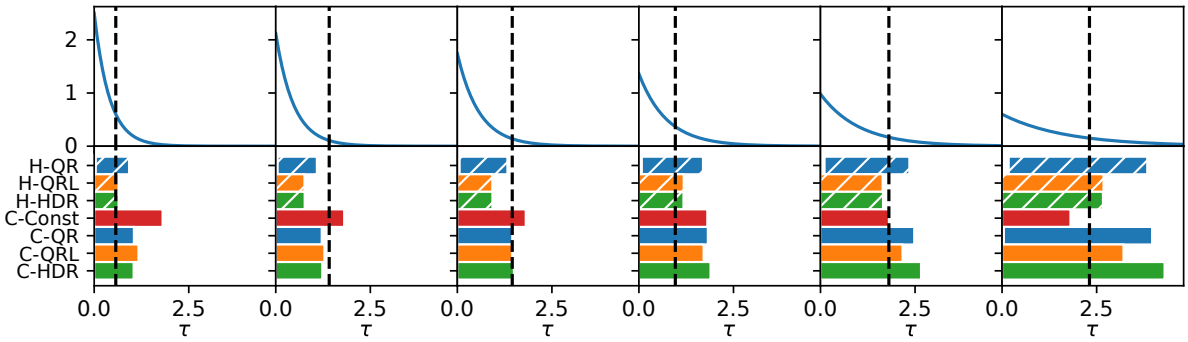


Figure F.13: Toy example with $\alpha = 0.2$ and a calibration dataset of 6 data points.

F.3.5 Computation time

In Table F.1, we present the computation times for evaluating the scores and regions across each conformal method utilized in our experiments. Except for `C-Const`, which incurs a minimal computation time primarily due to data loading, the computational demands of the other methods are of a comparable magnitude.

For all methods excluding `C-Const`, computation time is primarily governed by the calculation of the CDF of the time and the joint PDF of the time and mark. Specifically, the most resource-intensive tasks involve computing the quantiles of the time or generating samples from the time distribution, as these operations require inverting the CDF using the bisection method, typically necessitating around 30 evaluations.

In the cases of `C-QR` and `C-QRL`, the computation time is dominated by computing the quantiles of the time. For `C-HDR` (time and joint), `C-PROB`, `C-APS`, and `C-RAPS`, the primary computational load comes from generating time samples. More specifically, for `C-HDR`, these samples are needed to compute HPD values. For `C-PROB`, `C-APS`, and `C-RAPS`, computing the marginal PMF of the mark relative to the time involves averaging the joint density over the time across these samples.

F.3.6 Additional examples of joint prediction sets

Figures F.14 to F.17 present additional prediction sets generated by conformal methods on the datasets MOOC, Reddit, Retweets and Stack Overflow, respectively. We observe that C-HDR

Table F.1: Time to compute the scores and regions for all considered conformal methods on real-world datasets using the CLNM model, averaged over 5 runs, in seconds.

Dataset	Compute type	Time				Mark			Joint		
		C-Const	C-QR	C-QRL	C-HDR	C-PROB	C-APS	C-RAPS	C-QRL -RAPS	C-HDR -RAPS	C-HDR
LastFM	Score	0.07	15.01	8.10	8.63	8.40	8.40	8.36	16.56	17.11	8.71
	Region	0.10	9.97	5.42	10.51	5.61	5.62	5.61	10.34	15.55	11.70
MOOC	Score	0.38	93.17	47.64	51.70	49.68	49.33	49.71	96.90	102.07	51.88
	Region	0.75	62.32	32.06	66.46	33.30	33.36	33.58	64.24	99.40	74.98
Reddit	Score	0.25	56.47	29.04	31.51	30.34	30.23	30.19	59.48	61.97	31.93
	Region	0.48	38.31	19.84	40.60	20.72	20.78	20.73	39.48	60.66	45.68
Retweets	Score	0.60	159.38	80.30	85.84	84.54	84.27	84.13	165.46	171.16	86.88
	Region	0.56	106.49	53.93	112.55	56.77	56.43	56.60	110.03	169.60	114.00
Stack Overflow	Score	0.45	105.68	53.33	57.20	55.86	55.92	55.76	113.25	112.95	57.55
	Region	0.58	71.12	35.91	75.17	37.90	38.14	37.88	79.21	111.91	79.11

generally selects more marks than C-QRL-RAPS and C-HDR-RAPS. However, the joint region produced by C-HDR is usually smaller.

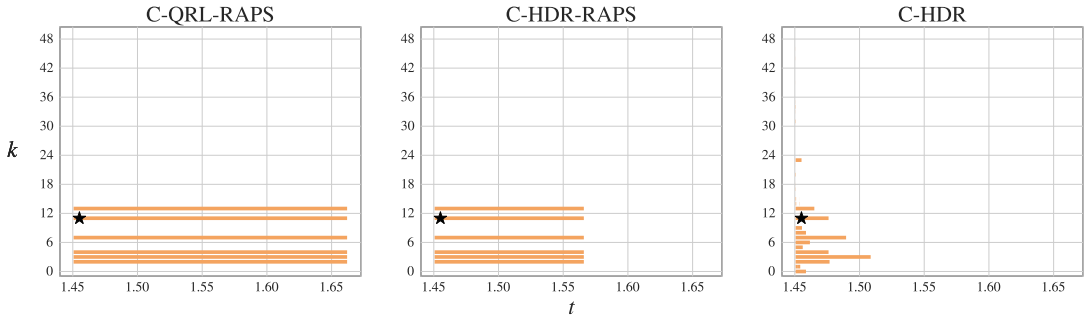


Figure F.14: Example of joint prediction sets generated for the last event of a test sequence in the MOOC dataset.

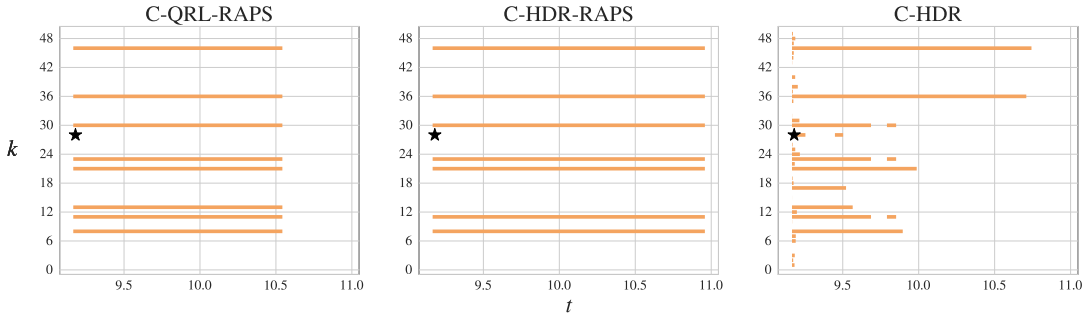


Figure F.15: Example of joint prediction sets generated for the last event of a test sequence in the Reddit dataset.

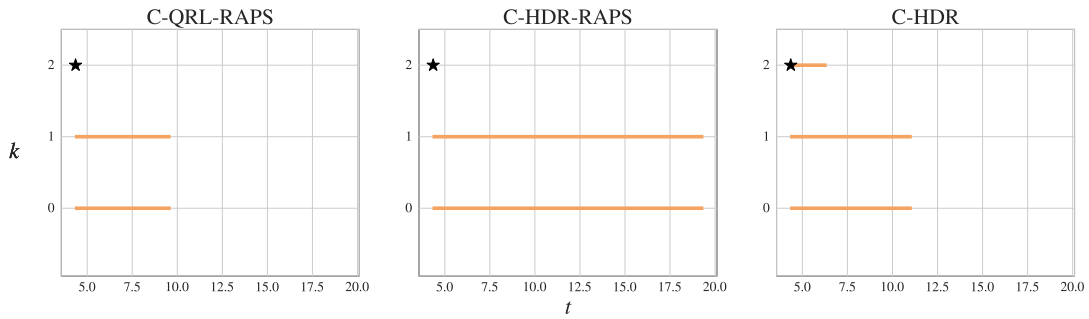


Figure F.16: Example of joint prediction sets generated for the last event of a test sequence in the Retweets dataset.

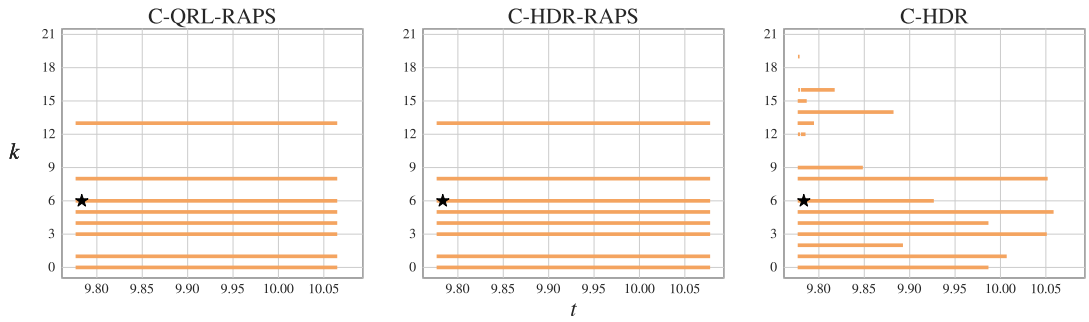


Figure F.17: Example of joint prediction sets generated for the last event of a test sequence in the Stack Overflow dataset.

APPENDIX G

Supplementary Material for Chapter 6

G.1. Details on Experimental Setup

This section aims to provide additional details on our experimental setup and implementation of the RCP algorithm.

Models. To facilitate fair comparison of different uncertainty estimation methods, we assume that the base prediction models are already trained. We focus on the regression problem and aim to construct prediction sets for these pre-trained models. All our models are based on a fully connected neural network of three hidden layers with 100 neurons in each layer and ReLU activations. We consider three types of base predictors with appropriate output layers and loss functions: the mean squared error for the *mean predictor*, the pinball loss for the *quantile predictor* or the negative log-likelihood loss for the *mixture predictor*. Training is performed with Adam optimizer.

Each dataset is split randomly into train, calibration, and test parts. We reserve 2048 points for calibration and the remaining data is split between 70% for training and 30% for testing. Each dataset is shuffled and split 10 times to replicate the experiment. This way we have 10 different models for each dataset and these models' predictions are used by every method that is tailored to the corresponding model type to estimate uncertainty. One fifth of the train dataset is reserved for early stopping.

RCP$_{\text{MLP}}$. This variation reserves a part (50%) of the original calibration set to train a quantile regression model for the $(1-\alpha)$-level quantile of the scores V. We again use three hidden layers with 100 units per layer for that task. The remaining half of the calibration set forms the "proper calibration set" and is used to compute the conformal correction.

RCP$_{\text{local}}$. The local quantile regression variant is similar to the previous one, so we use the same splitting of the available calibration data. Since only one bandwidth needs to be tuned, we use a simple grid search on a log-scale grid in the interval $[10^{-3}, 1]$.

Datasets. Table A.2 presents characteristics of multi-output tabular datasets from Tsoumakas

et al. (2011), Feldman et al. (2023), and Z. Wang et al. (2023). We restrict our selection to those with at least a total of 2000 instances. For data preprocessing, we follow the procedure of (Grinsztajn et al., 2022).

G.2. Additional Results

G.2.1 Additional results on marginal and conditional coverage

Figure G.1 extends the results of Figure 6.3 by displaying additionally the marginal coverage and worst-slab coverage. As expected, all methods obtain a correct marginal coverage. Furthermore, the methods with the best worst slab coverage (closest to $1 - \alpha$) also obtain a small conditional coverage error, supporting our conclusions in Section 6.7.

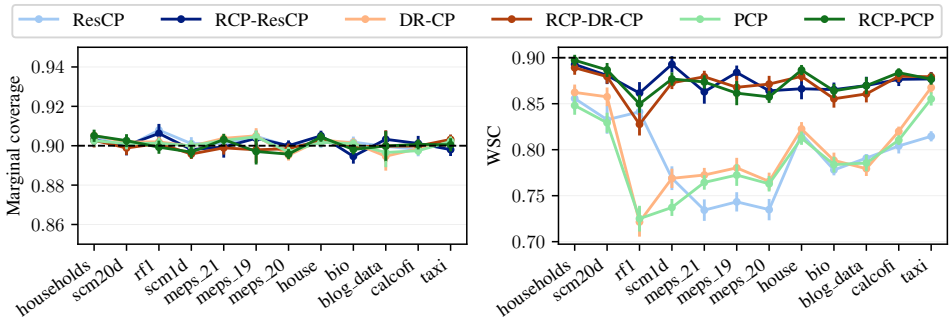


Figure G.1: Marginal coverage and worst-slab coverage for three conformal methods and their RCP counterparts, on datasets sorted by total size.

G.2.2 Estimation of conditional quantile function

Figure G.2 compares two ways of estimating $\widehat{\tau}$ (see Section 6.4). RCP_{MLP} corresponds to quantile regression based on a neural network as in Section 6.7, while $\text{RCP}_{\text{local}}$ corresponds to local quantile regression. On many datasets, the more flexible RCP_{MLP} is able to obtain better conditional coverage. However, local quantile regression has theoretical guarantees on its conditional coverage (see Section 6.6).

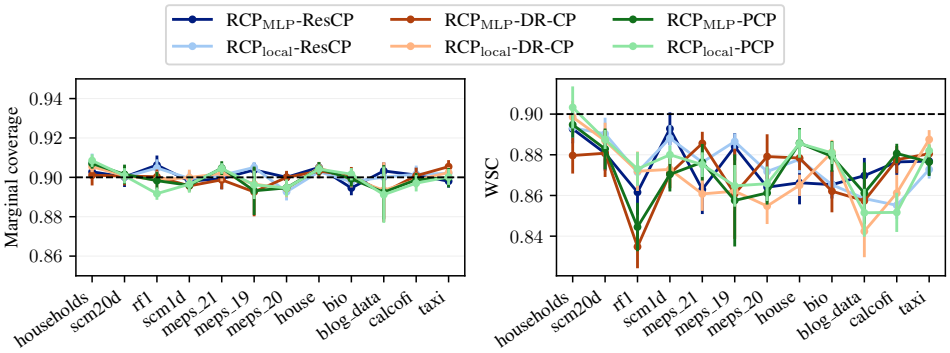


Figure G.2: Marginal coverage and worst-slab coverage for two types of quantile estimators in combination with different conformal methods, on datasets sorted by total size.

G.2.3 Choice of adjustment function

Figure G.3 compares RCP with difference $(-)$ and linear $(*)$ adjustments when combined with the DR-CP method. Since RCP with any adjustment function adheres to the SCP framework, marginal coverage is guaranteed, as shown in Panel 1.

The conformity score for DR-CP is defined as $V(x, y) = -\log \hat{p}(y \mid x)$, which can take negative values, implying that $\mathbb{T} = \mathbb{R}$. However, the linear adjustment requires $\mathbb{T} \subseteq \mathbb{R}_+^*$, violating **H2** and resulting in a failure to approximate conditional coverage accurately. This issue is evident in Panel 2. In contrast, the difference adjustment does not impose such a restriction.

Panel 3 compares PCP and ResCP when used with difference and linear adjustments. Since the conformity scores for these methods are always positive, i.e., $\mathbb{T} = \mathbb{R}_+^*$, both adjustment methods satisfy **H2**. In general, we observe no significant differences between the two adjustment methods.

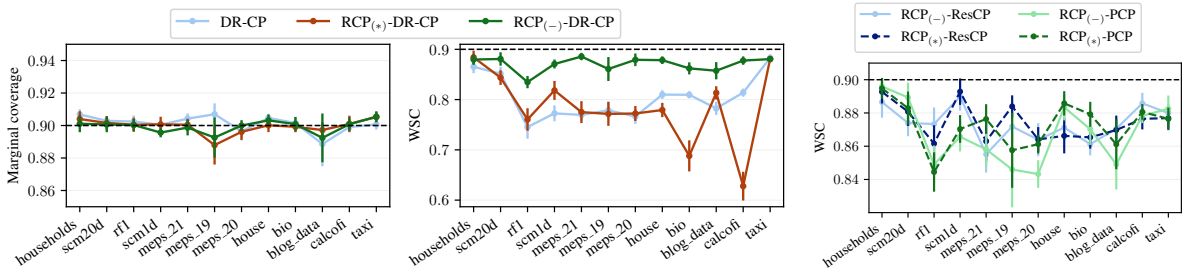


Figure G.3: Marginal coverage and worst-slab coverage obtained for two types of adjustments.

G.2.4 Additional adjustment functions

We consider two additional adjustment functions, namely $f_t(v) = \exp(t + v)$, denoted $\exp -$, and $f_t(v) = \exp(tv)$, denoted $\exp *$. To apply these custom adjustment functions we need to ensure that the conditions **H2** and **H3** are satisfied. For the first function we have: $\tilde{f}_\varphi^{-1}(v) = (\ln v) - \varphi \in \mathbb{T}$ and $\varphi = 0$. Then $\tilde{f}_\varphi^{-1}(v) > 0 \Rightarrow \ln v > 0 \Rightarrow v > 1$. For the second function we can take $\varphi = 1$ and by similar argument we arrive at the same requirement $v > 1$. In practice, conformity scores

are usually non-negative as is the case with PCP and residual scores that we consider here, and we can always add a constant 1 to satisfy this requirement.

Figures G.4 and G.5 show the marginal coverage and worst-slab coverage obtained with these adjustment functions.

