Probabilistic Calibration by Design for Neural Network Regression

Victor Dheur **Souhaib Ben Taieb**
Department of Computer Science,
University of Mons, Belgium

Abstract

Generating calibrated and sharp neural network predictive distributions for regression problems is essential for optimal decision-making in many real-world applications. To address the miscalibration issue of neural networks, various methods have been proposed to improve calibration, including post-hoc methods that adjust predictions after training and regularization methods that act during training. While post-hoc methods have shown better improvement in calibration compared to regularization methods, the post-hoc step is completely independent of model training. We introduce a novel end-to-end model training procedure called Quantile Recalibration Training, integrating post-hoc calibration directly into the training process without additional parameters. We also present a unified algorithm that includes our method and other post-hoc and regularization methods, as particular cases. We demonstrate the performance of our method in a large-scale experiment involving 57 tabular regression datasets, showcasing improved predictive accuracy while maintaining calibration. We also conduct an ablation study to evaluate the significance of different components within our proposed method, as well as an in-depth analysis of the impact of the base model and different hyperparameters on predictive accuracy.

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

1 INTRODUCTION

Critical decisions depend on the predictions made by neural networks in many applications such as medical diagnostics and autonomous driving (Begoli et al., 2019; Michelmore et al., 2018). To make decisions effectively, it is often crucial to quantify predictive uncertainty accurately (Gawlikowski et al., 2021; Abdar et al., 2021). Yet, neural networks might exhibit miscalibration (Guo et al., 2017).

We focus on regression models that output a predictive distribution. Central to our study, *probabilistic calibration*[1] (Gneiting et al., 2007) is an important property that states that all quantiles must be calibrated. This implies that the predicted 90% quantiles should exceed 90% of the corresponding realizations.

Several methods have been proposed to improve probabilistic calibration and they can be divided into two main categories. Post-hoc methods such as Quantile Recalibration (Kuleshov et al., 2018) act after training a base model and transform the predictions based on a separate calibration dataset. Regularization methods act during training and add a regularization term that penalizes calibration (Chung et al., 2021). Empirical evidence suggests that post-hoc methods outperform regularization methods in terms of calibration within the context of regression (Dheur and Ben Taieb, 2023). This superiority has been attributed to the finite-sample guarantee from which post-hoc methods benefit.

This paper introduces a novel method called *Quantile Recalibration Training* that seamlessly integrates post-hoc calibration into the training process, resulting in an end-to-end approach. Our method leverages the concept of minimizing the sharpness of predictions while ensuring calibration (Gneiting et al., 2007). By minimizing the negative log-likelihood (NLL), our approach achieves the desired sharpness, while simultaneously ensuring calibration at each training step using a

[1] In this paper, we refer to probabilistic calibration as calibration to simplify terminology.

dedicated calibration dataset. Recalibration Training stands apart from other regularization methods by offering improvements in both the NLL and calibration of the final model. Our approach aligns with the recommendation made by Wang et al. (2021) to view model training and post-hoc calibration as an integrated framework rather than treating them as separate steps. The code base, available at `https://github.com/Vekteur/quantile-recalibration-training`, 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 *Quantile Recalibration Training* (see Section 3). We also propose an algorithm which unifies our Quantile Recalibration Training 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 5).

3. We provide an in-depth analysis of the impact of the base model and different hyperparameters on predictive accuracy and calibration. We also conduct an ablation study to evaluate the significance of different components within our proposed method (see Section 6).

2 BACKGROUND ON PROBABILISTIC CALIBRATION

We consider a univariate regression problem where a target variable $Y \in \mathcal{Y}$ depends on an input variable $X \in \mathcal{X}$, where $\mathcal{X}$ is the input space and $\mathcal{Y} \subseteq \mathbb{R}$ is the target space. Our goal is to approximate the conditional distribution $P_{Y|X}$ based on i.i.d. training data $\mathcal{D} = \{ (X_i, Y_i) \}_{i=1}^N$ where $(X_i, Y_i) \sim P_{X,Y} = P_{Y|X} P_X$.

A probabilistic predictor, denoted as $F_\theta : \mathcal{X} \to \mathcal{F}$ is defined by its parameters θ from the parameter space Θ. This function maps an input $x \in \mathcal{X}$ to a predictive cumulative distribution function (CDF) $F_\theta(\cdot \mid x)$ in the space $\mathcal{F}$ of distributions over $\mathbb{R}$. This CDF has an associated probability density function (PDF) given by $f_\theta(\cdot \mid x)$.

Probabilistic calibration Given a possibly miscalibrated CDF F_θ, let $Z = F_\theta(Y \mid X) \in [0, 1]$ denote the probability integral transform (PIT) of Y conditional on X and $F_Z(\alpha) = \Pr(Z \leq \alpha)$ the corresponding CDF. The model F_θ is probabilistically calibrated (also known as PIT-calibrated) if

$$F_Z(\alpha) = \alpha \quad \forall \alpha \in [0, 1]. \tag{1}$$

The CDF F_Z is usually estimated from data using the empirical CDF, that we denote $\Phi_\theta^{\text{EMP}}(\alpha) = \frac{1}{N} \sum_{i=1}^N \mathbb{1}(Z_i \leq \alpha)$, where $Z_i = F_\theta(Y_i \mid X_i)$. We measure probabilitic calibration using the probabilistic calibration error (PCE), defined as

$$\text{PCE}(F_\theta) = \frac{1}{M} \sum_{j=1}^M \left| \alpha_j - \Phi_\theta^{\text{EMP}}(\alpha_j) \right|, \tag{2}$$

where $0 < \alpha_1 < \cdots < \alpha_M < 1$ are equidistant quantile levels. The number of levels M is fixed at 100 in the paper. In essence, PCE computes the discrepancy between the r.h.s and l.h.s. of (1) for multiple values of α.

Quantile recalibration *Quantile Recalibration* (QR, Kuleshov et al., 2018) computes a probabilistically calibrated CDF $F'_\theta = F_Z \circ F_\theta$, where F_Z is estimated from data. In fact, for each quantile level $\alpha \in [0, 1]$, we have:

$$\begin{aligned} \Pr(F'_\theta(Y \mid X) \leq \alpha) &= \Pr(F_\theta(Y \mid X) \leq F_Z^{-1}(\alpha)) && (3) \\ &= F_Z(F_Z^{-1}(\alpha)) && (4) \\ &= \alpha, && (5) \end{aligned}$$

which shows that F'_θ is calibrated.

Calibration map The estimator of F_Z, called a calibration map, can be the empirical CDF Φ_θ^{EMP} computed from the PITs $Z'_i = F_\theta(Y'_i \mid X'_i)$ of a separate i.i.d. calibration dataset $\mathcal{D}' = \{ (X'_i, Y'_i) \}_{i=1}^{N'}$.