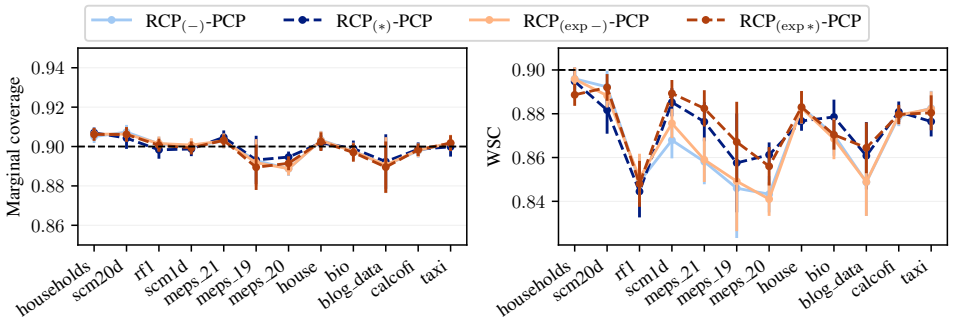


Figure G.4: Marginal coverage and worst-slab coverage for two additional types of adjustments combined with the method PCP.

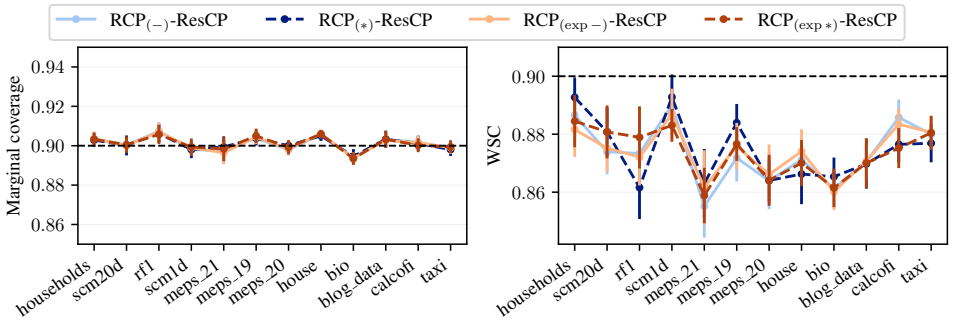


Figure G.5: Marginal coverage and worst-slab coverage for two additional types of adjustments combined with the method ResCP.

G.2.5 Direct comparison with CQR

Here we present a direct comparison of RCP with conformalized quantile regression (CQR; Romano et al. (2019)). We use the same underlying neural network architectures for the models as in our main experiment. Similarly to ResCP, to handle multi-dimensional outputs, we follow (Diquigiovanni et al., 2021b) and define the conformity score of CQR as the l^∞ norm of the CQR conformity scores across dimensions. Specifically, we compare CQR to DR-CP and its RCP-DR-CP counterpart, which achieves the best median volume overall.

Figure G.6 shows that CQR matches the conditional coverage of RCP-DR-CP. However, it produces larger median prediction sets due to less flexible shapes.

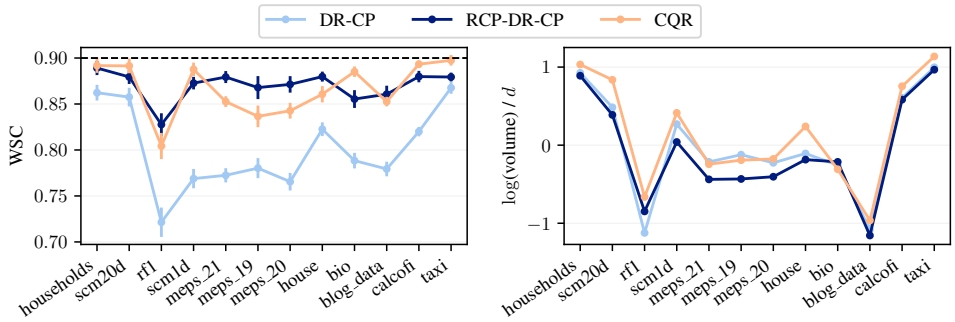


Figure G.6: Worst slab coverage and (logarithm) median prediction set volume (scaled by d).

G.2.6 Comparison of prediction set volumes

Table G.1 shows the average volume obtained by the methods compared in Section 6.7. Non-RCP methods obtain a smaller average volume across all datasets. The larger average volume of RCP is explained by the larger sets produced for instances with larger uncertainty.

Table G.1: Mean prediction set volume per dataset.

dataset	PCP	RCP-PCP	DR-CP	RCP-DR-CP	ResCP	RCP-ResCP
households	88.3	1.33e+02	**47.4**	1.02e+02	1.81e+02	4.51e+02
scm20d	**4.26e+05**	7.92e+06	1.11e+06	3.95e+07	5.22e+05	7.15e+12
rf1	0.0274	0.190	**0.000562**	4.35e+04	0.0276	7.94e+08
scm1d	**1.92e+04**	1.30e+08	2.30e+04	1.67e+08	6.27e+04	2.04e+15
meps_21	1.65	10.0	**0.746**	6.07	5.35	8.32e+12
meps_19	90.0	3.27e+04	**3.64**	3.27e+04	5.56	3.56e+22
meps_20	1.68	5.50	**0.761**	6.20	5.38	6.27e+13
house	0.676	0.936	**0.519**	0.751	2.92	3.88
bio	0.579	1.12	**0.414**	0.645	1.05	1.16
blog_data	0.459	8.45e+02	**0.143**	6.37e+02	1.26	6.79e+21
calcofi	3.47	4.12	**2.45**	3.06	4.53	4.47
taxi	9.21	9.63	**5.69**	6.40	12.4	12.8

In contrast, Table G.2 shows that RCP obtains smaller sets across most datasets when comparing the median volume, avoiding outliers.

Table G.2: Median prediction set volume per dataset.

dataset	PCP	RCP-PCP	DR-CP	RCP-DR-CP	ResCP	RCP-ResCP
households	67.4	56.5	39.2	**32.1**	1.81e+02	1.67e+02
scm20d	3.11e+04	**1.42e+04**	7.03e+05	7.74e+04	5.22e+05	2.00e+05
rf1	0.0110	0.00525	**0.000583**	33.1	0.0276	0.0231
scm1d	1.16e+02	**2.05**	1.01e+04	25.7	6.27e+04	1.70e+03
meps_21	1.04	0.704	0.433	**0.227**	5.35	2.42
meps_19	3.85	0.754	2.10	**0.303**	5.56	2.54
meps_20	1.05	0.616	0.416	**0.254**	5.38	2.51
house	0.596	0.519	0.471	**0.386**	2.92	2.67
bio	0.507	0.435	0.374	**0.344**	1.05	0.829
blog_data	0.229	0.209	0.0869	**0.0597**	1.26	1.16
calcofi	3.83	3.98	2.85	**2.77**	4.53	4.75
taxi	8.67	8.25	**5.22**	5.27	12.4	10.1

G.2.7 Improved data efficiency using cross-validation

As explained in Section 6.7, RCP requires to divide the calibration dataset $\mathcal{D}$ into two parts, one to estimate $\widehat{\tau}$, and one for SCP.

In this section, we consider a more data-efficient approach using the training dataset $\mathcal{D}_{\text{train}}$. Using K-fold cross-validation on $\mathcal{D}_{\text{train}}$, for each fold index k, we train a model on the $K-1$ remaining folds and evaluate the conformity score on the fold k. This yields a dataset $\mathcal{D}_\tau$ of size $|\mathcal{D}_{\text{train}}|$ with inputs and their associated conformity scores based on which $\widehat{\tau}$ is estimated. This also removes the need to split the calibration dataset. An additional model is fitted on the complete training dataset $\mathcal{D}_{\text{train}}$ to produce the non-rectified conformity scores.

Figure G.7 shows a comparison of learning $\widehat{\tau}$ on half the calibration dataset (cal), or using 10-fold cross-validation (CV). The cross-validation approach yields improved worst-slab coverage on most datasets. This improved conditional coverage comes at the computational cost of training K additional models.

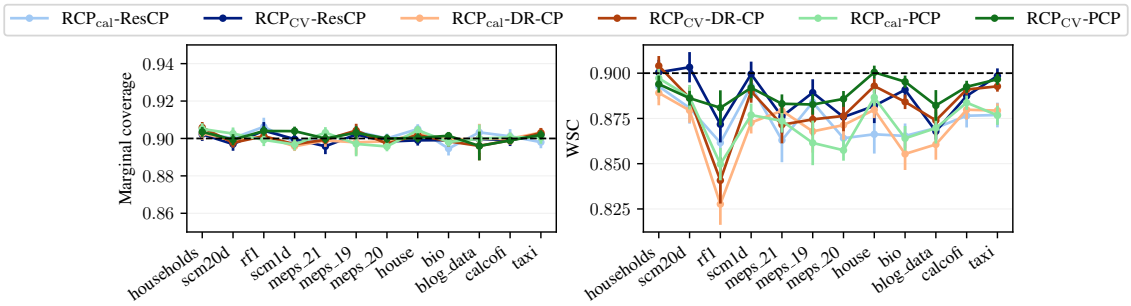


Figure G.7: Worst-slab coverage of RCP with $\widehat{\tau}$ trained on half the calibration dataset (cal) or using 10-fold cross-validation (CV).

G.2.8 Comparison with CPCG

We conduct an additional experiment comparing RCP with Conditional Prediction with Conditional Guarantees (CPCG; Gibbs et al. (2025)). We evaluate RCP using both the full calibration dataset (RCP_{cal}) and cross-validation (RCP_{CV}), as described in Section G.2.7. All methods are run on CPU (AMD Ryzen Threadripper PRO 5965WX) with 6 CPU threads per experiment.

Table G.3 shows that all methods achieve comparable worst-slab coverage, close to the nominal level. However, Table G.4 reveals a stark contrast in computational efficiency: CPCG is 200-100,000 times slower than RCP_{cal} and 10-100 times slower than RCP_{CV} overall. This significant overhead is because CPCG must solve an optimization problem involving the entire calibration set *for each test instance*. Consequently, CPCG's computational demands become prohibitive for large calibration and test sets, hindering its practical application. Moreover, CPCG failed to find a solution on the "house" and "calcofi" datasets, precluding results for these cases. These factors highlight RCP's substantial practical advantage in efficiency, especially for large-scale datasets.

Table G.3: Comparison of worst-slab coverage on multi-output datasets.

	PCP	RCP_{cal}-PCP	RCP_{CV}-PCP	CPCG-PCP	DR-CP	RCP_{cal}-DR-CP	RCP_{CV}-DR-CP	CPCG-DR-CP
households	0.825	0.905	0.899	0.888	0.853	0.891	0.900	0.900
scm20d	0.830	0.892	0.891	0.897	0.877	0.877	0.868	0.899
rf1	0.731	0.830	0.877	0.838	0.715	0.863	0.827	0.872
scm1d	0.758	0.882	0.895	0.910	0.756	0.902	0.896	0.882
meps_21	0.739	0.874	0.904	0.881	0.789	0.881	0.879	0.905
meps_19	0.762	0.875	0.867	0.880	0.788	0.884	0.889	0.878
meps_20	0.731	0.842	0.871	0.890	0.719	0.880	0.884	0.892
house	0.835	0.895	0.903	/	0.817	0.878	0.906	/
bio	0.784	0.860	0.900	0.887	0.774	0.879	0.880	0.880
blog_data	0.770	0.877	0.893	0.886	0.749	0.844	0.888	0.888
calcofi	0.810	0.889	0.888	/	0.828	0.885	0.892	/
taxi	0.837	0.885	0.884	0.881	0.846	0.872	0.879	0.879

Table G.4: Comparison of computational time (in seconds) on multi-output datasets.

	PCP	RCP_{cal}-PCP	RCP_{CV}-PCP	CPCG-PCP	DR-CP	RCP_{cal}-DR-CP	RCP_{CV}-DR-CP	CPCG-DR-CP
households	0.258	0.604	104	8840	0.00759	0.531	104	8164
scm20d	2.63	4.22	772	6409	0.0182	0.852	766	6012
rf1	0.667	1.35	340	10682	0.00836	0.348	339	9674
scm1d	2.27	3.57	1209	4971	0.0133	0.867	1205	4692
meps_21	0.236	0.607	581	6283	0.0123	0.261	581	6031
meps_19	0.272	0.515	493	6411	0.0123	0.184	492	6128
meps_20	0.255	0.520	621	7147	0.0119	0.238	621	7032
house	0.315	0.594	1034	/	0.0159	0.327	1033	/
bio	0.630	1.22	3161	79422	0.0279	0.782	3163	63178
blog_data	0.752	0.850	1119	41192	0.0336	0.155	1121	43289
calcofi	0.699	1.02	456	/	0.0356	0.200	455	/
taxi	0.680	1.17	866	77828	0.0269	0.276	866	70139

G.3. Proofs

G.3.1 Proof for the first example

We provide here a completely elementary proof. The result actually follows from Theorem 10. In this example, we set $\tau_\star(x) = Q_{\mathbf{s}|X=x}(1-\alpha)$, where $\mathbf{s} = s(X,Y)$. We assume that for all $x \in \mathcal{X}$, $\tau_\star(x) > 0$. We denote $\tilde{s}(x,y) = s(x,y)/\tau_\star(x)$ and $\tilde{\mathbf{s}} = \tilde{s}(X,Y)$.

$$Q_{\tilde{\mathbf{s}}}(1-\alpha) = \inf\{t \in \mathbb{R} : \mathbb{P}(s(X,Y) \leqslant t\tau_\star(X)) \geqslant 1-\alpha\}.$$

We will first prove that, for all $x \in \mathcal{X}$, we get that $1 = Q_{\tilde{\mathbf{s}}|X=x}(1-\alpha)$:

$$\begin{aligned} Q_{\tilde{\mathbf{s}}|X=x}(1-\alpha) &= \inf\{t \in \mathbb{R} : \mathbb{P}(s(X,Y) \leqslant t\tau_\star(x)|X=x) \geqslant 1-\alpha\} \\ &= \inf\{t \in \mathbb{R} : \mathbb{P}(\mathbf{s} \leqslant tQ_{\mathbf{s}|X=x}(1-\alpha)|X=x) \geqslant 1-\alpha\} = 1. \end{aligned}$$

We then show that $Q_{\tilde{\mathbf{s}}}(1-\alpha) \leqslant 1$. Indeed, for any $t > 1$, by the tower property of conditional expectation, we get:

$$\begin{aligned} \mathbb{P}(s(X,Y) \leqslant t\tau_\star(X)) &= \mathbb{P}(\mathbf{s} \leqslant tQ_{\mathbf{s}|X}(1-\alpha)) \\ &= \mathbb{E}\big[\mathbb{P}(\mathbf{s} \leqslant tQ_{\mathbf{s}|X}(1-\alpha)|X)\big] \geqslant 1-\alpha. \end{aligned}$$

Assume now that $Q_{\tilde{\mathbf{s}}}(1-\alpha) < 1$. Then for any $t \in (Q_{\tilde{\mathbf{s}}}(1-\alpha), 1)$, using again the tower property of conditional expectation, we get

$$1-\alpha \leqslant \mathbb{P}(s(X,Y) \leqslant t\tau_\star(X)) = \mathbb{E}\big[\mathbb{P}(\mathbf{s} \leqslant tQ_{\mathbf{s}|X}(1-\alpha)|X)\big] \tag{G.1}$$

$$= \mathbb{E}\big[\mathbb{P}(\mathbf{s} \leqslant tQ_{\mathbf{s}|X}(1-\alpha)|X)\big] < 1-\alpha \tag{G.2}$$

by the definition of the conditional quantile. This yields a contradiction. Therefore, for P_X-a.e. $x \in \mathcal{X}$,

$$Q_{\tilde{\mathbf{s}}}(1-\alpha) = Q_{\tilde{\mathbf{s}}|X=x}(1-\alpha).$$

G.3.2 Proof for the second example

We set in this case $\tilde{s}(x,y) = s(x,y) - \tau_\star(x)$, where $\tau_\star(x) = Q_{\mathbf{s}|X=x}(1-\alpha)$ and $\tilde{\mathbf{s}} = \tilde{s}(X,Y)$. We will show that $Q_{\tilde{\mathbf{s}}|X=x}(1-\alpha) = 0$ for all $x \in \mathcal{X}$. We have indeed:

$$Q_{\tilde{\mathbf{s}}|X=x}(1-\alpha) = \inf\{t \in \mathbb{R} : \mathbb{P}(\tilde{s}(X,Y) \leqslant t|X=x) \geqslant 1-\alpha\} \tag{G.3}$$

$$= \inf\{t \in \mathbb{R} : \mathbb{P}(s(X,Y) \leqslant \tau_\star(x) + t|X=x) \geqslant 1-\alpha\} = 0. \tag{G.4}$$

We will now show that $Q_{\tilde{\mathbf{s}}}(1-\alpha) \leqslant 0$. Indeed, for all $t > 0$, by the tower property of conditional expectation and the definition of the conditional quantile, we get

$$\mathbb{P}(\tilde{s}(X,Y) \leqslant t) = \mathbb{E}[\mathbb{P}(s(X,Y) \leqslant \tau_\star(X) + t|X)] \geqslant 1-\alpha. \tag{G.5}$$

On the other hand, assume $Q_{\tilde{\mathbf{s}}}(1-\alpha) < 0$. Set $t \in (Q_{\tilde{\mathbf{s}}}(1-\alpha), 0)$. We get

$$1-\alpha \leqslant \mathbb{P}(\tilde{s}(X,Y) \leqslant t) = \mathbb{P}(s(X,Y) \leqslant t + \tau_\star(X)) \tag{G.6}$$

$$= \mathbb{E}[\mathbb{P}(s(X,Y) \leqslant t + \tau_\star(X)|X)] < 1-\alpha, \tag{G.7}$$

which leads to a contradiction.

G.3.3 Proof of equality (6.8)

Theorem 10. Assume **H2**-**H3** hold. For $x \in \mathcal{X}$, set $\tau_\star(x) = Q_{\mathbf{s}_\varphi|X=x}(1-\alpha)$, where $s_\varphi(x,y) = \tilde{f}_\varphi^{-1}(s(x,y))$. Set $\tilde{s}_\star(x,y) = f^{-1}_{\tau_\star(x)}(s(x,y))$ and $\tilde{\mathbf{s}}_\star = \tilde{s}_\star(X,Y)$. Then, for all $x \in \mathcal{X}$,

$$\varphi = Q_{\tilde{\mathbf{s}}_\star|X=x}(1-\alpha) = Q_{\tilde{\mathbf{s}}_\star}(1-\alpha).$$

Proof. Set $\psi(x) = Q_{\tilde{\mathbf{s}}_\star|X=x}(1-\alpha)$. We must prove that $\psi(x) = \varphi$ for all $x \in \mathcal{X}$. First, we will show $\psi(x) \leqslant \varphi$. Note indeed

$$\mathbb{P}(\tilde{s}_\star(X,Y) \leqslant \varphi | X = x) = \mathbb{P}(s(X,Y) \leqslant f_{\tau_\star(X)}(\varphi) | X = x) \overset{(a)}{=} \mathbb{P}(s(X,Y) \leqslant \tilde{f}_\varphi(\tau_\star(X)) | X = x) \tag{G.8}$$

$$\overset{(b)}{=} \mathbb{P}(\tilde{f}_\varphi^{-1}(s(X,Y)) \leqslant \tau_\star(X) | X = x) \overset{(c)}{\geqslant} 1-\alpha, \tag{G.9}$$

where (a) follows from $f_t(\varphi) = \tilde{f}_\varphi(t)$, (b) from the fact that $\tilde{f}_\varphi$ is invertible, and (c) from the definition of $\tau_\star(x)$.

Now, suppose that $\psi(x) < \varphi$. Since for any t, f_t is increasing, we get that $f_{\tau_\star(x)}(\psi(x)) < f_{\tau_\star(x)}(\varphi)$. Moreover, using that $\tau_\star(x)$ belongs to the interior of $\mathbb{T}$, combined with the continuity of $t \in \mathbb{T} \mapsto \tilde{f}_\varphi(t)$; it implies the existence of $\tilde{t} \in \mathbb{T}$ such that $\tilde{t} < \tau_\star(x)$ and also $f_{\tau_\star(x)}(\psi(x)) < f_{\tilde{t}}(\varphi)$. We can rewrite

$$\begin{aligned} 1-\alpha \leqslant \mathbb{P}(s(X,Y) \leqslant f_{\tau_\star(X)}(\psi(X)) \mid X = x) &\leqslant \mathbb{P}(s(X,Y) \leqslant f_{\tilde{t}}(\varphi) \mid X = x) \\ &= \mathbb{P}(\tilde{f}_\varphi^{-1}(s(X,Y)) \leqslant \tilde{t} \mid X = x) < 1-\alpha. \end{aligned}$$

which yields to a contradiction.

We now show that $Q_{\tilde{\mathbf{s}}_\star}(1-\alpha) = \varphi$. We first show that $Q_{\tilde{\mathbf{s}}_\star}(1-\alpha) \leqslant \varphi$. This follows from

$$\mathbb{P}(\tilde{s}_\star(X,Y) \leqslant \varphi) \overset{(a)}{=} \mathbb{E}[\mathbb{P}(\tilde{s}_\star(X,Y) \leqslant \varphi | X)] \overset{(b)}{\geqslant} 1-\alpha,$$

where (a) follows from the tower property of conditional expectation and (b) from $\varphi = Q_{\tilde{\mathbf{s}}_\star|X=x}(1-\alpha)$ for all $x \in \mathcal{X}$.

Assume now that $Q_{\tilde{\mathbf{s}}_\star}(1-\alpha) < \varphi$. Choose $q \in (Q_{\tilde{\mathbf{s}}_\star}(1-\alpha), \varphi)$. Then,

$$1-\alpha \leqslant \mathbb{P}(\tilde{s}_\star(X,Y) \leqslant q) \overset{(a)}{=} \mathbb{E}[\mathbb{P}(\tilde{s}_\star(X,Y) \leqslant q | X)] \overset{(b)}{<} 1-\alpha,$$

where (a) follows from the tower property of conditional expectation and (b) $q < \varphi = Q_{\tilde{\mathbf{s}}_\star|X=x}(1-\alpha)$ for all $x \in \mathcal{X}$. This yields to a contradiction which concludes the proof. □

APPENDIX H

Supplementary Material for Chapter 7

H.1. Proofs

H.1.1 Jacobian determinant of the radial transform

Recall that the transformation R is defined as

$$R(z) = \frac{r(l)}{l} z \tag{H.1}$$

where $l = \|z\|$, for $z \neq 0$. It maps z to a new vector $R(z)$ such that its norm becomes $r(l)$ while its direction z/l is preserved (for $z \neq 0$). We analyze this transformation using hyperspherical coordinates $(l, \omega_1, \ldots, \omega_{d-1})$, where $l = \|z\|$ is the radial distance and $(\omega_1, \ldots, \omega_{d-1})$ are the angular coordinates. The transformation R maps these coordinates from $(l, \omega_1, \ldots, \omega_{d-1})$ to $(r(l), \omega_1, \ldots, \omega_{d-1})$, as only the radial distance is altered.

The Cartesian volume element $\mathrm{d}^d z$ is related to the hyperspherical volume element by $\mathrm{d}^d z = l^{d-1} \mathrm{d}l \, \mathrm{d}\Omega_{d-1}$, where $\mathrm{d}\Omega_{d-1}$ is the surface element on the unit $(d-1)$-sphere. Under the transformation R, the new radial coordinate is $l' = r(l)$, so its differential is $\mathrm{d}l' = \frac{\partial r(l)}{\partial l} \mathrm{d}l$. The angular part $\mathrm{d}\Omega_{d-1}$ remains unchanged. The transformed volume element $\mathrm{d}^d R(z)$ is thus given by:

$$\mathrm{d}^d R(z) = (r(l))^{d-1} \left(\frac{\partial r(l)}{\partial l} \mathrm{d}l \right) \mathrm{d}\Omega_{d-1}. \tag{H.2}$$

The Jacobian determinant $|\det (\nabla_z R(z))|$ is the ratio of the transformed volume element $\mathrm{d}^d R(z)$ to the original volume element $\mathrm{d}^d z$:

$$|\det (\nabla_z R(z))| = \frac{(r(l))^{d-1} \frac{\partial r(l)}{\partial l} \mathrm{d}l \, \mathrm{d}\Omega_{d-1}}{l^{d-1} \mathrm{d}l \, \mathrm{d}\Omega_{d-1}} = \left(\frac{r(l)}{l} \right)^{d-1} \frac{\partial r(l)}{\partial l}, \tag{H.3}$$

which corresponds to (7.9). This holds for $l = \|z\| > 0$. Since $r : \mathbb{R}_+ \to \mathbb{R}_+$, $r(l) \geqslant 0$. Furthermore, $r(l)$, as defined by (7.4), is a composition of non-decreasing functions (a CDF and an inverse CDF), making it non-decreasing, so $\frac{\partial r(l)}{\partial l} \geqslant 0$. Thus, the expression is non-negative.

We note that radial flows (D. Rezende and Mohamed, 2015) are a special case of (H.1) with $r(l) = l \cdot (\alpha + l + \beta)/(\alpha + l)$ where $\alpha \in \mathbb{R}_+$ and $\beta \in \mathbb{R}$ are learned parameters.

H.2. Differentiable calibration maps using density estimation

To obtain a differentiable calibration map $\hat{F}_{\hat{L}}$, we estimate the PDF $\hat{f}_{\hat{L}}$ of the calibration data using density estimation. We identified two approaches that performed well in our experiments.

As an implementation detail for both approaches, density estimation was generally improved by first applying the transformation $g(t) = t^{1/3}$ to the calibration scores $\{\hat{L}_i\}_{i=1}^n$. After density estimation in the transformed space, the data is rescaled using the inverse transformation $g^{-1}(t) = t^3$.

H.2.1 Kernel density estimation with Gamma kernels

We found that kernel density estimation (KDE) with Gamma kernels is effective because the Gamma distribution has positive support, which is appropriate for the calibration scores $\hat{L}_i \geqslant 0$. Let $\Gamma(\alpha_\Gamma, \lambda_\Gamma)$ denote a Gamma distribution with shape $\alpha_\Gamma > 0$ and rate $\lambda_\Gamma > 0$. A Gamma distribution $\Gamma(\mu\lambda_\Gamma, \lambda_\Gamma)$ has a mean of μ. We center a Gamma kernel at each calibration score $\hat{L}_i$, using the distribution $\Gamma(\hat{L}_i\lambda_\Gamma, \lambda_\Gamma)$, which has a mean of $\hat{L}_i$.

The resulting estimated CDF $\hat{F}_{\hat{L}}(t)$ is given by the average of the individual kernel CDFs:

$$\hat{F}_{\hat{L}}(t) = \frac{1}{n} \sum_{i=1}^{n} F_{\Gamma(\hat{L}_i\lambda_\Gamma, \lambda_\Gamma)}(t). \tag{H.4}$$

The rate parameter λ_Γ is chosen by minimizing the NLL of the calibration dataset under the KDE model. This is done using 10-fold cross-validation over the grid $\left\{10^{-5+10\cdot\frac{i}{99}}\right\}_{i=0}^{99}$. This hyperparameter selection process is efficient and performed once per run.

Figure H.1 shows an example fit on all datasets, illustrating the empirical and estimated smooth CDFs (left y axis) and the estimated log PDF (right y axis).

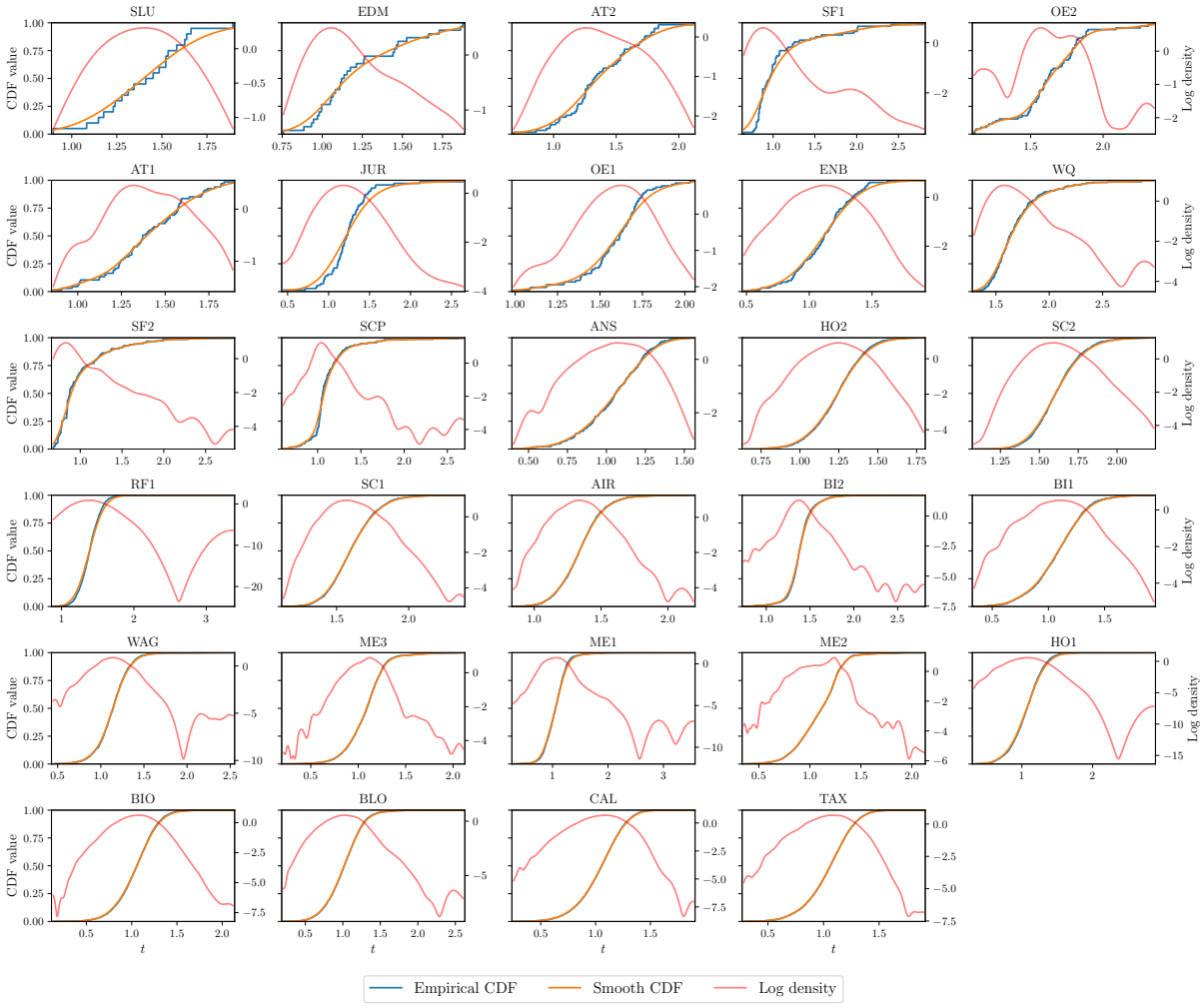


Figure H.1: Density estimation using KDE with a Gamma kernel.

H.2.2 Rational Quadratic Splines

Rational Quadratic Splines (Durkan et al., 2019) provide a flexible framework for defining invertible and differentiable transformations. A base spline Φ maps $[-1, 1]$ to $[-1, 1]$. To handle the unbounded domain of the latent norms, we use the transformation $\Psi = \tanh^{-1} \circ \Phi \circ \tanh$, which maps $\mathbb{R}$ to $\mathbb{R}$ and retains invertibility and differentiability. This transformation is used to model the distribution of the latent norms by learning a mapping from a standard Gaussian distribution to the data distribution.

For training, the data is normalized to have zero mean and unit variance. The parameters of the spline are optimized to minimize the NLL of the calibration dataset under the defined model. Optimization is performed using Adam (Kingma and Ba, 2015). To maximize data information, we perform early stopping on the training dataset itself and stop if the loss did not improve by 1e-4 for 50 epochs. Overfitting is prevented by limiting the number of bins and thus the flexibility of the spline. Specifically, we use 4 bins if $n \leqslant 30$, 5 bins if $n \leqslant 50$, 6 bins if $n \leqslant 70$, 7 bins if

$n \leqslant 80$, 8 bins if $n \leqslant 90$, and 9 bins if $n \leqslant 100$.

Similar to Figure H.1, Figure H.2 shows an example fit for all datasets.

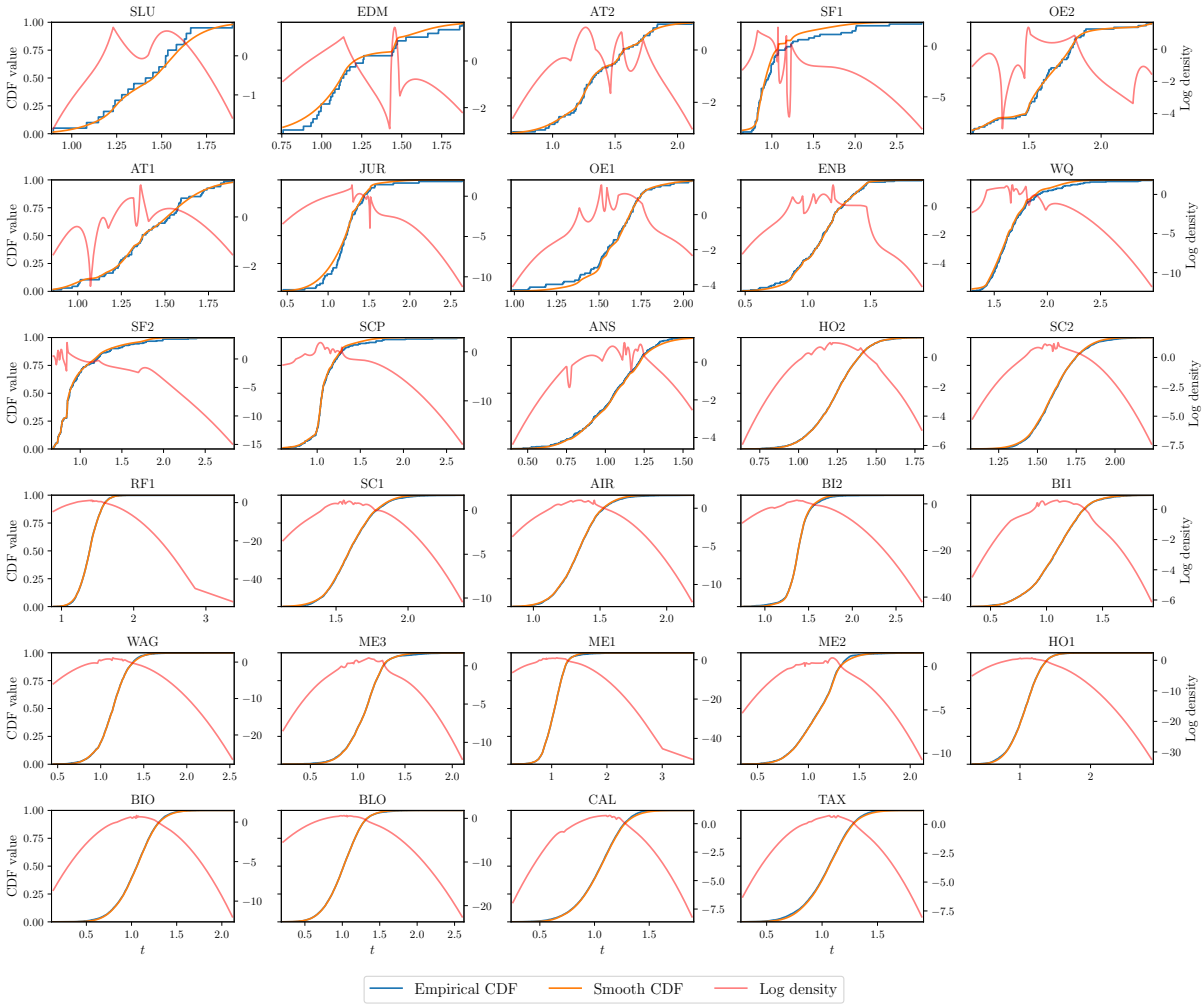


Figure H.2: Density estimation using a rational quadratic spline.