Since Φ_θ^{EMP} is not differentiable, the resulting calibrated CDF F'_θ is not differentiable either. Dheur and Ben Taieb (2023) proposed to compute a differentiable calibration map

$$\Phi_\theta^{\text{KDE}}(\alpha) = \frac{1}{N'} \sum_{i=1}^{N'} F_{\text{Log}}(\alpha; Z'_i, b^2 N'^{-2/5}), \tag{6}$$

based on kernel density estimation (KDE). This corresponds to a mixture of logistic CDFs F_{Log} with means $Z'_1, \ldots, Z'_N$ and a variance $b^2 N'^{-2/5}$ following Scott's rule (Scott, 1979). The bandwidth $b > 0$ is a hyperparameter controlling the smoothness of the calibration map. Note that Φ_θ^{KDE} converges to Φ_θ^{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:

$$\Pr(\Phi_\theta^{\mathrm{DCP}}(F_\theta(Y \mid X)) \leq \alpha) = \frac{\lceil (N'+1)\alpha \rceil}{N'+1} \approx \alpha, \quad (7)$$

where $\Phi_\theta^{\mathrm{DCP}}(\alpha) = \frac{1}{N'+1} \sum_{i=1}^{N'} \mathbb{1}(Z_i' \leq \alpha)$ 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 $\Phi_\theta^{\mathrm{EMP}}$ and $\Phi_\theta^{\mathrm{KDE}}$. We note that the probability in (7) is also taken over the calibration dataset $\mathcal{D}'$.

Quantile Regularization Recently, there has been a surge of interest in regularization strategies for calibration based on differentiable objectives that are optimized during training (see Section 4). *Quantile Regularization* (QREG, Utpala and Rai, 2020) minimizes a loss function of the form

$$-\frac{1}{N} \sum_{i=1}^{N} \log f_\theta(Y_i \mid X_i) + \lambda \mathcal{R}_{\mathrm{QREG}}(\theta), \quad (8)$$

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 Z and a uniformly distributed random variable U. This reduces to maximizing the differential entropy $H(Z)$ of Z since $D_{\mathrm{KL}}(Z \parallel U) = -H(Z)$.

Utpala and Rai, 2020 propose to estimate $H(Z)$ using sample-spacing entropy estimation (Vasicek, 1976):

$$\mathcal{R}_{\mathrm{QREG}}(\theta) \quad (9)$$

$$= \frac{1}{N-k} \sum_{i=1}^{N-k} \log \left[\frac{N+1}{k} (Z_{(i+k)} - Z_{(i)}) \right] \quad (10)$$

$$\approx -H(Z), \quad (11)$$

where k is a hyperparameter such that $1 \leq k \leq N$ and $Z_{(i)}$ is the ith order statistics of $Z_1, \ldots, Z_N$. 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.

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 Z.

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.

3.1 The QRT learning procedure

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

$$\sum_{i=1}^{N} -\log f_\theta'(Y_i \mid X_i) \quad (12)$$

$$= \sum_{i=1}^{N} -\log f_\theta(Y_i \mid X_i) - \log f_Z(F_\theta(Y_i \mid X_i)) \quad (13)$$

$$= \sum_{i=1}^{N} -\log f_\theta(Y_i \mid X_i) + \hat{H}(Z). \quad (14)$$

The first term in (14) is the NLL of the base model F_θ and $\hat{H}(Z)$ can be interpreted as the entropy of Z.

Interestingly, the second term $\hat{H}(Z)$ corresponds to the opposite of the regularization term of QREG (11). This observation could seem counter-intuitive since it implies that, when training QRT, the PCE of F_θ will be maximized by the second term $\hat{H}(Z)$ in the decomposition. However, QRT is valid since it produces F_θ' by minimizing the NLL of F_θ', which is a strictly proper scoring rule.

To compute the second term in (13), we estimate f_Z using the derivative of the calibration map $\Phi_\theta^{\mathrm{KDE}}$, which has a closed-form expression given by

$$\phi_\theta^{\mathrm{KDE}}(\alpha) = {}^{\partial \Phi_\theta^{\mathrm{KDE}}(\alpha)}/_{\partial \alpha} \quad (15)$$

$$= \frac{1}{N} \sum_{i=1}^{N} f_{\mathrm{Log}}(\alpha; Z_i, b^2 N^{-2/5}), \quad (16)$$

where f_{Log} is the PDF of a logistic distribution, as described in Section 2.

During training, $\phi_\theta^{\mathrm{KDE}}$ is computed on the current batch and F_θ' is thus, by design, calibrated on the current batch. However, it does not satisfy the calibration guarantee (7) since the current batch has been seen 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 3.3.

Finally, to account for the finite domain $[0, 1]$ of the PIT Z, we perform a slight adjustment to the calibration map $\phi_\theta^{\mathrm{KDE}}$. 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

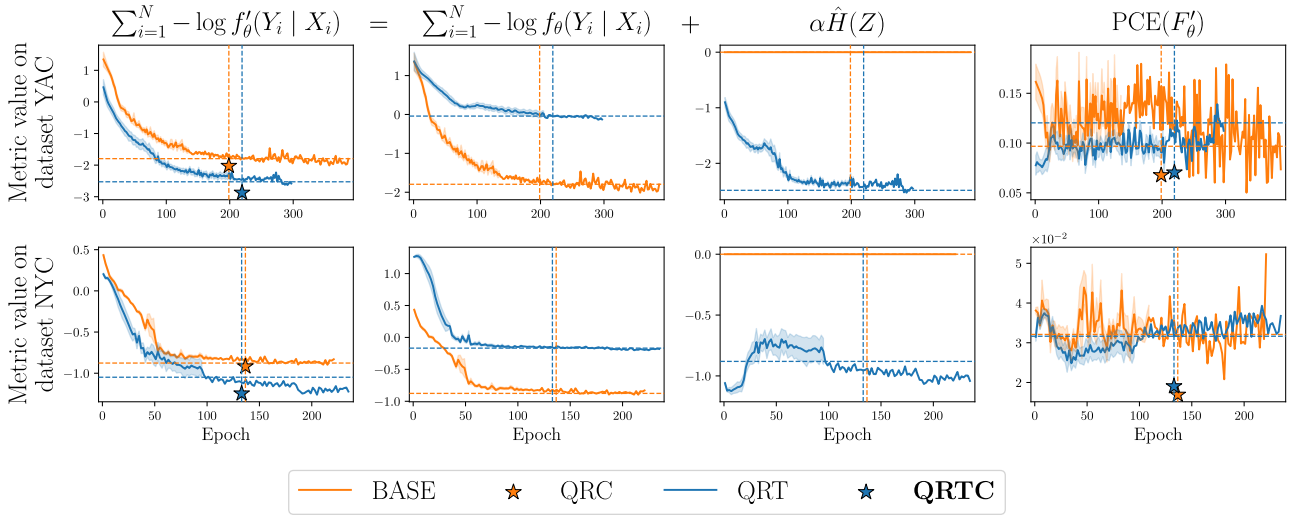


Figure 1: Comparison of QRT and BASE according to different metrics computed on the validation dataset. The three first columns show the decomposition of the NLL of QRT, where $\alpha = 1$ for QRT and $\alpha = 0$ for BASE. Each row represents one dataset and each column one metric. The training curves 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.

density slightly outside of $[0, 1]$ near 0 and 1, assuming that $\phi_\theta^{\mathrm{KDE}}(x) = 0$ for $x \notin [-1, 2]$. The resulting calibration map $\phi_\theta^{\mathrm{REFL}}$ is defined as:

$$\phi_\theta^{\mathrm{REFL}}(x) = \tag{17}$$
$$\begin{cases} \phi_\theta^{\mathrm{KDE}}(x) + \phi_\theta^{\mathrm{KDE}}(-x) + \phi_\theta^{\mathrm{KDE}}(2-x) & \text{if } x \in [0,1] \\ 0 & \text{if } x \notin [0,1]. \end{cases}$$

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 $\Phi_\theta^{\mathrm{REFL}}$, are given in Appendix D.