H.2.3 Challenges with numerical precision

In this section, we address a potential alternative approach and explain why it is impractical due to numerical precision constraints. Theoretically, one could attempt to estimate the calibration map $\hat{F}_{\hat{U}}$ using density estimation on the quantity $\hat{U} = F_{\rho_{\hat{\mathcal{Z}}}(Z)}(\ell_{\hat{T}}(Y; X))$. Since $\hat{U}$ is expected to follow a standard uniform distribution under ideal conditions, estimating its density might appear practical.

However, this approach faces significant numerical precision issues, particularly when the latent space dimensionality d is large. When d is large, the CDF $F_{\rho_{\hat{\mathcal{Z}}}(Z)}$ for $\rho_{\hat{\mathcal{Z}}}(Z) \sim \chi_d$ becomes extremely steep around its mode $\sqrt{d-1}$. For example, in our image application with $d = 196,608$, a proportion 99% of the probability mass is concentrated in the narrow interval $[440.7, 446.2]$. In

single-precision floating-point arithmetic, the CDF saturates quickly: $F_{\chi_d}(t)$ is numerically 0.0 for $t < 433.4$ and 1.0 for $t > 447.2$.

If the latent model is miscalibrated, the values of $\ell_{\hat{T}}(Y; X)$ for the calibration data can fall outside this narrow range where the CDF has fine-grained variation. This results in many $\hat{U}$ values being numerically 0.0 or 1.0. For smaller dimensions, similar issues can occur, although less frequently. For instance, with $d = 1$, $F_{\chi_1}(t)$ is numerically 1.0 for $t > 5.54$ in single precision. When a significant portion of the calibration data for $\hat{U}$ consists of values numerically identical to 0.0 or 1.0, accurate density estimation becomes impossible.

For this reason, in Section 7.4.2, we based our calibration map on density estimation of $\hat{L} = \ell_{\hat{T}}(Y; X)$ directly, using $\hat{F}_{\hat{L}}$.

H.3. Additional details on experimental setup

Computing the main tabular data results requires approximately 24 hours on an RTX A6000 GPU, and reproducing the image results requires approximately 6 hours on an RTX 6000 GPU. Experiments can require up to 48 GB of VRAM, primarily due to large batch sizes during sampling for evaluation metrics. Decreasing the batch size reduces VRAM requirements but increases computation time.

For tabular datasets, we tune hyperparameters using grid search, selecting those that yield the lowest NLL on the validation set. The number of units in the input convex neural network is chosen from $[10, 20, 40]$, the number of layers from $[2, 3, 5]$, and the learning rate from $[5 \times 10^{-3}, 10^{-3}, 2 \times 10^{-4}]$. All models are trained by minimizing the NLL with the Adam optimizer (Kingma and Ba, 2015) using a batch size of 1024.

H.3.1 Evaluation metrics

NLL. We compute the average NLL over the test set as $\mathcal{D}_{\text{test}}$:

$$\widehat{\text{NLL}} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(X,Y) \in \mathcal{D}_{\text{test}}} - \log \hat{f}_{Y|X=X}(Y). \tag{H.5}$$

L-ECE. For each test point $(X^{(i)}, Y^{(i)}) \in \mathcal{D}_{\text{test}}$, we compute the PIT of the latent norm $\hat{U}_i = F_{\rho_Z(Z)}\left(\ell_{\hat{T}}(Y^{(i)}; X^{(i)})\right)$. The Latent Expected Calibration Error (L-ECE) is then estimated as the L_1 distance between the empirical CDF of $\{\hat{U}_i\}_{i=1}^{|\mathcal{D}_{\text{test}}|}$ and the uniform CDF:

$$\widehat{\text{L-ECE}} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{j=1}^{|\mathcal{D}_{\text{test}}|} \left| \hat{U}_{(j)} - \frac{j}{|\mathcal{D}_{\text{test}}| + 1} \right|, \tag{H.6}$$

where $\hat{U}_{(j)}$ denotes the j-th order statistic of the computed PIT values.

Energy Score. For each test point $(X, Y) \in \mathcal{D}_{\text{test}}$, we generate two independent sets of K samples, $\mathcal{S}_x$ and $\mathcal{S}'_x$, from the predictive distribution $\hat{f}_{Y|X=x}(\cdot)$. The Energy Score (ES) is

estimated as:

$$\widehat{\mathrm{ES}} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(X,Y)\in\mathcal{D}_{\text{test}}} \left(\frac{1}{K} \sum_{\hat{y}\in\mathcal{S}_x} \|\hat{y} - Y\| - \frac{1}{2K^2} \sum_{\hat{y}\in\mathcal{S}_x, \hat{y}'\in\mathcal{S}'_x} \|\hat{y} - \hat{y}'\| \right). \tag{H.7}$$

In our experiments, we use $K = 100$.

HDR-ECE. For each test point $(X^{(i)}, Y^{(i)}) \in \mathcal{D}_{\text{test}}$, we compute $G_i = \mathrm{HPD}_{\hat{f}_{Y|X=X^{(i)}}}(Y^{(i)})$, as defined in Table 7.1. The HDR Expected Calibration Error (HDR-ECE) is estimated similarly to L-ECE:

$$\widehat{\text{HDR-ECE}} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{j=1}^{|\mathcal{D}_{\text{test}}|} \left| G_{(j)} - \frac{j}{|\mathcal{D}_{\text{test}}| + 1} \right|, \tag{H.8}$$

where $G_{(j)}$ is the j-th order statistic of the computed HDR pre-ranks. Note that computing the HDR-ECE for `HDR-R` exactly is not possible as `HDR-R` does not yield an explicit recalibrated density $\hat{f}'_{Y|X}$. Following Y. Chung et al. (2024), we use the density $\hat{f}_{Y|X}$ of the original (non-recalibrated) model for `HDR-R` when evaluating its HDR-ECE.

BPD. For image datasets, we report Bits Per Dimension (BPD), calculated as in Zhai et al. (2025), where d is the output dimensionality (e.g., $d = 3 \times 256 \times 256$ for AFHQ). The BPD is then:

$$\widehat{\mathrm{BPD}} = (\widehat{\mathrm{NLL}}/d + \log 128)/\log 2. \tag{H.9}$$

Here, the $\log 128$ term accounts for the scaling of pixel values from $[0, 255]$ to $[-1, 1]$, and division by $\log 2$ converts the NLL from nats to bits.

Relative NLL or ES. To better visualize improvements in NLL or ES relative to the baseline model `BASE`, we report the difference in these scores, normalized by the absolute value of the score of `BASE`. For example, the relative NLL for `LR` is computed as $(\widehat{\mathrm{NLL}}_{\texttt{LR}} - \widehat{\mathrm{NLL}}_{\texttt{BASE}})/|\widehat{\mathrm{NLL}}_{\texttt{BASE}}|$. A negative value indicates improvement by `LR`.

H.4. Additional Results

H.4.1 Decision-making experiment

To make the benefits of a full PDF concrete, we have conducted an experiment on a decision-making task.

Experiment setup. We use the SLUMP dataset, where inputs are ingredients for producing concrete and outputs $Y = (S, F, C) \in \mathbb{R}^3$ are three concrete properties. A manufacturer must decide among 3 actions $\mathcal{A} = \{A, B, D\}$ whether a given batch of ingredients is suitable for one of two projects (A or B), each with specific requirement regions, or if it should be discarded (D). The decision has different financial utilities and risks:

Table H.1: Comparison of estimation strategies and methods. The best performing combination is highlighted in bold.

Method	Estimation Strategy	Average Utility
`BASE`	Sampling	62.53 ± 11.33
`HDR-R`	Sampling	32.38 ± 10.81
`BASE`	PDF (Numerical Integration)	76.23 ± 11.99
`LR`	**PDF (Numerical Integration)**	$\mathbf{113.31 \pm 12.91}$

- Requirements for Project A: $7 \leqslant S \leqslant 20, 55 \leqslant F \leqslant 65, 25 \leqslant C \leqslant 40$.
- Requirements for Project B: $20 \leqslant S \leqslant 29, 70 \leqslant F \leqslant 100, 15 \leqslant C \leqslant 30$.

The expected utility for an agent with policy $a : \mathcal{X} \to \mathcal{A}$ is given by

$$\mathbb{E}[u(Y, a(X))] \text{ with } u(y, a) = \begin{cases} 2000, & \text{if } a = A \text{ and } y \in \text{Region}_A \\ -30, & \text{if } a = A \text{ and } y \notin \text{Region}_A \\ 1500, & \text{if } a = B \text{ and } y \in \text{Region}_B \\ -15, & \text{if } a = B \text{ and } y \notin \text{Region}_B \\ -10, & \text{if } a = C. \end{cases} \tag{H.10}$$

The optimal action is chosen by maximizing the estimated expected utility. This requires estimating the probabilities $\hat{P}(Y \in \text{Region}_A \mid X)$ and $\hat{P}(Y \in \text{Region}_B \mid X)$, which are computed using two approaches: (1) Monte Carlo estimation with 125 samples, or (2) numerical integration of the PDF over a 5x5x5 grid via the trapezoidal rule.

The agent acts according to the policy $a^*(X) = \arg\max_{a \in \{A,B,C\}} u_a(X)$ with

$$\begin{aligned} u_A(X) &= 2000\hat{P}(Y \in \text{Region}_A \mid X) - 30\hat{P}(Y \notin \text{Region}_A \mid X) \\ u_B(X) &= 1500\hat{P}(Y \in \text{Region}_B \mid X) - 15\hat{P}(Y \notin \text{Region}_B \mid X) \\ u_C(X) &= -10. \end{aligned}$$

Results. Table H.1 shows two key observations:

1. Using the PDF via numerical integration leads to better decisions (higher utility) than relying on a finite number of samples.
2. The improved calibration from `LR` provides a more accurate PDF, leading to a significant further increase in utility. `HDR-R`, which relies on resampling from the original uncalibrated density, actually harmed decision quality in this task.

This demonstrates a concrete scenario where an explicit, calibrated PDF is not just a theoretical advantage but a practical necessity for optimal decision-making.

H.4.2 Reliability diagrams

Figure H.3 shows reliability diagrams for latent calibration. These diagrams plot the nominal probability levels $\alpha \in [0, 1]$ against the empirical probabilities $\hat{F}_{\hat{U}}(\alpha)$, where $\hat{U} = F_{\rho_{\mathcal{Z}}(Z)}\left(\ell_{\hat{T}}(Y; X)\right)$ are the PIT values computed on the test set. We also report 90% consistency bands, represented by the shaded area around the diagonal, as described by Gneiting et al. (2023). The `BASE` model often exhibits miscalibration (deviations from the diagonal), whereas `LR` consistently aligns closely with the diagonal, demonstrating significantly improved latent calibration.

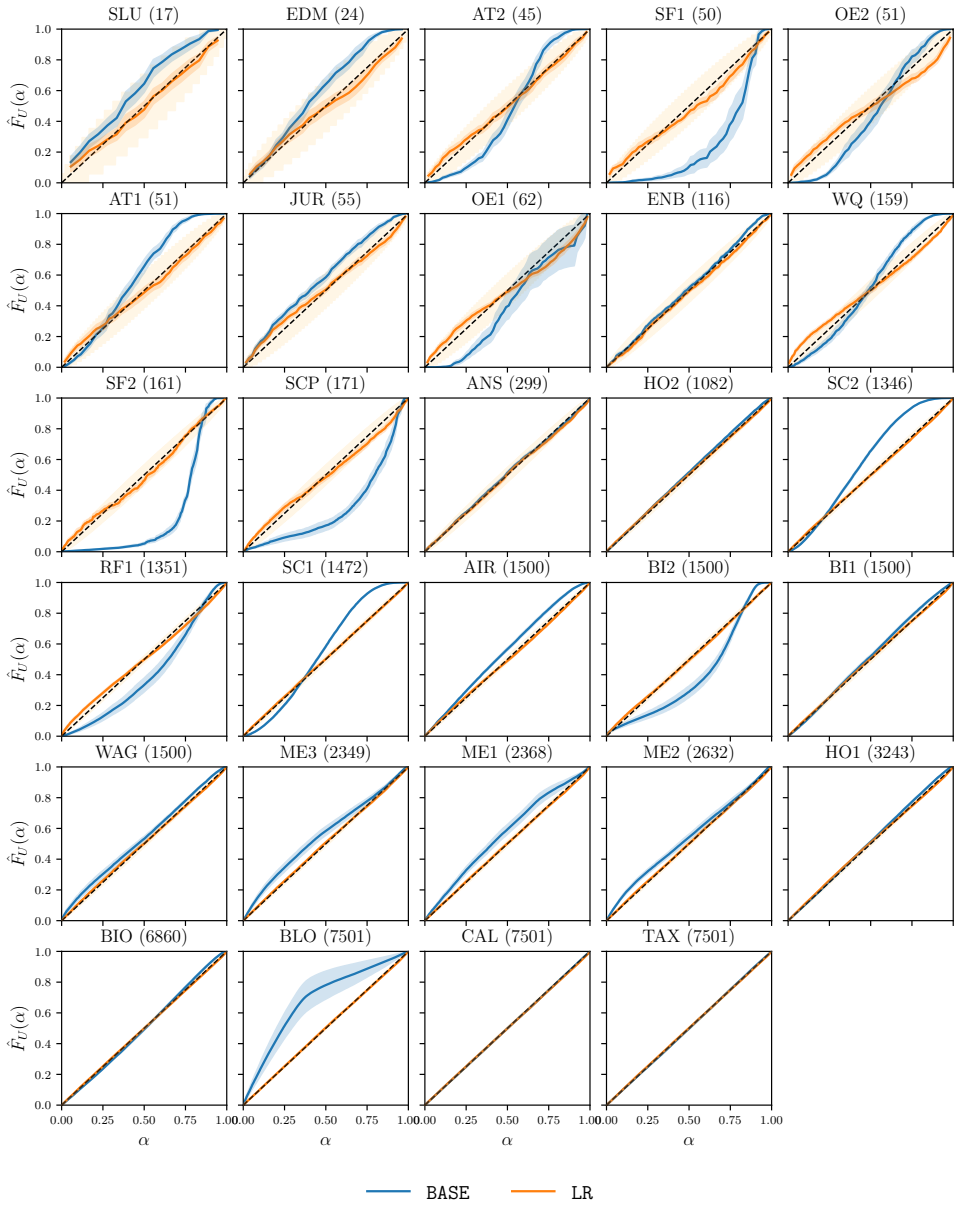


Figure H.3: Latent calibration diagrams

Figure H.4 shows reliability diagrams for HDR calibration. These diagrams plot nominal probability levels α against empirical probabilities $\hat{F}_{\hat{U}}(\alpha)$, where $\hat{U} = \mathrm{HPD}_{\hat{f}_{Y|X}}(Y)$ are the HDR pre-rank values from the test set. Again, the `BASE` model frequently shows miscalibration. Both `LR` and `HDR-R` improve HDR calibration, though for `LR` this improvement is a beneficial side effect rather than a direct optimization target, unlike its consistent improvement of latent calibration shown in Figure H.3.

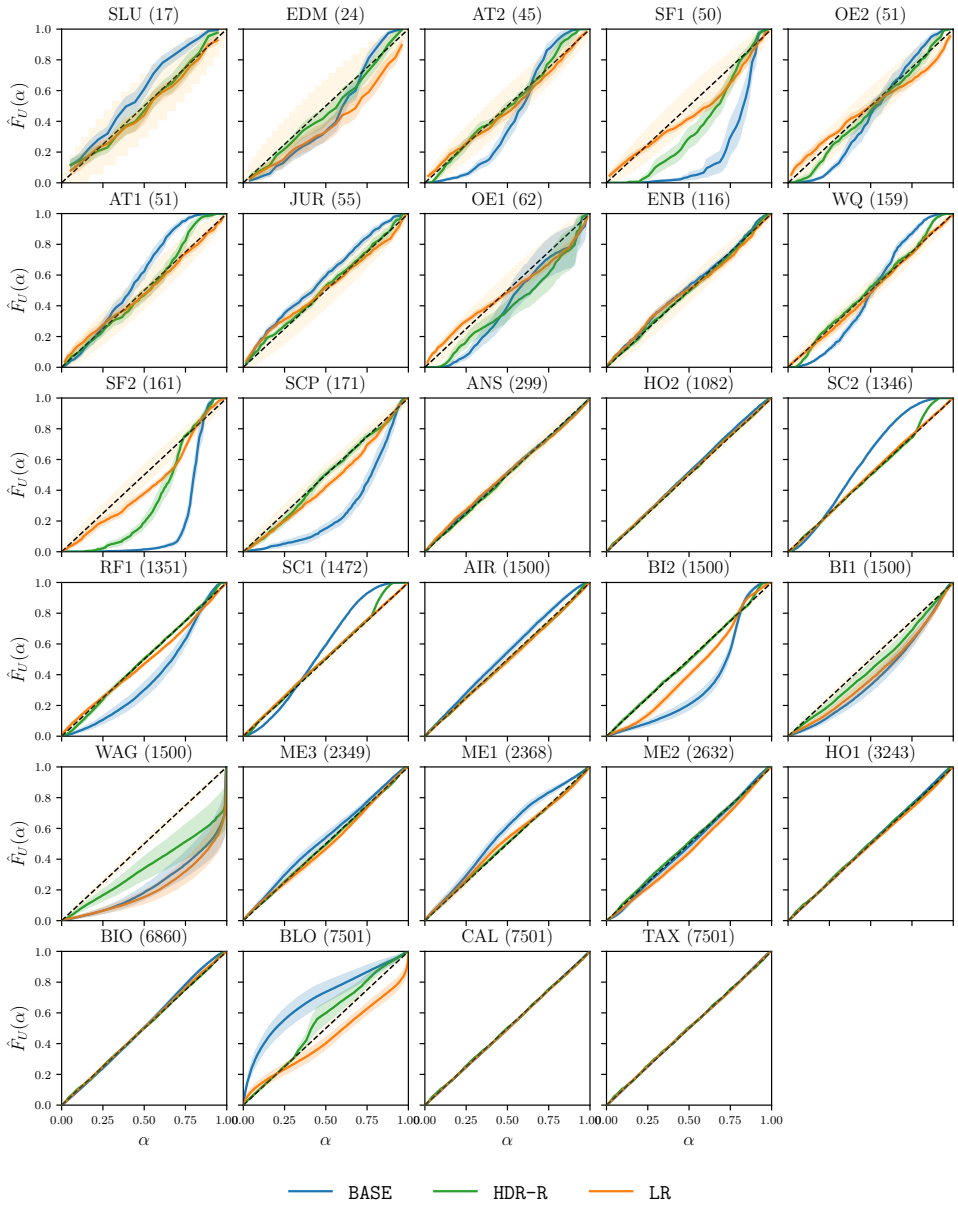


Figure H.4: HDR-calibration diagrams

H.4.3 Examples of predictive distributions on real-world tabular datasets

Figures H.5 and H.6 display examples of predictive PDFs on real-world tabular datasets with two-dimensional outputs ($d = 2$). Each row corresponds to a different dataset. For each dataset, two random test instances, $(x^{(1)}, y^{(1)})$ and $(x^{(2)}, y^{(2)})$ from $\mathcal{D}_{\text{test}}$, are shown.

Columns 1 and 3 show the predictive densities from the uncalibrated base predictor `BASE` (i.e., $\hat{f}_{Y|X=x^{(1)}}(\cdot)$ and $\hat{f}_{Y|X=x^{(2)}}(\cdot)$). Columns 2 and 4 show the corresponding predictive densities from the `LR`-recalibrated model (i.e., $\hat{f}'_{Y|X=x^{(1)}}(\cdot)$ and $\hat{f}'_{Y|X=x^{(2)}}(\cdot)$). All densities are visualized in orange. The true target observations ($y^{(1)}$ and $y^{(2)}$) are marked with a blue dot. The negative log-likelihood of the true target under the respective predictive density is provided in the bottom right corner of each plot. Black contour lines indicate level sets of the PIT of the latent norm ($F_{\rho_{\hat{Z}}(Z')}(\ell_{\hat{T}}(y;x))$ for the `LR` model, and $F_{\rho_{\hat{Z}}(Z)}(\ell_{\hat{T}}(y;x))$ for the `BASE` model) at probability levels 0.01, 0.1, 0.5, and 0.9.

In many cases, when the `BASE` model is already reasonably well-calibrated, `LR` applies a subtle adjustment that is difficult to perceive visually. In other instances, the recalibration effect is more pronounced, visibly altering the shape and spread of the predictive distribution to better align with latent calibration. Note that two-dimensional datasets often benefit from smaller NLL improvements according to Table H.7, suggesting that stronger adjustments should be perceived in higher dimensions.

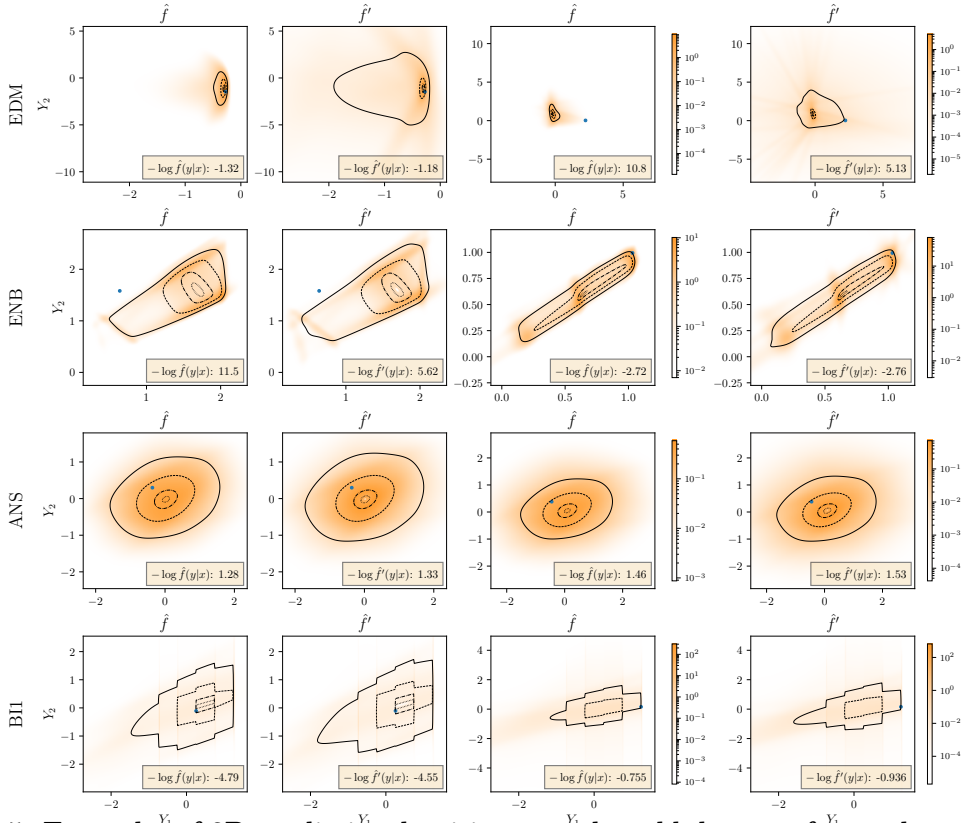


Figure H.5: Examples of 2D predictive densities on real-world datasets for random test points $(x, y) \in \mathcal{D}_{\text{test}}$.

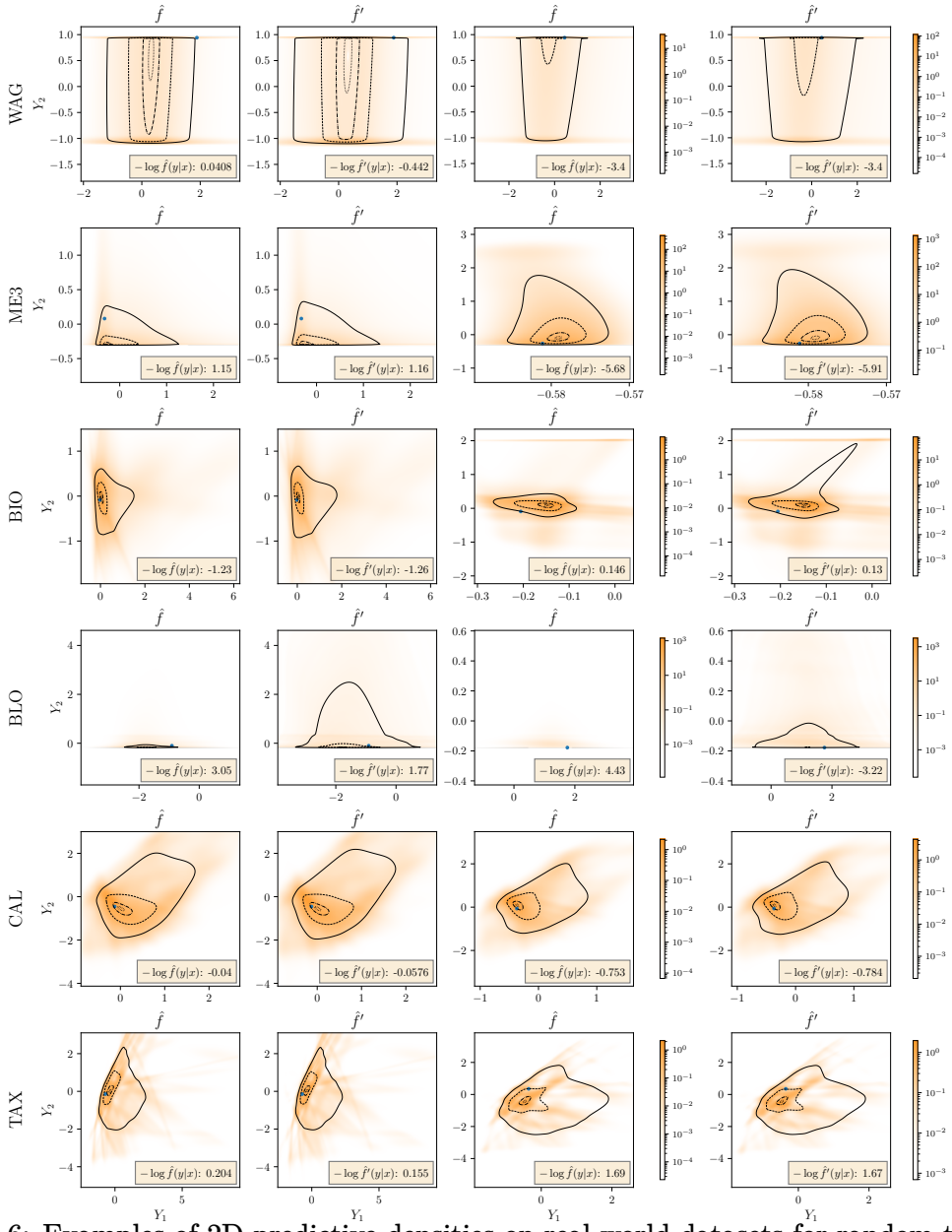


Figure H.6: Examples of 2D predictive densities on real-world datasets for random test points $(x, y) \in \mathcal{D}_{\text{test}}$.

H.4.4 Additional results with convex potential flows

Investigating the NLL performance gain

To investigate the source of the NLL improvement, we consider the decomposition of the NLL of the recalibrated model:

$$-\log \hat{f}'_{Y|X=x}(y) = -\log f_Z(z) - \log \left|\det\left(\nabla_z R(z)\right)\right|^{-1} - \log \left|\det\left(\nabla_y \hat{T}^{-1}(y;x)\right)\right| \tag{H.11}$$

with $z' = \hat{T}^{-1}(y; x)$ and $z = R^{-1}(z')$. The third term is identical for both `BASE` and `LR`. All reported terms are averaged over the test set and over 10 runs.

Table H.2: Analysis of the NLL of `LR` compared to `BASE` with the terms described in (H.11).

	`BASE` $-\log f_Z(z')$	`LR` $-\log f_Z(z)$	`LR` $-\log \lvert\det(\nabla_z R(z))\rvert^{-1}$	`BASE` $-\log \hat{f}_{Y\mid X=x}(y)$	`LR` $-\log \hat{f}'_{Y\mid X=x}(y)$
SLU	5.49	4.35	0.828	2.61	2.29
EDM	4.94	2.81	2.04	-0.0350	-0.123
AT2	11.7	8.65	0.889	4.05	1.86
SF1	11.7	4.33	2.53	4.41	-0.381
OE2	54.6	22.6	0.896	37.6	6.46
AT1	11.6	8.47	1.56	1.63	0.0783
JUR	5.17	4.24	0.638	2.16	1.87
OE1	98.0	22.6	-1.65	80.0	2.93
ENB	3.08	2.80	0.244	-1.08	-1.12
WQ	74.3	19.9	1.51	60.8	7.95
SF2	10.9	4.29	-1.10	-3.37	-11.1
SCP	32.1	4.19	-0.757	20.1	-8.55
ANS	2.82	2.78	0.0700	1.76	1.79
HO2	5.81	5.67	0.143	2.39	2.38
SC2	25.1	22.6	1.80	0.795	0.189
RF1	9.70e+02	11.3	-0.0264	9.54e+02	-4.50
SC1	24.9	22.7	1.39	-1.86	-2.63
AIR	9.15	8.47	0.565	3.03	2.91
BI2	6.99	5.68	-0.208	-11.5	-13.0
BI1	3.10	2.82	0.151	-2.27	-2.40
WAG	3.09	2.82	0.184	-3.25	-3.34
ME3	3.07	2.83	0.146	-2.60	-2.69
ME1	3.21	2.81	0.225	-2.00	-2.17
ME2	3.19	2.82	0.143	-3.00	-3.22
HO1	3.03	2.82	0.144	-0.299	-0.364
BIO	3.08	2.82	0.128	-1.12	-1.25
BLO	9.18	2.82	1.14	3.11	-2.11
CAL	2.84	2.82	0.0255	0.575	0.575
TAX	2.85	2.83	0.0208	1.53	1.53

The table reveals a clear pattern. The NLL improvement from `LR` is primarily driven by the first term. By radially transforming the latent codes z' to new points z that are more consistent with the base density f_Z, the latent density term $-\log f_Z(z)$ is significantly reduced. The recalibration Jacobian (second term) typically adds a small penalty (increases NLL), but this is almost always outweighed by the large gains from the first term. This confirms that `LR` works by finding more "plausible" latent codes for the observed data under the base latent distribution.

Computational efficiency

The difference in computation time can be measured in two aspects:

- For calibration, the computational complexity of `HDR-R` is $O(MFn)$ and `LR` is $O(Rn)$ where $M = 100$ corresponds to the number of samples of `HDR-R` per instance, F the time for the forward mapping $\hat{T}$ and R the time for the reverse mapping $\hat{T}^{-1}$.

- For inference, it is a bit more subtle. Given a test instance x, `HDR-R` requires to sample at least M times ($O(MF)$) to obtain a recalibrated sample, which can be a weakness, e.g., if only one conditional sample is needed. `LR` only incurs a low fixed cost C for evaluating the recalibration map ($O(C + F)$). Thus, the inference time is not directly comparable.

We report the calibration time of `HDR-R` and `LR` in seconds on the largest datasets using the convex potential flow model and averaged over 10 runs.

Table H.3: Calibration times (part 1)

Method	HO2	SC2	RF1	SC1	AIR	BI2	BI1	WAG
`HDR-R`	1.56	2.78	4.90	2.57	4.86	18.80	8.80	15.00
`LR`	0.232	0.185	0.156	0.187	0.149	0.142	0.133	0.133

Table H.4: Calibration times (part 2)

Method	ME3	ME1	ME2	HO1	BIO	BLO	CAL	TAX
`HDR-R`	9.98	12.70	19.30	7.19	37.70	168.00	8.53	10.50
`LR`	0.152	0.153	0.163	0.192	0.290	0.328	0.482	0.324

On CIFAR-10 with TarFlow, the time difference is larger and can be prohibitive for `HDR-R`:

Method	CIFAR-10
`HDR-R`	183182
`LR`	1259

Discriminative ability of the energy score and NLL

For a comprehensive evaluation of `LR`, we report the ES in addition to the NLL. While `LR` often leads to improved NLL, the ES remains largely unchanged. We hypothesize that this stems from the score's fundamental limitations in discriminative ability.

Theoretical considerations. As established in Pinson and Tastu (2013) and corroborated by Alexander et al. (2022), the ES is sensitive to shifts in the mean but notoriously insensitive to misspecifications in variance, correlation, and overall dependency structure. `LR` is a post-hoc procedure that primarily corrects the shape and spread of the predictive distribution. Therefore, the ES is fundamentally ill-suited to capture the specific improvements `LR` provides.

In contrast, the NLL is uniquely suited for this evaluation. As the only local strictly proper scoring rule, its value depends only on the probability density at the precise location of the observed outcome (H. Du, 2021). This locality makes it highly discerning of the very improvements `LR` makes to the distributional shape, which is why we observe significant and consistent NLL reductions.

Empirical illustration. To provide a clear, empirical illustration, we designed a controlled synthetic experiment based on the dataset in Figure 7.1. The goal here is to isolate this specific property of the scoring rules in a setting free from the confounding variables of complex, real-world data.

We use an oracle predictor that knows the true data-generating distribution from Figure 7.1 for everything except the spread around the arc, which is controlled by a standard deviation parameter σ. We then evaluate the predictor's NLL and ES (based on 100 samples) as we vary its estimate of σ. The true value is $\sigma = 0.05$.

Table H.5: Metrics averaged over 10 runs, with standard error

σ	NLL	ES
0.01	$12.01_{0.195}$	$0.8733_{0.00298}$
0.03	$1.274_{0.0222}$	$0.8724_{0.00267}$
0.04	$0.9232_{0.0129}$	$\mathbf{0.8723_{0.00295}}$
0.05 (True)	$\mathbf{0.8557_{0.00870}}$	$0.8740_{0.00171}$
0.06	$0.8781_{0.00629}$	$0.8741_{0.00182}$
0.07	$0.9365_{0.00483}$	$0.8753_{0.00214}$
0.10	$1.161_{0.00272}$	$0.8741_{0.00163}$
0.20	$1.774_{0.00106}$	$0.8782_{0.00176}$

This experiment clearly illustrates the issue:

- The NLL shows a sharp, clear minimum at the true value of $\sigma = 0.05$, correctly identifying the best model.
- The ES remains almost completely flat for a wide range of σ values (from 0.01 to 0.10). It fails to reliably distinguish a model with the correct variance from one that is substantially over- or under-confident.

This insensitivity is so profound that detecting a statistically significant signal with the ES requires an impractically large number of samples. The table below shows that only with 5000 runs does the ES minimum align with the true σ, and even then the differences are minuscule:

Table H.6: Metrics averaged over 5000 runs, with standard error

σ	NLL	ES
0.01	$11.91_{0.00773}$	$0.8754_{0.000139}$
0.03	$1.263_{0.000868}$	$0.8751_{0.000139}$
0.04	$0.9177_{0.000495}$	$0.8751_{0.000139}$
0.05 (True)	$\mathbf{0.8487_{0.000323}}$	$\mathbf{0.8750_{0.000139}}$
0.06	$0.8732_{0.000231}$	$0.8751_{0.000138}$
0.07	$0.9329_{0.000176}$	$0.8751_{0.000138}$
0.10	$1.159_{0.000104}$	$0.8756_{0.000137}$
0.20	$1.774_{6.91e\text{-}05}$	$0.8797_{0.000133}$

This controlled experiment, therefore, proposes an explanation for the insensitivity of the ES to LR.

H.4.5 Additional tables

For reference, Tables H.7 and H.8 provide the precise mean values and standard errors for NLL, Energy Score, L-ECE, and HDR-ECE across all tabular datasets when using convex potential flows as the base predictor. For each metric and dataset, all values that are statistically indistinguishable from the best value according to a Z-test at significance level 0.1 are highlighted in bold.