3.2 Illustrative example

Figure 1 illustrates the decomposition (14), where the NLL of F'_θ (first column) is equal to the sum of the NLL of the base model F_θ (second column) and the entropy of the PIT (third column). To allow a comparison between `QRT` and `BASE`, we alter the decomposition by introducing a coefficient α to the second term. When $\alpha = 1$, we obtain the exact decomposition of `QRT`. When $\alpha = 0$, 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 Appendix G. 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 models `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 F_θ is higher, which means that `QRT` relies on the calibration map 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 5 are available in Appendix G, where we obtain similar observations on most datasets despite their heterogeneity.

3.3 A Unified Algorithm

Algorithm 1 unifies QRT, QREG and BASE, with or without QR, where the methods only differ by the hyperparameters α and C, as indicated in Table 1. The

hyperparameter α, introduced in Section 3.2, is a coefficient for the second term of the decomposition (14). A value of $\alpha = 1$ corresponds to QRT and $\alpha = 0$ corresponds to NLL minimization without QRT. Tuning the hyperparameter α in order to minimize $\mathrm{PCE}(F_\theta)$ corresponds to QREG with regularization strength $\lambda = -\alpha$. The hyperparameter C controls whether the final model is recalibrated on a separate calibration dataset using QR.

Algorithm 1: QRT framework

Input : Predictive CDF F_θ, regularization strength $\alpha \in \mathbb{R}$, boolean C, training dataset $\mathcal{D}$, calibration dataset $\mathcal{D}'$

foreach *minibatch* $\{ (X_i, Y_i) \}_{i=1}^B \subseteq \mathcal{D}$, *until early stopping* **do**

Compute $Z_i \leftarrow F_\theta(Y_i \mid X_i)$, for $i = 1, \dots, B$

Define $\phi_\theta^{\mathrm{REFL}}$ from $Z_1, \dots, Z_B$ using (17)

$\mathcal{L}(\theta) = -\frac{1}{B} \sum_{i=1}^B \underbrace{\log f_\theta(Y_i \mid X_i) + \alpha \log \phi_\theta^{\mathrm{REFL}}(Z_i)}_{\log f'_\theta(Y_i \mid X_i)}$

Update parameters θ using $\nabla_\theta \mathcal{L}(\theta)$

if C *is True* **then**

Compute $Z'_i \leftarrow F_\theta(Y'_i \mid X'_i)$, for $i = 1, \dots, |\mathcal{D}'|$

Define $\Phi_\theta^{\mathrm{REFL}}$ from $Z'_1, \dots, Z'_{|\mathcal{D}'|}$

return the predictive CDF $\Phi_\theta^{\mathrm{REFL}} \circ F_\theta$

else

return the predictive CDF F_θ

In Algorithm 1, the calibration map $\phi_\theta^{\mathrm{KDE}}$ 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 (14). In Appendix B, 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 1 provides similar NLL than the approach in Appendix B, while being more computationally efficient. In Appendix F, we confirm that $\alpha = 1$ provides the best NLL compared to other values of α.

Table 1: Summary of the compared methods, which differ only by the hyperparameters α and C in Algorithm 1. We recommend using `QRTC`.

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

3.4 Time complexity

The proposed method can introduce increased computational demand due to evaluating $\log \phi_\theta^{\mathrm{KDE}}(Z_i)$, 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 $\phi_\theta^{\mathrm{REFL}}$ (see Appendix D). 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 twofold compared to a method without QR, as detailed in Appendix K.

4 RELATED WORK

Post-hoc calibration methods Many post-hoc calibration methods have been proposed for classification problems (Kumar et al., 2019; Gupta et al., 2020). The most popular one is called temperature scaling (Guo et al., 2017) and has the useful property of preserving accuracy. Conformal prediction, pioneered by Vovk et al. (2005), is an attractive approach due to the finite-sample coverage guarantee that it provides. In regression, multiple approaches based on conformal prediction have been proposed, including Conformal Quantile Regression (Romano et al., 2019) and Distributional Conformal Prediction (Chernozhukov et al., 2021). Quantile Recalibration (Kuleshov et al., 2018) is another method which transforms predictive distributions using a recalibration map, and has been shown to be closely related to Distributional Conformal Prediction (Dheur and Ben Taieb, 2023). Finally, methods have been proposed to target a stronger notion of calibration, called distribution calibration (Song et al., 2019; Kuleshov and Deshpande, 2022).

Regularization methods Regularization-based calibration methods aim to improve calibration during training, e.g. using 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). Multiple regularization objectives have been proposed in classification (A. Kumar et al., 2018; Karandikar et al., 2021; Popordanoska et al., 2022; Yoon et al., 2023) and regression (Pearce et al., 2018; Utpala and Rai, 2020; Chung et al., 2021; Dheur and Ben Taieb, 2023). While these methods allow to improve calibration, they may negatively impact other accuracy metrics. In fact, Karandikar et al. (2021) and Yoon et al. (2023) reported selecting hyperparameters that minimize the expected calibration error while decreasing the accu-

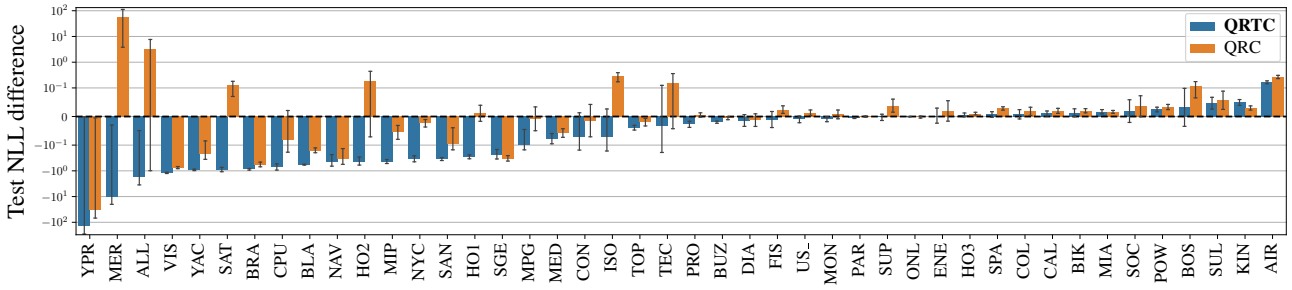


Figure 2: Difference in test NLL between two post-hoc methods (`QRTC` and `QRC`) and `BASE`, where negative values indicate an improvement compared to `BASE`, averaged over 5 runs with error bars corresponding to one standard error. We observe that `QRTC` achieves a lower NLL than `BASE` and `QRC` on most datasets. Note that, for `BASE`, F_θ is trained with a larger dataset that includes the calibration data of `QRTC` and `QRC`. The experimental setup is described in Section 5

racy by about 1%. Similarly, Dheur and Ben Taieb (2023) selected the regularization factor λ that minimizes the PCE with a maximum CRPS increase of 10%. In contrast to these methods, Recalibration Training did not impose such a trade-off in our large-scale experiment and resulted in both improved NLL and PCE.