Table H.7: Full comparative table where the base predictor is a convex potential flow.

	NLL		Energy score		
	BASE	LR	BASE	HDR-R	LR
SLU	**$2.61_{0.19}$**	**$2.29_{0.14}$**	**$0.791_{0.038}$**	**$0.795_{0.033}$**	**$0.785_{0.033}$**
EDM	**$-0.0350_{0.46}$**	**$-0.123_{1.3}$**	**$0.647_{0.049}$**	**$0.648_{0.050}$**	**$0.635_{0.044}$**
AT2	$4.05_{0.88}$	**$1.86_{0.42}$**	**$0.861_{0.044}$**	**$0.870_{0.044}$**	**$0.862_{0.042}$**
SF1	**$4.41_{3.2}$**	**$-0.381_{3.6}$**	**$0.673_{0.086}$**	**$0.639_{0.093}$**	**$0.670_{0.085}$**
OE2	**$37.6_{3.0e+01}$**	**$6.46_{1.3}$**	**$1.25_{0.083}$**	**$1.26_{0.083}$**	**$1.26_{0.085}$**
AT1	$1.63_{0.46}$	**$0.0783_{0.29}$**	**$0.582_{0.032}$**	**$0.587_{0.031}$**	**$0.591_{0.030}$**
JUR	**$2.16_{0.24}$**	**$1.87_{0.15}$**	**$0.617_{0.034}$**	**$0.618_{0.033}$**	**$0.618_{0.034}$**
OE1	**$80.0_{6.9e+01}$**	**$2.93_{0.71}$**	**$1.23_{0.15}$**	**$1.23_{0.15}$**	**$1.17_{0.15}$**
ENB	**$-1.08_{0.11}$**	**$-1.12_{0.10}$**	**$0.249_{0.010}$**	**$0.250_{0.010}$**	**$0.249_{0.010}$**
WQ	**$60.8_{3.7e+01}$**	**$7.95_{3.6}$**	**$2.47_{0.025}$**	**$2.49_{0.024}$**	**$2.47_{0.024}$**
SF2	$-3.37_{3.0}$	**$-11.1_{0.63}$**	**$0.587_{0.044}$**	**$0.593_{0.046}$**	**$0.598_{0.045}$**
SCP	**$20.1_{2.6e+01}$**	**$-8.55_{0.49}$**	**$0.389_{0.094}$**	**$0.383_{0.095}$**	**$0.382_{0.095}$**
ANS	**$1.76_{0.022}$**	**$1.79_{0.020}$**	**$0.529_{0.0052}$**	**$0.531_{0.0047}$**	**$0.529_{0.0053}$**
HO2	**$2.39_{0.034}$**	**$2.38_{0.035}$**	**$0.862_{0.0076}$**	**$0.866_{0.0075}$**	**$0.862_{0.0076}$**
SC2	$0.795_{0.17}$	**$0.189_{0.15}$**	**$1.25_{0.011}$**	**$1.28_{0.012}$**	**$1.26_{0.011}$**
RF1	**$9.54e+02_{6.8e+02}$**	**$-4.50_{1.5}$**	**$0.534_{0.073}$**	**$0.528_{0.071}$**	**$0.529_{0.072}$**
SC1	$-1.86_{0.080}$	**$-2.63_{0.075}$**	**$0.824_{0.0047}$**	**$0.833_{0.0045}$**	**$0.825_{0.0045}$**
AIR	**$3.03_{0.30}$**	**$2.91_{0.30}$**	**$1.17_{0.0086}$**	**$1.19_{0.0090}$**	**$1.18_{0.0084}$**
BI2	**$-11.5_{0.71}$**	**$-13.0_{0.61}$**	**$0.833_{0.013}$**	**$0.848_{0.015}$**	**$0.834_{0.014}$**
BI1	**$-2.27_{0.26}$**	**$-2.40_{0.26}$**	**$0.708_{0.0056}$**	**$0.711_{0.0053}$**	**$0.708_{0.0057}$**
WAG	**$-3.25_{0.31}$**	**$-3.34_{0.32}$**	**$0.802_{0.048}$**	**$0.803_{0.043}$**	**$0.805_{0.046}$**
ME3	**$-2.60_{0.13}$**	**$-2.69_{0.12}$**	**$0.358_{0.0082}$**	**$0.362_{0.0082}$**	**$0.360_{0.0083}$**
ME1	**$-2.00_{0.13}$**	**$-2.17_{0.13}$**	**$0.357_{0.0075}$**	**$0.370_{0.0095}$**	**$0.365_{0.0087}$**
ME2	**$-3.00_{0.12}$**	**$-3.22_{0.13}$**	**$0.361_{0.0041}$**	**$0.362_{0.0039}$**	**$0.362_{0.0040}$**
HO1	**$-0.299_{0.038}$**	**$-0.364_{0.029}$**	**$0.346_{0.0074}$**	**$0.351_{0.0079}$**	**$0.340_{0.012}$**
BIO	$-1.12_{0.072}$	**$-1.25_{0.019}$**	**$0.207_{0.0038}$**	**$0.210_{0.0039}$**	**$0.204_{0.0051}$**
BLO	$3.11_{2.8}$	**$-2.11_{0.25}$**	**$0.305_{0.029}$**	**$0.365_{0.037}$**	$0.473_{0.081}$
CAL	**$0.575_{0.0097}$**	**$0.575_{0.0088}$**	**$0.419_{0.0014}$**	**$0.421_{0.0015}$**	**$0.419_{0.0014}$**
TAX	**$1.53_{0.0069}$**	**$1.53_{0.0068}$**	**$0.692_{0.0019}$**	$0.696_{0.0021}$	**$0.690_{0.0027}$**

Table H.8: Full comparative table where the base predictor is a convex potential flow.

	L-ECE		HDR-ECE		
	BASE	LR	BASE	HDR-R	LR
SLU	**$0.146_{0.026}$**	**$0.106_{0.016}$**	**$0.129_{0.022}$**	**$0.116_{0.014}$**	**$0.102_{0.013}$**
EDM	**$0.122_{0.016}$**	**$0.0905_{0.016}$**	**$0.128_{0.014}$**	**$0.101_{0.019}$**	$0.169_{0.014}$
AT2	$0.129_{0.0076}$	**$0.0637_{0.010}$**	$0.149_{0.0094}$	**$0.0817_{0.0095}$**	**$0.0688_{0.012}$**
SF1	$0.279_{0.026}$	**$0.0701_{0.0082}$**	$0.321_{0.022}$	$0.171_{0.019}$	**$0.0786_{0.0096}$**
OE2	$0.136_{0.011}$	**$0.0785_{0.0076}$**	$0.129_{0.0098}$	**$0.0768_{0.012}$**	**$0.0766_{0.0079}$**
AT1	$0.124_{0.013}$	**$0.0668_{0.0098}$**	$0.125_{0.012}$	$0.0906_{0.0095}$	**$0.0643_{0.012}$**
JUR	$0.0883_{0.011}$	**$0.0520_{0.0052}$**	$0.0866_{0.011}$	**$0.0515_{0.0063}$**	**$0.0537_{0.0049}$**
OE1	$0.197_{0.044}$	**$0.0589_{0.0068}$**	$0.195_{0.044}$	**$0.144_{0.049}$**	**$0.0636_{0.0074}$**
ENB	**$0.0272_{0.0034}$**	**$0.0292_{0.0034}$**	**$0.0375_{0.0049}$**	**$0.0347_{0.0039}$**	**$0.0397_{0.0061}$**
WQ	$0.0883_{0.0048}$	**$0.0479_{0.0044}$**	$0.0975_{0.0032}$	**$0.0443_{0.0060}$**	**$0.0432_{0.0086}$**
SF2	$0.272_{0.0084}$	**$0.0434_{0.0050}$**	$0.312_{0.0061}$	$0.164_{0.015}$	**$0.0797_{0.010}$**
SCP	$0.218_{0.025}$	**$0.0502_{0.0060}$**	$0.223_{0.025}$	**$0.0543_{0.010}$**	**$0.0614_{0.0093}$**
ANS	**$0.0190_{0.0029}$**	**$0.0254_{0.0036}$**	**$0.0199_{0.0030}$**	**$0.0234_{0.0026}$**	**$0.0249_{0.0037}$**
HO2	**$0.0162_{0.0024}$**	**$0.0158_{0.0018}$**	$0.0179_{0.0023}$	**$0.0122_{0.00080}$**	**$0.0121_{0.0017}$**
SC2	$0.101_{0.0034}$	**$0.0122_{0.00087}$**	$0.105_{0.0030}$	$0.0223_{0.00081}$	**$0.0117_{0.0011}$**
RF1	$0.112_{0.021}$	**$0.0260_{0.0049}$**	$0.129_{0.024}$	**$0.0193_{0.0034}$**	$0.0288_{0.0045}$
SC1	$0.0845_{0.0015}$	**$0.0106_{0.0013}$**	$0.0857_{0.0017}$	$0.0229_{0.0011}$	**$0.0116_{0.0016}$**
AIR	$0.0527_{0.0059}$	**$0.0162_{0.0010}$**	$0.0449_{0.0083}$	**$0.0160_{0.0012}$**	$0.0286_{0.0052}$
BI2	$0.122_{0.016}$	**$0.0170_{0.0022}$**	$0.167_{0.019}$	**$0.0219_{0.0023}$**	$0.0731_{0.011}$
BI1	$0.0279_{0.0023}$	**$0.0102_{0.00084}$**	**$0.111_{0.026}$**	**$0.0580_{0.024}$**	**$0.0923_{0.023}$**
WAG	$0.0478_{0.0082}$	**$0.0137_{0.0018}$**	**$0.267_{0.050}$**	**$0.161_{0.071}$**	**$0.290_{0.046}$**
ME3	$0.0732_{0.012}$	**$0.00957_{0.00084}$**	$0.0520_{0.010}$	**$0.00899_{0.00064}$**	$0.0323_{0.0042}$
ME1	$0.0853_{0.010}$	**$0.0103_{0.00078}$**	$0.0779_{0.010}$	**$0.00976_{0.00092}$**	$0.0432_{0.0050}$
ME2	$0.0497_{0.0099}$	**$0.0101_{0.00066}$**	$0.0415_{0.0061}$	**$0.0136_{0.0023}$**	$0.0417_{0.0061}$
HO1	$0.0166_{0.0023}$	**$0.00950_{0.0013}$**	**$0.0103_{0.0014}$**	**$0.00782_{0.00071}$**	**$0.0109_{0.0018}$**
BIO	$0.0178_{0.0019}$	**$0.00561_{0.00076}$**	$0.0220_{0.0027}$	**$0.00667_{0.00051}$**	**$0.00727_{0.0015}$**
BLO	$0.231_{0.036}$	**$0.00735_{0.0010}$**	$0.207_{0.040}$	**$0.0485_{0.024}$**	$0.114_{0.014}$
CAL	$0.00749_{0.00096}$	**$0.00502_{0.00074}$**	$0.00760_{0.00079}$	**$0.00555_{0.00026}$**	**$0.00543_{0.00064}$**
TAX	$0.00949_{0.0013}$	**$0.00522_{0.00053}$**	**$0.00584_{0.00063}$**	**$0.00684_{0.00050}$**	**$0.00561_{0.00086}$**

H.4.6 Results with MAFs

For completeness, this section reports results using a MAF (Papamakarios, Pavlakou, et al., 2017) as the base predictor. The architecture consists of stacked flow layers, where each layer's conditioner is a masked autoencoder (Germain et al., 2015) parameterizing rational quadratic spline transformations (Durkan et al., 2019). We tune hyperparameters using grid search. The number of stacked flows is chosen from $[3, 5, 8]$, the number of hidden units per flow from $[32, 64]$, and the number of hidden layers per flow from $[2, 3]$. The learning rate is selected from $[5 \times 10^{-3}, 10^{-3}]$. Each flow learns a rational quadratic spline transformation.

The findings, illustrated in Figure H.7 and Figure H.8 (and detailed in Tables H.9 and H.10), are consistent with the main tabular results reported in Section 7.5. Specifically, `LR` provides notable improvements in L-ECE, NLL, and HDR-ECE, while achieving an energy score comparable to that of the `BASE` model.

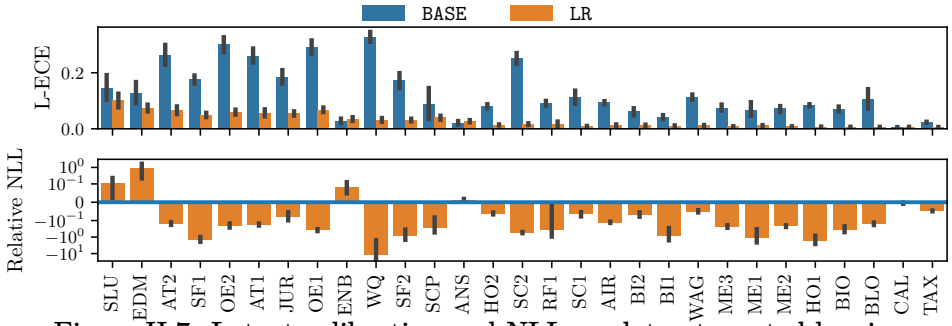


Figure H.7: Latent calibration and NLL on datasets sorted by size.

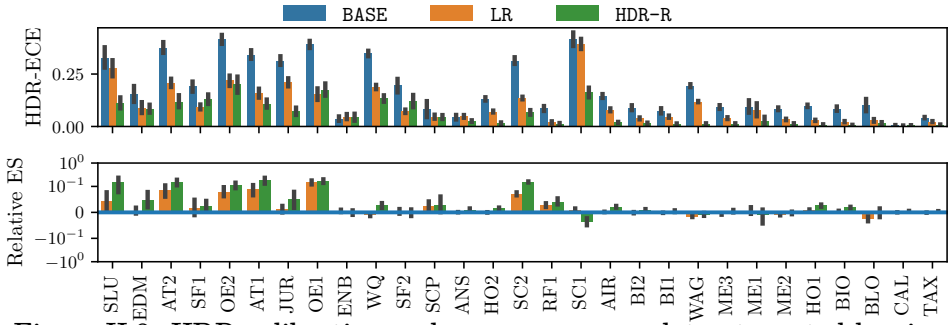


Figure H.8: HDR calibration and energy score on datasets sorted by size.

Table H.9: Full comparative table where the base predictor is a MAF.

	NLL		Energy score		
	BASE	LR	BASE	HDR-R	LR
SLU	**$4.50_{0.37}$**	**$5.08_{0.52}$**	**$0.831_{0.051}$**	$0.945_{0.034}$	**$0.858_{0.025}$**
EDM	**$0.202_{0.30}$**	**$0.590_{0.35}$**	**$0.577_{0.035}$**	**$0.602_{0.033}$**	**$0.578_{0.022}$**
AT2	$7.45_{0.52}$	**$6.22_{0.24}$**	**$0.970_{0.054}$**	$1.11_{0.044}$	**$1.05_{0.029}$**
SF1	$-1.47_{0.39}$	**$-3.42_{0.26}$**	**$0.698_{0.087}$**	**$0.710_{0.085}$**	**$0.700_{0.053}$**
OE2	$22.4_{2.6}$	**$16.8_{0.90}$**	**$1.67_{0.13}$**	**$1.85_{0.13}$**	**$1.79_{0.087}$**
AT1	**$4.63_{0.61}$**	**$3.80_{0.32}$**	**$0.698_{0.049}$**	$0.815_{0.043}$	**$0.754_{0.029}$**
JUR	**$3.99_{0.29}$**	**$3.65_{0.15}$**	**$0.680_{0.037}$**	**$0.710_{0.031}$**	**$0.686_{0.022}$**
OE1	$16.9_{2.7}$	**$9.68_{0.75}$**	**$1.31_{0.13}$**	**$1.51_{0.12}$**	**$1.48_{0.080}$**
ENB	**$-0.939_{0.10}$**	**$-0.877_{0.071}$**	**$0.273_{0.0080}$**	**$0.273_{0.0077}$**	**$0.275_{0.0054}$**
WQ	$0.944_{0.91}$	**$-1.23_{0.57}$**	**$2.50_{0.035}$**	$2.56_{0.030}$	**$2.47_{0.019}$**
SF2	**$-6.21_{1.3}$**	**$-8.72_{0.77}$**	**$0.640_{0.051}$**	**$0.637_{0.048}$**	**$0.641_{0.034}$**
SCP	**$-5.17_{2.2}$**	**$-7.58_{0.51}$**	**$0.392_{0.099}$**	**$0.398_{0.099}$**	**$0.400_{0.069}$**
ANS	**$1.89_{0.025}$**	**$1.91_{0.017}$**	**$0.531_{0.0053}$**	**$0.536_{0.0049}$**	**$0.532_{0.0036}$**
HO2	$3.06_{0.052}$	**$2.87_{0.029}$**	**$0.881_{0.0077}$**	**$0.894_{0.0068}$**	**$0.881_{0.0050}$**
SC2	$1.88_{0.25}$	**$0.936_{0.12}$**	**$1.01_{0.0091}$**	$1.17_{0.0096}$	$1.09_{0.0060}$
RF1	**$-14.3_{1.3}$**	**$-15.5_{0.28}$**	**$0.203_{0.023}$**	**$0.210_{0.023}$**	**$0.208_{0.016}$**
SC1	**$-4.48_{2.9}$**	**$-5.00_{2.0}$**	**$2.62_{0.19}$**	**$2.52_{0.18}$**	**$2.65_{0.14}$**
AIR	$4.29_{0.15}$	**$3.74_{0.10}$**	**$1.21_{0.0097}$**	$1.23_{0.0090}$	**$1.21_{0.0065}$**
BI2	$-11.6_{0.27}$	**$-12.3_{0.16}$**	**$0.831_{0.018}$**	**$0.839_{0.017}$**	**$0.830_{0.011}$**
BI1	**$0.622_{0.12}$**	**$0.443_{0.077}$**	**$0.719_{0.0049}$**	**$0.722_{0.0049}$**	**$0.718_{0.0034}$**
WAG	**$-2.12_{0.089}$**	**$-2.22_{0.060}$**	$0.721_{0.0045}$	**$0.716_{0.0037}$**	**$0.710_{0.0027}$**
ME3	$-1.95_{0.097}$	**$-2.37_{0.049}$**	**$0.401_{0.0080}$**	**$0.403_{0.0078}$**	**$0.398_{0.0054}$**
ME1	**$-1.51_{0.36}$**	**$-1.96_{0.23}$**	**$0.531_{0.066}$**	**$0.517_{0.052}$**	**$0.540_{0.053}$**
ME2	$-1.94_{0.076}$	**$-2.35_{0.045}$**	**$0.412_{0.0047}$**	**$0.412_{0.0045}$**	**$0.409_{0.0033}$**
HO1	$-0.153_{0.045}$	**$-0.323_{0.023}$**	**$0.327_{0.0068}$**	**$0.335_{0.0070}$**	**$0.330_{0.0048}$**
BIO	$-1.10_{0.075}$	**$-1.42_{0.016}$**	**$0.203_{0.0040}$**	**$0.207_{0.0040}$**	**$0.204_{0.0027}$**
BLO	**$-3.37_{0.35}$**	**$-3.85_{0.24}$**	**$0.372_{0.015}$**	**$0.372_{0.018}$**	**$0.362_{0.0093}$**
CAL	**$0.581_{0.0083}$**	**$0.578_{0.0055}$**	**$0.419_{0.0014}$**	**$0.421_{0.0014}$**	**$0.419_{0.00094}$**
TAX	$1.62_{0.0069}$	**$1.54_{0.0042}$**	**$0.696_{0.0023}$**	**$0.698_{0.0021}$**	**$0.695_{0.0015}$**

Table H.10: Full comparative table where the base predictor is a MAF.