Towards unifying model training and post-hoc calibration Despite the potential benefits of combining post-hoc and regularization strategies to improve calibration, empirical evidence from both classification (Wang et al., 2021) and regression (Dheur and Ben Taieb, 2023) contexts has indicated that frequently utilized regularization methods result in neural networks that are less calibratable. The method we propose is consistent with the recommendation made by Wang et al. (2021) to regard model training and post-hoc calibration as a unified framework. Finally, recent works in classification (Stutz et al., 2022; Einbinder et al., 2022) proposed integrating conformal prediction into neural network training. The outcome is precise coverage with smaller prediction sets.

5 A LARGE-SCALE EXPERIMENTAL STUDY

We compare the performance of `QRTC` (Section 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) [2] 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.

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

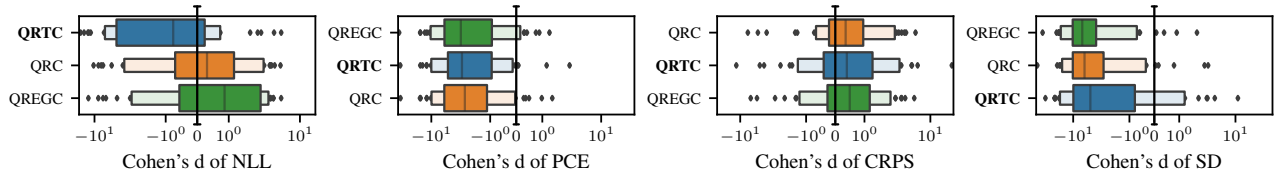
5.1 Benchmark datasets

In our study, we analyze a total of 57 data sets, including 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.

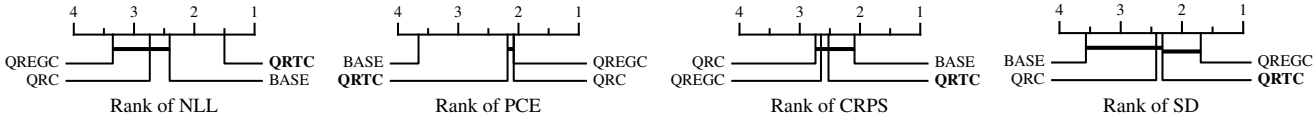
To avoid a potential bias in our analysis, we exclude certain datasets that could not be suited for regression. We identify these datasets using the proportion of targets Y that are among the 10 most frequent values in the dataset, and we call this proportion the level of discreteness. Table 3 in the 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. `QRTC` was able to perform better on these datasets, as discussed in Appendix I where full results are available.

5.2 Experimental setup

Base neural network model The base model F_θ is a mixture of $K = 3$ Gaussians, where the means $\mu_k(X)$, standard deviations $\sigma_k(X)$, and weights $w_k(X)$, for each component $k = 1, ..., K$ are obtained as outputs of a hypernetwork, which is a 3-layer MLP with 128 hidden units per layer. We also consider other base models in Appendix A.



(a) Letter-value plots showing Cohen's d for different metrics with respect to `Base`.



(b) Critical difference diagrams for different metrics.

Figure 3: Comparison of `QRTC`, `QRC`, `QREGC` and `BASE`, as detailed in Section 5.

Compared methods We compare all methods in Table 1 where QR is applied, namely `QRC`, `QRTC` and `QREGC`. We also compare `BASE` as a baseline method, where, to ensure a fair comparison, the calibration dataset is used as additional training data for the base model F_θ since there is no need for a calibration dataset. For `QRTC` and `QREGC`, the bandwidth b of Φ_θ^{KDE} is selected by minimizing the validation NLL in the set $\{0.01, 0.05, 0.1, 0.2\}$. Appendix C 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 $\lambda = -\alpha$ 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 (1). Finally, we measure sharpness using the mean standard deviation of the predictions, denoted by SD.

Comparison of multiple models over many datasets Given the different scales of NLL, CRPS, and SD across datasets, we report Cohen's d, a standardized effect size metric to compare the mean performance of a method against a baseline. Cohen's d values of -0.8 and -2 are regarded as large and huge effect sizes, respectively. Owing to the diverse nature of the datasets used in our study, the performance metrics of our models can exhibit substantial variations. To effectively illustrate the results, we employ letter-value plots to depict the distribution of Cohen's d. These plots highlight the quantiles at levels $1/8$, $1/4$, $1/2$, $3/4$ and $7/8$, as well as any outliers. A median value below zero indicates an improvement in the metric across more than half of the datasets by the model. Letter-value plots are ordered based on the median value to facilitate an easy identification of the top-performing methods.

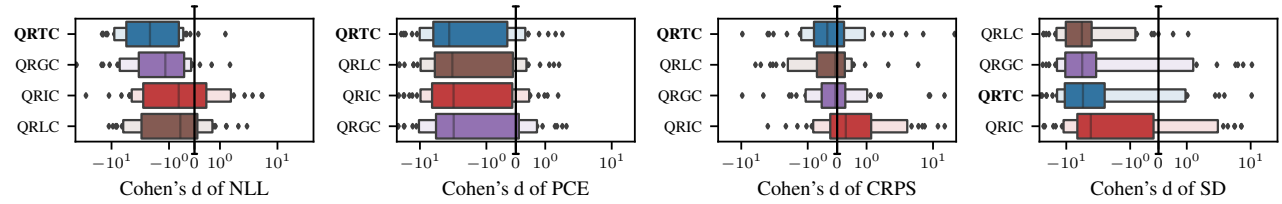
In order to determine if there's a significant difference in model performance, we first apply the Friedman test (Friedman, 1940). Subsequently, we carry out a pairwise post-hoc analysis, as advocated by Benavoli et al. (2016), using a Wilcoxon signed-rank test (Wilcoxon, 1945) complemented by Holm's alpha correction (Holm, 1979). These findings are represented by a critical difference diagram (Demšar, 2006). The lower the rank (further to the right), the superior the model's performance. A thick horizontal line illustrates a set of models with statistically indistinguishable performance, at a significance level of 0.05.

5.3 Results

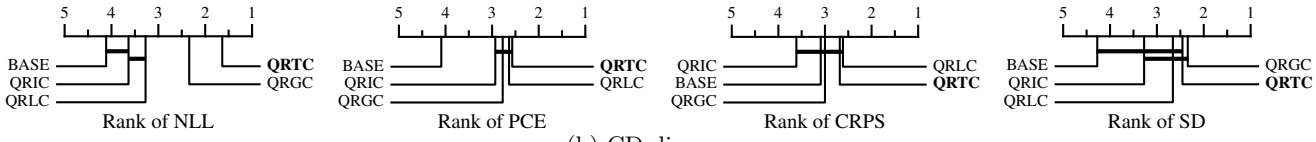
Figure 2 illustrates the comparison in NLL of `QRTC` and `QRC` across various datasets, relative to `BASE`. We observe that `QRTC` consistently achieves a lower NLL on the majority of the datasets. This suggests that allowing the model to adapt to the calibration map during training improves the final predictive accuracy, without the need for extra parameters.

Figure 3 shows the letter-values plots for Cohen's d of different metrics (top panel) as well as the associated critical difference diagram (bottom panel), for all methods and datasets. The reference model is `BASE`. Figure 3(b) 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 2. In terms of PCE, since all considered methods except `BASE` are combined with QR, they benefit from the finite sample guarantee (7) 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



(a) Letter-value plots showing Cohen's d for different metrics with respect to `Base`.



(b) CD diagrams

Figure 4: Comparison of `QRTC`, `QRGC`, `QRIC`, `QRLC` and `BASE` as detailed in Section 6.1.

at the observed/realized test data points while the characteristics measured by CRPS, a distance-sensitive scoring rule (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`.

5.4 The importance of the base model

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; 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 model, 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 Appendix A.

6 AN ABLATION STUDY AND ANALYSIS OF QUANTILE RECALIBRATION TRAINING

6.1 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 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 Z_i' in Algorithm 1. 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 recalibration map*, corresponds to `QRTC` except that the PITs Z_i in Algorithm 1 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 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`.

7 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 demonstrates enhanced performance in both predictive accuracy (NLL) and calibration compared to the baseline. Compared to Quantile Recalibration, QRT achieves a similar calibration improvement with an additional enhancement in NLL. 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 (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.

References

[1] Moloud Abdar et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". *An international journal on information fusion* 76 (Dec. 2021), pp. 243–297.

[2] Alexander Amini et al. "Deep Evidential Regression" (July 2019), pp. 14927–14937. arXiv: 1910.02600 [cs.LG].

[3] Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. "The need for uncertainty quantification in machine-assisted medical decision making". *Nature Machine Intelligence* 1.1 (July 2019), pp. 20–23.

[4] Alessio Benavoli, Giorgio Corani, and Francesca Mangili. "Should We Really Use Post-Hoc Tests Based on Mean-Ranks?" *Journal of machine learning research: JMLR* 17.5 (2016), pp. 1–10.

[5] Jaroslaw Blasiok and Preetum Nakkiran. "Smooth ECE: Principled Reliability Diagrams via Kernel Smoothing" (21 9 2023). arXiv: 2309.12236 [cs.LG].

[6] Victor Chernozhukov, Kaspar Wüthrich, and Yinchu Zhu. "Distributional conformal prediction". *Proceedings of the National Academy of Sciences of the United States of America* 118.48 (Nov. 2021).

[7] Youngseog Chung et al. "Beyond Pinball Loss: Quantile Methods for Calibrated Uncertainty Quantification". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 10971–10984.

[8] Janez Demšar. "Statistical Comparisons of Classifiers over Multiple Data Sets". *Journal of machine learning research: JMLR* 7 (Dec. 2006), pp. 1–30.

[9] Victor Dheur and Souhaib Ben Taieb. "A Large-Scale Study of Probabilistic Calibration in Neural Network Regression". In: *The 40th International Conference on Machine Learning*. PMLR, 2023.

[10] Hailiang Du. "Beyond Strictly Proper Scoring Rules: The Importance of Being Local". *Weather and Forecasting* 36.2 (Jan. 2021), pp. 457–468.

[11] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017.

[12] Bat-Sheva Einbinder et al. "Training Uncertainty-Aware Classifiers with Conformalized Deep Learning" (May 2022). arXiv: 2205.05878 [stat.ML].

[13] Milton Friedman. "A Comparison of Alternative Tests of Significance for the Problem of m Rankings". *The Annals of Mathematical Statistics* 11.1 (Mar. 1940), pp. 86–92.

[14] Jakob Gawlikowski et al. "A Survey of Uncertainty in Deep Neural Networks" (July 2021). arXiv: 2107.03342 [cs.LG].

[15] Pieter Gijsbers et al. "An Open Source AutoML Benchmark" (July 2019). arXiv: 1907.00909 [cs.LG].

[16] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. "Probabilistic forecasts, calibration and sharpness". *Journal of the Royal Statistical Society. Series B, Statistical methodology* 69.2 (Apr. 2007), pp. 243–268.

[17] Yury Gorishniy et al. "Revisiting deep learning models for tabular data". *Advances in neural information processing systems* 34 (2021), pp. 18932–18943.

[18] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. "Why do tree-based models still outperform deep learning on tabular data?" (July 2022). arXiv: 2207.08815 [cs.LG].

[19] Aditya Grover et al. "Stochastic Optimization of Sorting Networks via Continuous Relaxations". In: *International Conference on Learning Representations*. 2019.

[20] Chuan Guo et al. "On Calibration of Modern Neural Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1321–1330.

[21] Kartik Gupta et al. "Calibration of Neural Networks using Splines" (23 6 2020). arXiv: 2006.12800 [cs.LG].

[22] Sture Holm. "A Simple Sequentially Rejective Multiple Test Procedure". *Scandinavian journal of statistics, theory and applications* 6.2 (1979), pp. 65–70.

[23] Rafael Izbicki, Gilson Shimizu, and Rafael Stern. "Flexible distribution-free conditional predictive bands using density estimators". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Vol. 108. Proceedings of Machine Learning Research. PMLR, 2020, pp. 3068–3077.

[24] Archit Karandikar et al. "Soft calibration objectives for neural networks". *Advances in neural information processing systems* 34 (2021), pp. 29768–29779.

[25] Jukka Kohonen and Jukka Suomela. "Lessons learned in the challenge: Making predictions and scoring them". In: *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, 2006, pp. 95–116.

[26] Volodymyr Kuleshov and Shachi Deshpande. "Calibrated and Sharp Uncertainties in Deep Learning via Density Estimation". In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 11683–11693.

[27] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. "Accurate Uncertainties for Deep Learning Using Calibrated Regression". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 2796–2804.

[28] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. "Trainable Calibration Measures for Neural Networks from Kernel Mean Embeddings". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 2805–2814.

[29] Kumar, Liang, and Ma. "Verified uncertainty calibration". *Advances in neural information processing systems* (2019).