	L-ECE		HDR-ECE		
	BASE	LR	BASE	HDR-R	LR
SLU	$0.146_{0.024}$	$\mathbf{0.101_{0.013}}$	$0.328_{0.026}$	$\mathbf{0.110_{0.014}}$	$0.277_{0.020}$
EDM	$0.128_{0.020}$	$\mathbf{0.0734_{0.0064}}$	$0.153_{0.021}$	$\mathbf{0.0849_{0.010}}$	$\mathbf{0.0889_{0.013}}$
AT2	$0.262_{0.019}$	$\mathbf{0.0651_{0.0070}}$	$0.376_{0.014}$	$\mathbf{0.118_{0.016}}$	$0.208_{0.010}$
SF1	$0.176_{0.0083}$	$\mathbf{0.0487_{0.0044}}$	$0.195_{0.013}$	$0.129_{0.012}$	$\mathbf{0.0935_{0.0062}}$
OE2	$0.300_{0.015}$	$\mathbf{0.0595_{0.0048}}$	$0.418_{0.011}$	$\mathbf{0.201_{0.021}}$	$\mathbf{0.220_{0.013}}$
AT1	$0.258_{0.013}$	$\mathbf{0.0566_{0.0069}}$	$0.342_{0.011}$	$\mathbf{0.106_{0.010}}$	$0.159_{0.011}$
JUR	$0.185_{0.013}$	$\mathbf{0.0540_{0.0042}}$	$0.314_{0.011}$	$\mathbf{0.0717_{0.0094}}$	$0.212_{0.010}$
OE1	$0.292_{0.013}$	$\mathbf{0.0676_{0.0043}}$	$0.391_{0.0097}$	$\mathbf{0.174_{0.016}}$	$\mathbf{0.153_{0.015}}$
ENB	$\mathbf{0.0277_{0.0044}}$	$\mathbf{0.0346_{0.0035}}$	$\mathbf{0.0362_{0.0059}}$	$\mathbf{0.0428_{0.0085}}$	$\mathbf{0.0461_{0.0060}}$
WQ	$0.328_{0.0095}$	$\mathbf{0.0312_{0.0037}}$	$0.349_{0.0067}$	$\mathbf{0.134_{0.0083}}$	$0.188_{0.0052}$
SF2	$0.174_{0.014}$	$\mathbf{0.0310_{0.0026}}$	$0.197_{0.018}$	$0.123_{0.016}$	$\mathbf{0.0726_{0.0042}}$
SCP	$0.0885_{0.029}$	$\mathbf{0.0397_{0.0038}}$	$0.0813_{0.020}$	$\mathbf{0.0445_{0.0052}}$	$\mathbf{0.0464_{0.0051}}$
ANS	$\mathbf{0.0212_{0.0032}}$	$\mathbf{0.0272_{0.0020}}$	$0.0443_{0.0055}$	$\mathbf{0.0253_{0.0021}}$	$0.0478_{0.0036}$
HO2	$0.0806_{0.0038}$	$\mathbf{0.0139_{0.00094}}$	$0.131_{0.0043}$	$\mathbf{0.0157_{0.0018}}$	$0.0712_{0.0023}$
SC2	$0.250_{0.011}$	$\mathbf{0.0166_{0.0015}}$	$0.313_{0.0083}$	$\mathbf{0.0672_{0.0061}}$	$0.136_{0.0032}$
RF1	$0.0923_{0.0046}$	$\mathbf{0.0172_{0.0038}}$	$0.0860_{0.0062}$	$\mathbf{0.0139_{0.0012}}$	$\mathbf{0.0189_{0.0028}}$
SC1	$0.112_{0.013}$	$\mathbf{0.00913_{0.00061}}$	$0.417_{0.017}$	$\mathbf{0.163_{0.013}}$	$0.394_{0.013}$
AIR	$0.0936_{0.0028}$	$\mathbf{0.0132_{0.0013}}$	$0.144_{0.0049}$	$\mathbf{0.0193_{0.0012}}$	$0.0790_{0.0037}$
BI2	$0.0607_{0.0068}$	$\mathbf{0.0122_{0.0013}}$	$0.0885_{0.0062}$	$\mathbf{0.0156_{0.0014}}$	$0.0385_{0.0024}$
BI1	$0.0408_{0.0039}$	$\mathbf{0.0105_{0.00081}}$	$0.0739_{0.0064}$	$\mathbf{0.0128_{0.00080}}$	$0.0463_{0.0027}$
WAG	$0.113_{0.0050}$	$\mathbf{0.0118_{0.0010}}$	$0.194_{0.0036}$	$\mathbf{0.0134_{0.00090}}$	$0.119_{0.0015}$
ME3	$0.0744_{0.0058}$	$\mathbf{0.00796_{0.00065}}$	$0.0919_{0.0046}$	$\mathbf{0.0128_{0.0011}}$	$0.0395_{0.0022}$
ME1	$0.0673_{0.013}$	$\mathbf{0.0113_{0.00091}}$	$0.0932_{0.016}$	$\mathbf{0.0274_{0.0093}}$	$0.0766_{0.016}$
ME2	$0.0716_{0.0057}$	$\mathbf{0.00801_{0.00052}}$	$0.0848_{0.0034}$	$\mathbf{0.0135_{0.00087}}$	$0.0333_{0.0018}$
HO1	$0.0843_{0.0019}$	$\mathbf{0.00693_{0.00053}}$	$0.0986_{0.0030}$	$\mathbf{0.00896_{0.00069}}$	$0.0292_{0.0014}$
BIO	$0.0713_{0.0048}$	$\mathbf{0.00615_{0.00056}}$	$0.0850_{0.0054}$	$\mathbf{0.00723_{0.00034}}$	$0.0230_{0.0016}$
BLO	$0.105_{0.019}$	$\mathbf{0.00565_{0.00068}}$	$0.102_{0.016}$	$\mathbf{0.0152_{0.0026}}$	$0.0285_{0.0034}$
CAL	$\mathbf{0.00491_{0.00067}}$	$\mathbf{0.00591_{0.00065}}$	$0.00543_{0.00065}$	$0.00529_{0.00036}$	$\mathbf{0.00416_{0.00027}}$
TAX	$0.0235_{0.00098}$	$\mathbf{0.00563_{0.00050}}$	$0.0424_{0.0017}$	$\mathbf{0.00797_{0.00075}}$	$0.0235_{0.0012}$

H.4.7 Results with Flow Matching

While our paper focuses on normalizing flows, `LR` is fully compatible with flow matching (FM) models (Section 2.2.3), which also learn invertible mappings and assume a known latent distribution. For these models, we tune hyperparameters using grid search. The number of hidden units is chosen from $[32, 64]$, the number of hidden layers from $[2, 3, 5]$, and the learning rate from $[5 \times 10^{-3}, 10^{-3}, 2 \times 10^{-4}]$.

The FM results are aligned with the NF results, with `LR` standing out particularly on the L-ECE and NLL metrics.

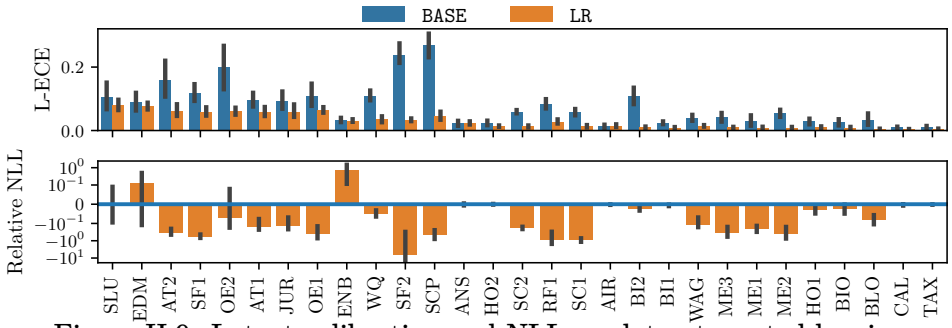


Figure H.9: Latent calibration and NLL on datasets sorted by size.

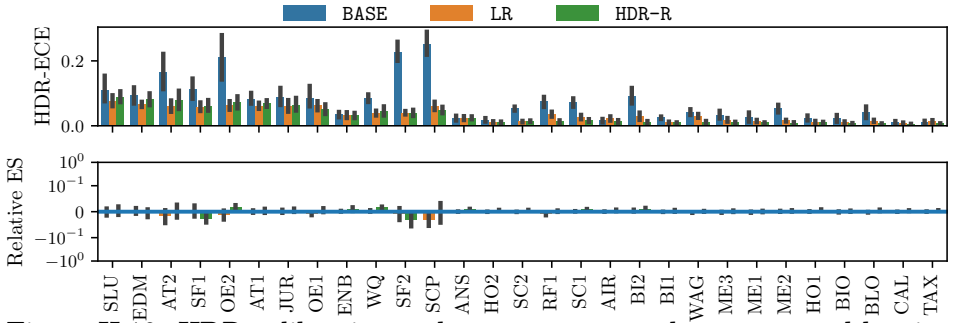


Figure H.10: HDR calibration and energy score on datasets sorted by size.

Table H.11: Full comparative table.

	NLL		Energy score		
	BASE	LR	BASE	HDR-R	LR
SLU	**$2.16_{0.20}$**	**$2.09_{0.13}$**	**$0.756_{0.033}$**	**$0.759_{0.033}$**	**$0.754_{0.029}$**
EDM	**$2.20_{0.18}$**	**$2.36_{0.35}$**	**$0.642_{0.036}$**	**$0.639_{0.037}$**	**$0.643_{0.033}$**
AT2	$4.76_{0.52}$	**$2.93_{0.21}$**	**$0.770_{0.036}$**	**$0.770_{0.032}$**	**$0.756_{0.032}$**
SF1	$2.38_{0.59}$	**$0.901_{0.17}$**	**$0.845_{0.091}$**	**$0.826_{0.092}$**	**$0.841_{0.084}$**
OE2	**$9.53_{1.1}$**	**$8.94_{1.4}$**	**$1.20_{0.066}$**	**$1.22_{0.066}$**	**$1.18_{0.061}$**
AT1	**$1.46_{0.30}$**	**$1.22_{0.24}$**	**$0.571_{0.032}$**	**$0.573_{0.032}$**	**$0.571_{0.031}$**
JUR	**$2.45_{0.25}$**	**$2.02_{0.11}$**	**$0.613_{0.027}$**	**$0.615_{0.027}$**	**$0.613_{0.026}$**
OE1	**$3.99_{1.4}$**	**$2.24_{0.83}$**	**$0.861_{0.057}$**	**$0.866_{0.058}$**	**$0.855_{0.057}$**
ENB	**$0.139_{0.041}$**	**$0.172_{0.035}$**	**$0.313_{0.0042}$**	**$0.316_{0.0047}$**	**$0.313_{0.0041}$**
WQ	$15.0_{0.27}$	**$14.3_{0.25}$**	**$2.40_{0.026}$**	**$2.44_{0.023}$**	**$2.41_{0.025}$**
SF2	**$2.30_{3.0}$**	**$-1.06_{0.36}$**	**$0.668_{0.043}$**	**$0.648_{0.043}$**	**$0.662_{0.041}$**
SCP	$-2.04_{0.59}$	**$-3.30_{0.18}$**	**$0.391_{0.092}$**	**$0.395_{0.097}$**	**$0.386_{0.096}$**
ANS	**$1.78_{0.023}$**	**$1.77_{0.021}$**	**$0.529_{0.0053}$**	**$0.533_{0.0048}$**	**$0.529_{0.0052}$**
HO2	**$2.53_{0.029}$**	**$2.53_{0.029}$**	**$0.852_{0.0069}$**	**$0.856_{0.0066}$**	**$0.852_{0.0068}$**
SC2	**$2.20_{0.17}$**	**$1.82_{0.17}$**	**$1.02_{0.0089}$**	**$1.02_{0.0090}$**	**$1.02_{0.0089}$**
RF1	$0.570_{2.7}$	**$-4.53_{0.20}$**	**$0.367_{0.018}$**	**$0.367_{0.018}$**	**$0.364_{0.018}$**
SC1	$0.525_{0.14}$	**$0.0719_{0.13}$**	**$0.818_{0.0077}$**	**$0.825_{0.0078}$**	**$0.819_{0.0078}$**
AIR	**$4.21_{0.039}$**	**$4.20_{0.040}$**	**$1.16_{0.0090}$**	**$1.17_{0.0089}$**	**$1.16_{0.0090}$**
BI2	**$-4.20_{0.17}$**	**$-4.31_{0.18}$**	**$0.788_{0.013}$**	**$0.796_{0.013}$**	**$0.792_{0.013}$**
BI1	**$2.11_{0.013}$**	**$2.10_{0.010}$**	**$0.703_{0.0057}$**	**$0.706_{0.0053}$**	**$0.703_{0.0056}$**
WAG	**$0.346_{0.036}$**	**$0.308_{0.035}$**	**$0.699_{0.0032}$**	**$0.700_{0.0034}$**	**$0.697_{0.0031}$**
ME3	**$-0.350_{0.073}$**	**$-0.423_{0.069}$**	**$0.392_{0.0087}$**	**$0.393_{0.0090}$**	**$0.391_{0.0086}$**
ME1	**$-0.444_{0.067}$**	**$-0.518_{0.070}$**	**$0.384_{0.0077}$**	**$0.384_{0.0078}$**	**$0.383_{0.0076}$**
ME2	**$-0.342_{0.058}$**	**$-0.422_{0.054}$**	**$0.395_{0.0045}$**	**$0.397_{0.0048}$**	**$0.396_{0.0045}$**
HO1	**$-0.619_{0.021}$**	**$-0.636_{0.018}$**	**$0.214_{0.0036}$**	**$0.216_{0.0038}$**	**$0.215_{0.0036}$**
BIO	**$-0.561_{0.040}$**	**$-0.570_{0.036}$**	**$0.236_{0.0050}$**	**$0.236_{0.0051}$**	**$0.236_{0.0049}$**
BLO	$-1.06_{0.030}$	**$-1.14_{0.026}$**	**$0.258_{0.0031}$**	**$0.259_{0.0030}$**	**$0.257_{0.0030}$**
CAL	**$0.645_{0.0065}$**	**$0.643_{0.0064}$**	**$0.421_{0.0014}$**	**$0.422_{0.0014}$**	**$0.421_{0.0014}$**
TAX	**$1.66_{0.0079}$**	**$1.66_{0.0077}$**	**$0.693_{0.0021}$**	**$0.696_{0.0020}$**	**$0.693_{0.0021}$**

Table H.12: Full comparative table where the base predictor is a MAF.