[30] Lakshminarayanan, Pritzel, et al. "Simple and scalable predictive uncertainty estimation using deep ensembles". *Advances in neural information processing systems* (2017).

[31] Rhiannon Michelmore, Marta Kwiatkowska, and Yarin Gal. "Evaluating Uncertainty Quantification in End-to-End Autonomous Driving Control" (16 11 2018). arXiv: `1811.06817 [cs.LG]`.

[32] Thomas Müller et al. "Neural Importance Sampling". *ACM transactions on graphics* 38.5 (Oct. 2019), pp. 1–19.

[33] Tim Pearce et al. "High-Quality Prediction Intervals for Deep Learning: A Distribution-Free, Ensembled Approach". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 4075–4084.

[34] Gabriel Pereyra et al. "Regularizing Neural Networks by Penalizing Confident Output Distributions" (Jan. 2017). arXiv: `1701.06548 [cs.NE]`.

[35] Teodora Popordanoska, Raphael Sayer, and Matthew Blaschko. "A consistent and differentiable lp canonical calibration error estimator". *Advances in neural information processing systems* (2022).

[36] Yaniv Romano, Evan Patterson, and Emmanuel Candes. "Conformalized quantile regression". *Advances in neural information processing systems* (2019).

[37] David W Scott. "On optimal and data-based histograms". *Biometrika* 66.3 (Dec. 1979), pp. 605–610.

[38] Hao Song et al. "Distribution calibration for regression". In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 5897–5906.

[39] David Stutz et al. "Learning Optimal Conformal Classifiers". May 2022.

[40] Saiteja Utpala and Piyush Rai. "Quantile Regularization: Towards Implicit Calibration of Regression Models" (Feb. 2020). arXiv: `2002.12860 [cs.LG]`.

[41] Oldrich Vasicek. "A Test for Normality Based on Sample Entropy". *Journal of the Royal Statistical Society. Series B, Statistical methodology* 38.1 (1976), pp. 54–59.

[42] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer International Publishing, 2005.

[43] Deng-Bao Wang, Lei Feng, and Min-Ling Zhang. "Rethinking Calibration of Deep Neural Networks: Do Not Be Afraid of Overconfidence". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 11809–11820.

[44] Frank Wilcoxon. "Individual Comparisons by Ranking Methods". *Biometrics Bulletin* 1.6 (1945), pp. 80–83.

[45] Hee Suk Yoon et al. "ESD: Expected Squared Difference as a Tuning-Free Trainable Calibration Measure". In: *The Eleventh International Conference on Learning Representations*. 2023.

[46] Hongyi Zhang et al. "mixup: Beyond Empirical Risk Minimization". In: *International Conference on Learning Representations*. 2018.

[47] Shengjia Zhao, Tengyu Ma, and Stefano Ermon. "Individual Calibration with Randomized Forecasting". In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 11387–11397.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Yes.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes.
   (b) Complete proofs of all theoretical results. Yes.
   (c) Clear explanations of any assumptions. Yes.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Yes.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). Yes.

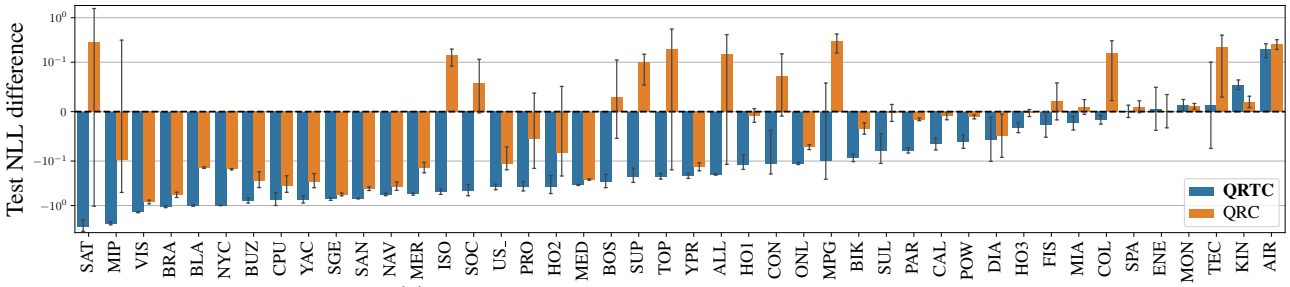
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. Yes.
   (b) The license information of the assets, if applicable. Yes.
   (c) New assets either in the supplemental material or as a URL, if applicable. Yes.
   (d) Information about consent from data providers/curators. Not Applicable.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable.
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable.

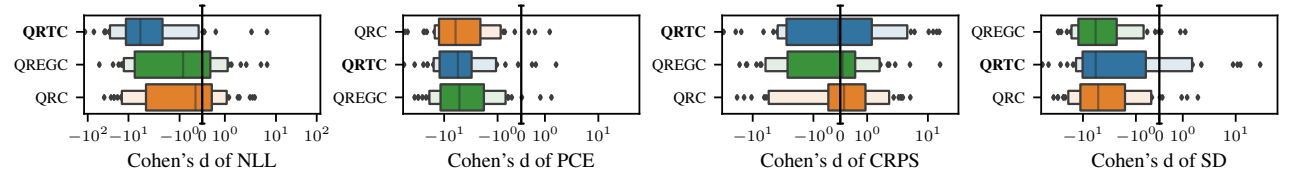
A RESULTS ON DIFFERENT BASE MODELS

We present detailed results on the significance of the base model in influencing the performance of QRT, as discussed in Section 5.4 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 5 to 7, we follow the exact same setup as in the main experiments except that the underlying neural network is modified.

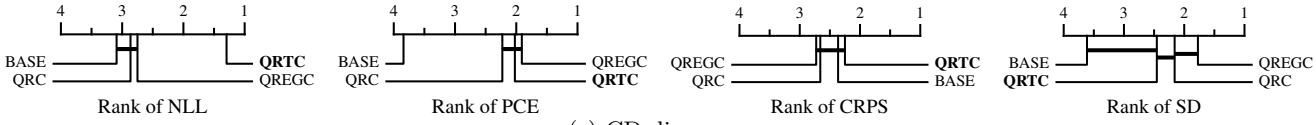
Figure 5 presents the results where the neural network is 3-layer MLP predicting a single Gaussian (i.e., one mean and one standard deviation). In this misspecified case, we observe on Figure 5(a) that, on many datasets, both QRTC and QRC provide an improvement in NLL compared to BASE, despite BASE having 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.



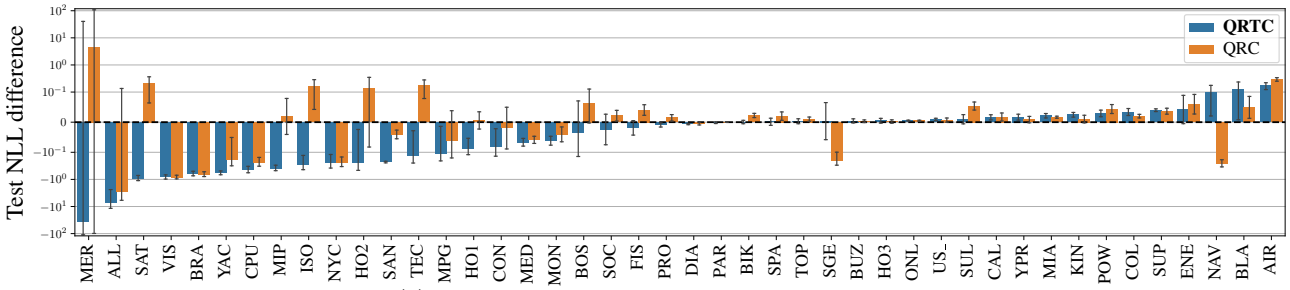
(b) Letter-value plots showing Cohen's d for different metrics with respect to BASE.



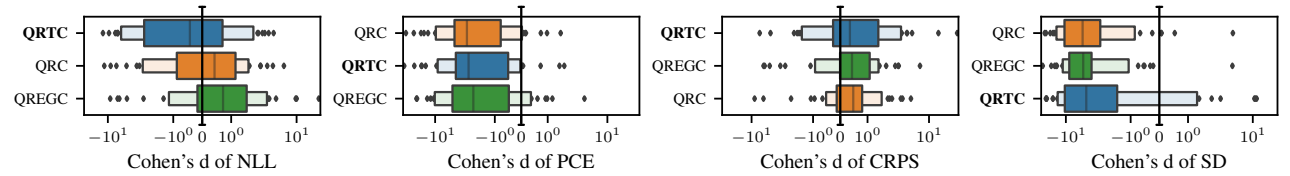
(c) CD diagrams

Figure 5: Same setup than the main experiments (Figure 3 in the main text), except that the underlying neural networks produces a single Gaussian instead of a mixture of 3 Gaussians.

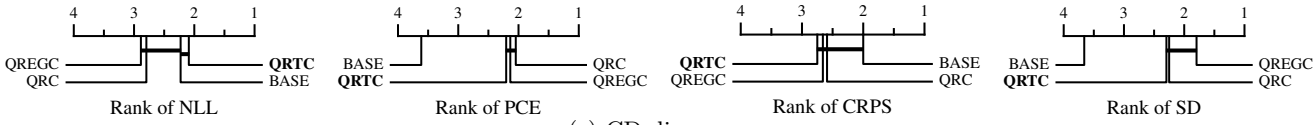
Figure 6 shows the same experiment except that the underlying neural network produces a mixture of 10 Gaussians (i.e., 10 means, 10 standard deviations, and 10 weights for each mixture component), offering high flexibility. In this case, QRTC provides an improvement in NLL in slightly more than half 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 with respect to `BASE`.



(c) CD diagrams

Figure 6: Same setup than the main experiments (Figure 3 in the main text), except that the underlying neural networks produces a mixture of 10 Gaussians instead of a mixture of 3 Gaussians.

Figure 7 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:

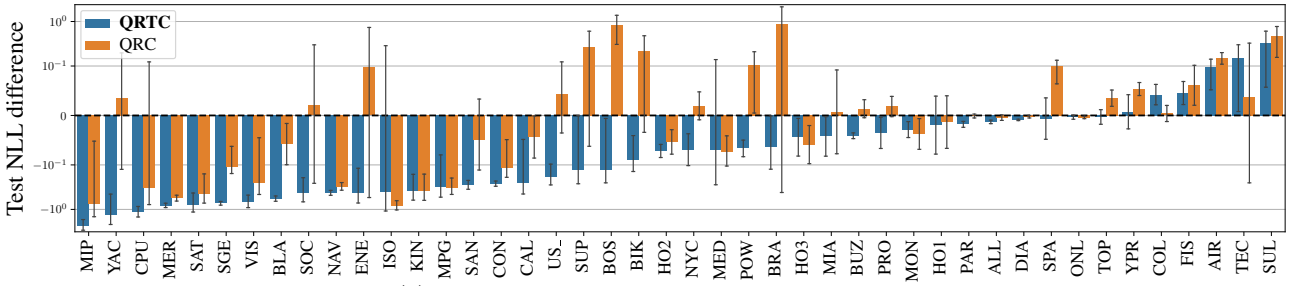
$$\begin{aligned}
\texttt{ResNet}(x) &= \texttt{Prediction}(\texttt{ResNetBlock}(\dots\texttt{ResNetBlock}(\texttt{Linear}(x)))) \\
\texttt{ResNetBlock}(x) &= x + \texttt{Dropout}(\texttt{Linear}(\texttt{Dropout}(\texttt{ReLU}(\texttt{Linear}(\texttt{BatchNorm}(x)))))) \\
\texttt{Prediction}(x) &= \texttt{Linear}(\texttt{ReLU}(\texttt{BatchNorm}(x)))
\end{aligned}$$

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:

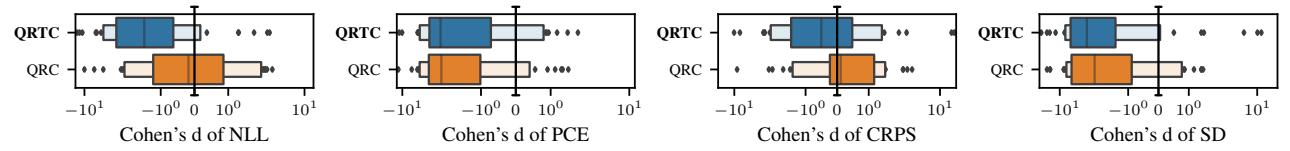
$$f_\theta(y \mid x) = \sum_{k=1}^{K} w_k(x)\mathcal{N}(y; \mu_k(x), \sigma_k^2(x)),$$

where $\mathcal{N}(y; \mu, \sigma^2)$ is the density of a normal distribution with mean μ and standard deviation σ evaluated at y.

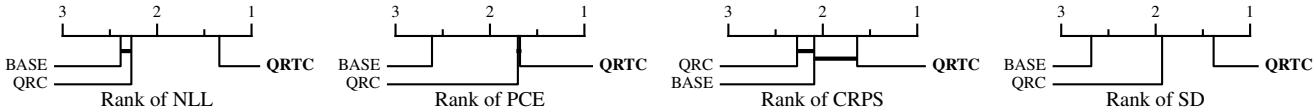
Figure 7(a) shows that `QRTC` remains advantageous even in deep models, with a notable improvement in NLL on most datasets compared to `QRC`. Similarly, observations from Figure 7 align with previous findings in PCE. Finally, `QRTC` is both significantly better than `QRC` in CRPS and significantly sharper.



(a) Difference of test NLL compared to `BASE`.