	L-ECE		HDR-ECE		
	BASE	LR	BASE	HDR-R	LR
SLU	**$0.105_{0.022}$**	**$0.0809_{0.0088}$**	**$0.109_{0.021}$**	**$0.0888_{0.0089}$**	**$0.0761_{0.0092}$**
EDM	**$0.0906_{0.015}$**	**$0.0776_{0.0057}$**	$0.0926_{0.014}$	**$0.0820_{0.0097}$**	**$0.0654_{0.0042}$**
AT2	$0.160_{0.030}$	**$0.0621_{0.010}$**	$0.165_{0.030}$	**$0.0784_{0.016}$**	**$0.0605_{0.0097}$**
SF1	$0.119_{0.015}$	**$0.0593_{0.0072}$**	$0.112_{0.017}$	**$0.0604_{0.0090}$**	**$0.0572_{0.0078}$**
OE2	$0.199_{0.036}$	**$0.0615_{0.0063}$**	$0.210_{0.036}$	**$0.0722_{0.0098}$**	**$0.0626_{0.0072}$**
AT1	$0.0959_{0.012}$	**$0.0597_{0.0074}$**	$0.0831_{0.0089}$	**$0.0679_{0.0057}$**	**$0.0615_{0.0051}$**
JUR	$0.0918_{0.015}$	**$0.0596_{0.011}$**	**$0.0865_{0.014}$**	**$0.0639_{0.011}$**	**$0.0590_{0.010}$**
OE1	$0.109_{0.019}$	**$0.0647_{0.0049}$**	$0.0863_{0.017}$	**$0.0514_{0.0078}$**	**$0.0618_{0.0077}$**
ENB	**$0.0330_{0.0037}$**	**$0.0314_{0.0021}$**	**$0.0340_{0.0044}$**	**$0.0312_{0.0042}$**	**$0.0318_{0.0048}$**
WQ	$0.110_{0.0086}$	**$0.0349_{0.0049}$**	$0.0842_{0.0061}$	**$0.0437_{0.0078}$**	**$0.0385_{0.0042}$**
SF2	$0.239_{0.016}$	**$0.0339_{0.0023}$**	$0.226_{0.015}$	**$0.0397_{0.0051}$**	**$0.0400_{0.0030}$**
SCP	$0.268_{0.020}$	**$0.0457_{0.0070}$**	$0.253_{0.019}$	**$0.0483_{0.0053}$**	**$0.0607_{0.0067}$**
ANS	**$0.0222_{0.0042}$**	**$0.0236_{0.0027}$**	**$0.0231_{0.0040}$**	**$0.0244_{0.0025}$**	**$0.0236_{0.0033}$**
HO2	$0.0232_{0.0038}$	**$0.0138_{0.0011}$**	$0.0180_{0.0028}$	**$0.0109_{0.0011}$**	**$0.0116_{0.0013}$**
SC2	$0.0590_{0.0028}$	**$0.0129_{0.0017}$**	$0.0532_{0.0032}$	**$0.0142_{0.0011}$**	**$0.0127_{0.0012}$**
RF1	$0.0843_{0.0078}$	**$0.0273_{0.0037}$**	$0.0740_{0.0082}$	**$0.0125_{0.0021}$**	$0.0353_{0.0039}$
SC1	$0.0573_{0.0056}$	**$0.0133_{0.0020}$**	$0.0720_{0.0066}$	**$0.0156_{0.0021}$**	$0.0267_{0.0037}$
AIR	**$0.0130_{0.0027}$**	**$0.0151_{0.0023}$**	**$0.0158_{0.0027}$**	**$0.0132_{0.0021}$**	$0.0220_{0.0035}$
BI2	$0.109_{0.014}$	**$0.00979_{0.0014}$**	$0.0916_{0.013}$	**$0.0112_{0.0016}$**	$0.0288_{0.0059}$
BI1	$0.0243_{0.0023}$	**$0.00924_{0.00093}$**	$0.0249_{0.0020}$	**$0.00927_{0.00061}$**	**$0.0101_{0.0013}$**
WAG	$0.0407_{0.0051}$	**$0.0144_{0.0013}$**	$0.0419_{0.0050}$	**$0.0120_{0.0014}$**	$0.0307_{0.0031}$
ME3	$0.0412_{0.0077}$	**$0.00965_{0.0011}$**	$0.0330_{0.0065}$	**$0.0104_{0.00098}$**	$0.0176_{0.0029}$
ME1	$0.0299_{0.0086}$	**$0.00915_{0.0014}$**	$0.0251_{0.0075}$	**$0.00935_{0.00058}$**	$0.0143_{0.0020}$
ME2	$0.0537_{0.0060}$	**$0.00883_{0.0013}$**	$0.0541_{0.0063}$	**$0.00892_{0.00094}$**	$0.0156_{0.0014}$
HO1	$0.0285_{0.0046}$	**$0.00990_{0.0016}$**	$0.0239_{0.0043}$	**$0.00974_{0.0011}$**	**$0.0119_{0.0016}$**
BIO	$0.0253_{0.0053}$	**$0.00766_{0.0019}$**	$0.0221_{0.0058}$	**$0.00726_{0.00062}$**	$0.0102_{0.0016}$
BLO	$0.0346_{0.010}$	**$0.00498_{0.00047}$**	$0.0402_{0.0099}$	**$0.00648_{0.00050}$**	$0.0153_{0.0018}$
CAL	$0.00951_{0.0014}$	**$0.00395_{0.00043}$**	$0.0112_{0.0018}$	**$0.00547_{0.00030}$**	$0.00746_{0.0011}$
TAX	$0.0110_{0.0018}$	**$0.00505_{0.00059}$**	$0.0122_{0.0017}$	**$0.00660_{0.00054}$**	$0.0133_{0.0022}$

H.4.8 Results with a misspecified convex potential flow

Tables H.13 and H.14 along with Figure H.11 and Figure H.12 present results for a deliberately misspecified convex potential flow. This misspecification was induced by training the base predictor for only two epochs, ensuring it has low predictive accuracy and is likely poorly calibrated.

We observe that, in this additional scenario, LR also leads to improved L-ECE and NLL on most datasets, indicating enhanced predictive accuracy compared to the BASE misspecified model. LR achieves similar or improved HDR-ECE and ES compared to HDR-R. These results highlight LR's ability to improve misspecified base predictors.

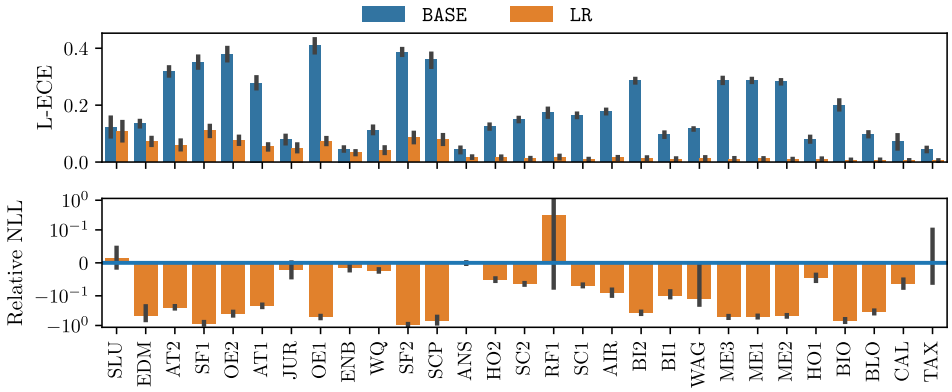


Figure H.11: Latent calibration and NLL on datasets sorted by size.

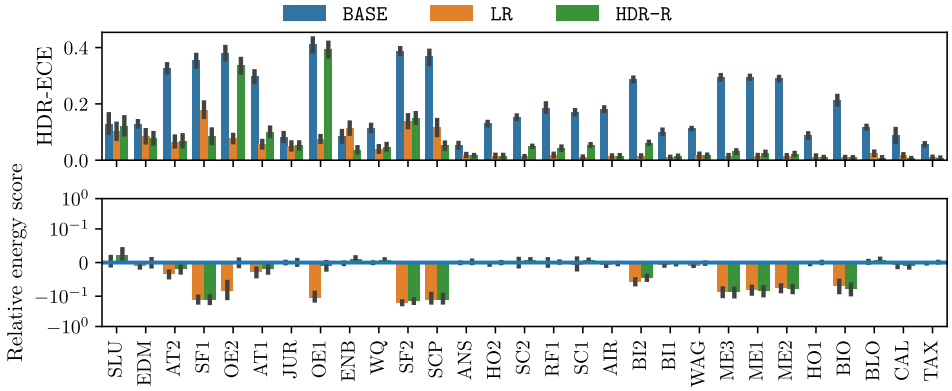


Figure H.12: HDR calibration and energy score on datasets sorted by size.

Table H.13: Full comparative table where the base predictor is a misspecified convex potential flow.

	NLL		Energy score		
	BASE	LR	BASE	HDR-R	LR
SLU	**$3.86_{0.13}$**	**$3.93_{0.16}$**	**$1.01_{0.043}$**	**$1.03_{0.050}$**	**$1.01_{0.047}$**
EDM	$2.84_{0.059}$	**$1.53_{0.32}$**	**$0.880_{0.028}$**	**$0.879_{0.029}$**	**$0.871_{0.029}$**
AT2	$7.99_{0.29}$	**$6.03_{0.14}$**	**$1.39_{0.051}$**	**$1.36_{0.051}$**	**$1.34_{0.054}$**
SF1	$4.05_{0.39}$	**$0.736_{0.22}$**	**$0.857_{0.075}$**	**$0.754_{0.083}$**	**$0.753_{0.079}$**
OE2	$23.1_{2.2}$	**$13.5_{0.80}$**	**$2.24_{0.14}$**	**$2.24_{0.15}$**	**$2.07_{0.16}$**
AT1	$7.68_{0.38}$	**$5.99_{0.26}$**	**$1.42_{0.070}$**	**$1.40_{0.078}$**	**$1.39_{0.076}$**
JUR	**$4.19_{0.13}$**	**$4.09_{0.075}$**	**$1.08_{0.034}$**	**$1.08_{0.036}$**	**$1.08_{0.034}$**
OE1	$23.8_{2.1}$	**$11.1_{0.65}$**	**$2.14_{0.13}$**	**$2.12_{0.14}$**	**$1.91_{0.15}$**
ENB	**$2.05_{0.093}$**	**$2.02_{0.090}$**	**$0.815_{0.020}$**	**$0.823_{0.020}$**	**$0.814_{0.019}$**
WQ	$20.0_{0.15}$	**$19.6_{0.15}$**	**$2.59_{0.027}$**	**$2.61_{0.028}$**	**$2.59_{0.027}$**
SF2	$4.71_{0.55}$	**$0.298_{0.25}$**	$0.821_{0.038}$	**$0.703_{0.039}$**	**$0.688_{0.038}$**
SCP	$5.06_{2.1}$	**$1.25_{0.36}$**	**$0.738_{0.10}$**	**$0.657_{0.11}$**	**$0.658_{0.11}$**
ANS	**$2.61_{0.021}$**	**$2.61_{0.020}$**	**$0.817_{0.0089}$**	**$0.819_{0.0087}$**	**$0.817_{0.0090}$**
HO2	$5.46_{0.020}$	**$5.18_{0.021}$**	**$1.23_{0.0063}$**	**$1.23_{0.0070}$**	**$1.22_{0.0067}$**
SC2	$22.2_{0.12}$	**$20.8_{0.12}$**	**$2.67_{0.019}$**	**$2.69_{0.020}$**	**$2.66_{0.021}$**
RF1	**$10.8_{0.84}$**	**$16.8_{7.5}$**	**$1.69_{0.026}$**	**$1.70_{0.026}$**	**$1.69_{0.032}$**
SC1	$21.4_{0.099}$	**$20.0_{0.10}$**	**$2.56_{0.023}$**	**$2.58_{0.024}$**	**$2.54_{0.023}$**
BI2	$5.62_{0.10}$	**$3.51_{0.085}$**	$1.08_{0.016}$	**$1.03_{0.016}$**	**$1.02_{0.016}$**
BI1	$2.66_{0.030}$	**$2.39_{0.020}$**	**$0.779_{0.0057}$**	**$0.776_{0.0057}$**	**$0.773_{0.0055}$**
AIR	$8.33_{0.068}$	**$7.58_{0.044}$**	**$1.50_{0.0089}$**	**$1.50_{0.0089}$**	**$1.49_{0.0090}$**
WAG	$2.78_{0.0069}$	**$2.43_{0.17}$**	$0.876_{0.0029}$	$0.875_{0.0027}$	**$0.868_{0.0029}$**
ME3	$2.31_{0.045}$	**$1.12_{0.046}$**	$0.602_{0.010}$	**$0.548_{0.0087}$**	**$0.549_{0.0091}$**
ME1	$2.20_{0.046}$	**$1.10_{0.048}$**	$0.588_{0.0092}$	**$0.539_{0.0073}$**	**$0.540_{0.0073}$**
ME2	$2.15_{0.027}$	**$1.13_{0.038}$**	$0.591_{0.0055}$	**$0.545_{0.0059}$**	**$0.546_{0.0058}$**
HO1	$2.46_{0.029}$	**$2.34_{0.030}$**	**$0.743_{0.0068}$**	**$0.743_{0.0072}$**	**$0.741_{0.0071}$**
BIO	$0.963_{0.14}$	**$0.302_{0.046}$**	$0.335_{0.0085}$	**$0.308_{0.0058}$**	**$0.311_{0.0062}$**
CAL	$1.37_{0.043}$	**$1.28_{0.035}$**	**$0.472_{0.0061}$**	**$0.468_{0.0050}$**	**$0.468_{0.0052}$**
BLO	$1.21_{0.033}$	**$0.785_{0.036}$**	**$0.685_{0.0024}$**	$0.691_{0.0030}$	**$0.687_{0.0029}$**
TAX	**$2.29_{0.019}$**	**$2.31_{0.11}$**	**$0.724_{0.0024}$**	**$0.725_{0.0024}$**	**$0.723_{0.0024}$**

Table H.14: Full comparative table where the base predictor is a misspecified convex potential flow.

	L-ECE		HDR-ECE		
	BASE	LR	BASE	HDR-R	LR
SLU	**$0.120_{0.018}$**	**$0.107_{0.017}$**	**$0.127_{0.018}$**	**$0.119_{0.017}$**	**$0.101_{0.015}$**
EDM	$0.135_{0.0052}$	**$0.0734_{0.0068}$**	$0.128_{0.0051}$	**$0.0778_{0.0097}$**	**$0.0846_{0.012}$**
AT2	$0.319_{0.0075}$	**$0.0593_{0.0083}$**	$0.327_{0.0078}$	**$0.0673_{0.011}$**	**$0.0642_{0.0095}$**
SF1	$0.351_{0.011}$	**$0.110_{0.0094}$**	$0.354_{0.011}$	**$0.0838_{0.012}$**	$0.177_{0.014}$
OE2	$0.379_{0.012}$	**$0.0773_{0.0067}$**	$0.379_{0.012}$	$0.335_{0.013}$	**$0.0769_{0.0071}$**
AT1	$0.278_{0.010}$	**$0.0533_{0.0051}$**	$0.297_{0.011}$	$0.100_{0.0080}$	**$0.0565_{0.0063}$**
JUR	$0.0785_{0.0070}$	**$0.0494_{0.0064}$**	$0.0819_{0.0078}$	**$0.0518_{0.0062}$**	**$0.0501_{0.0068}$**
OE1	$0.410_{0.013}$	**$0.0733_{0.0057}$**	$0.410_{0.013}$	$0.395_{0.013}$	**$0.0749_{0.0054}$**
ENB	$0.0458_{0.0030}$	**$0.0326_{0.0028}$**	$0.0854_{0.0099}$	**$0.0367_{0.0049}$**	$0.114_{0.010}$
WQ	$0.112_{0.0060}$	**$0.0405_{0.0054}$**	$0.114_{0.0060}$	**$0.0467_{0.0051}$**	**$0.0391_{0.0053}$**
SF2	$0.386_{0.0057}$	**$0.0869_{0.0087}$**	$0.387_{0.0057}$	**$0.150_{0.0090}$**	**$0.139_{0.011}$**
SCP	$0.361_{0.013}$	**$0.0808_{0.0082}$**	$0.369_{0.013}$	**$0.0517_{0.0063}$**	$0.115_{0.014}$
ANS	$0.0428_{0.0043}$	**$0.0172_{0.0012}$**	$0.0524_{0.0047}$	**$0.0162_{0.0011}$**	**$0.0183_{0.0019}$**
HO2	$0.125_{0.0038}$	**$0.0155_{0.0019}$**	$0.130_{0.0035}$	**$0.0155_{0.0016}$**	**$0.0156_{0.0020}$**
SC2	$0.150_{0.0030}$	**$0.0123_{0.00093}$**	$0.153_{0.0032}$	$0.0491_{0.00096}$	**$0.0124_{0.00095}$**
RF1	$0.173_{0.0076}$	**$0.0172_{0.0024}$**	$0.186_{0.0081}$	$0.0422_{0.0030}$	**$0.0180_{0.0025}$**
SC1	$0.164_{0.0033}$	**$0.00888_{0.0010}$**	$0.170_{0.0040}$	$0.0541_{0.0016}$	**$0.0103_{0.0014}$**
BI2	$0.286_{0.0034}$	**$0.0133_{0.0014}$**	$0.287_{0.0034}$	$0.0616_{0.0019}$	**$0.0146_{0.0013}$**
BI1	$0.0961_{0.0039}$	**$0.00962_{0.0014}$**	$0.100_{0.0040}$	$0.0126_{0.0016}$	**$0.0102_{0.00095}$**
AIR	$0.178_{0.0034}$	**$0.0146_{0.0016}$**	$0.181_{0.0033}$	**$0.0151_{0.0013}$**	**$0.0140_{0.0016}$**
WAG	$0.116_{0.0011}$	**$0.0127_{0.0021}$**	$0.113_{0.0011}$	**$0.0158_{0.0019}$**	**$0.0166_{0.0033}$**
ME3	$0.287_{0.0047}$	**$0.0105_{0.0016}$**	$0.294_{0.0040}$	$0.0304_{0.0025}$	**$0.0140_{0.0022}$**
ME1	$0.287_{0.0029}$	**$0.0110_{0.0011}$**	$0.295_{0.0027}$	$0.0237_{0.0031}$	**$0.0149_{0.0020}$**
ME2	$0.282_{0.0031}$	**$0.00893_{0.00095}$**	$0.290_{0.0030}$	$0.0220_{0.0021}$	**$0.0144_{0.0017}$**
HO1	$0.0801_{0.0046}$	**$0.0102_{0.0012}$**	$0.0880_{0.0041}$	**$0.00989_{0.00082}$**	**$0.0120_{0.0024}$**
BIO	$0.200_{0.0086}$	**$0.00695_{0.00065}$**	$0.213_{0.0080}$	**$0.00985_{0.00048}$**	**$0.00916_{0.0013}$**
CAL	$0.0711_{0.013}$	**$0.00549_{0.00046}$**	$0.0883_{0.013}$	**$0.00681_{0.00041}$**	$0.0169_{0.0024}$
BLO	$0.0972_{0.0039}$	**$0.00657_{0.00092}$**	$0.117_{0.0025}$	**$0.00843_{0.00092}$**	$0.0241_{0.0033}$
TAX	$0.0444_{0.0032}$	**$0.00490_{0.00060}$**	$0.0565_{0.0020}$	**$0.00804_{0.00053}$**	**$0.00918_{0.0015}$**

List of Figures

List of Tables