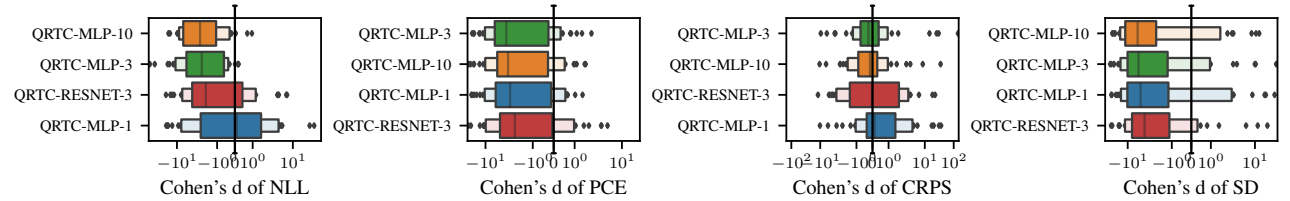
(b) Letter-value plots showing Cohen's d for different metrics with respect to `BASE`.



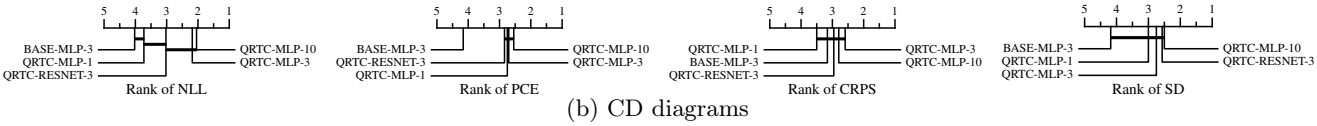
(c) CD diagrams

Figure 7: Same setup than the main experiments (Figure 3 in the main text), except that the underlying neural networks is a ResNet.

Finally, we provide a comparison of the performance of `QRTC` on all base models under consideration. Each model is denoted by `QRTC-<BM>-`K where `<BM>` is the base model and K is the mixture size. As illustrated in Figure 8, 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 with respect to `BASE` (using an MLP model and mixture predictions of size 3 as in the main text).



(b) CD diagrams

Figure 8: Comparison of `QRTC` with different base models.

B IMPACT OF THE SIZE OF THE CALIBRATION MAP

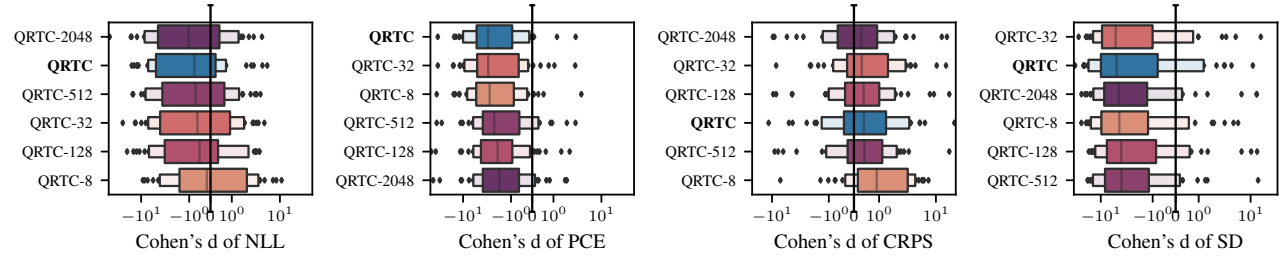
As discussed in Section 3.3, we investigate the impact of computing the calibration map from a dataset of size M sampled randomly from the training dataset instead of the current batch. Specifically, this would correspond to changing the training loop of Algorithm 1 as depicted by Algorithm 2.

Algorithm 2: QRT framework where $\phi_\theta^{\text{REFL}}$ is computed from a random sample of the training dataset.

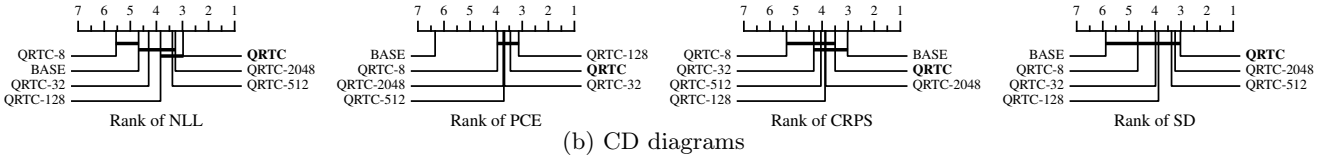
Input : Predictive CDF F_θ, training dataset $\mathcal{D}$, size of calibration map M
$M \leftarrow \min\{M, |D|\}$
foreach *minibatch* $\{(X_i, Y_i)\}_{i=1}^B \subseteq \mathcal{D}$, *until early stopping* **do**
  Sample $\{(X_i', Y_i')\}_{i=1}^M$ from $\mathcal{D}$ without replacement
  Compute $Z_i' \leftarrow F_\theta(Y_i' \mid X_i')$, for $i = 1, \dots, M$
  Define $\phi_\theta^{\text{REFL}}$ from $Z_1', \dots, Z_M'$ using (21)
  Compute $Z_i \leftarrow F_\theta(Y_i \mid X_i)$, for $i = 1, \dots, B$
  $\mathcal{L}(\theta) = -\frac{1}{B}\sum_{i=1}^B \underbrace{\log f_\theta(Y_i \mid X_i) + \log \phi_\theta^{\text{REFL}}(Z_i)}_{\log f_\theta'(Y_i|X_i)}$
  Update parameters θ using $\nabla_\theta \mathcal{L}(\theta)$

While the approach proposed in the main text requires B neural network evaluations per minibatch, Algorithm 2 requires $M + B$ neural network evaluations per minibatch, making it relatively slower.

In Figure 9, we investigate the performance of `QRTC` using Algorithm 2 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 9(b) 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, with respect to `BASE`.

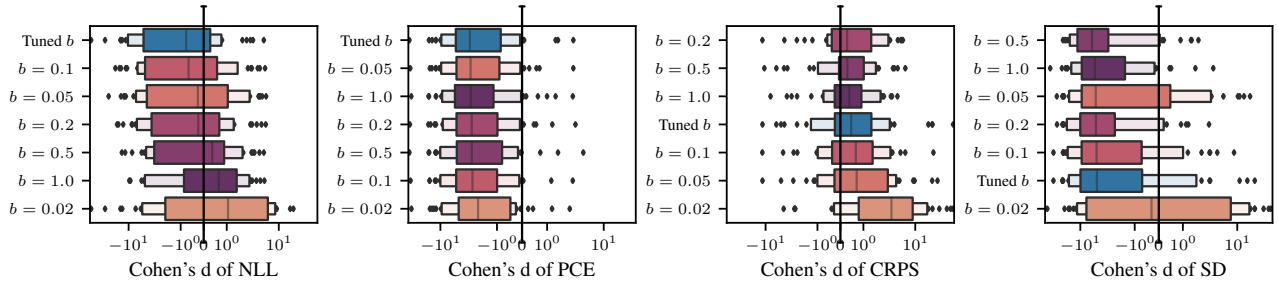


(b) CD diagrams

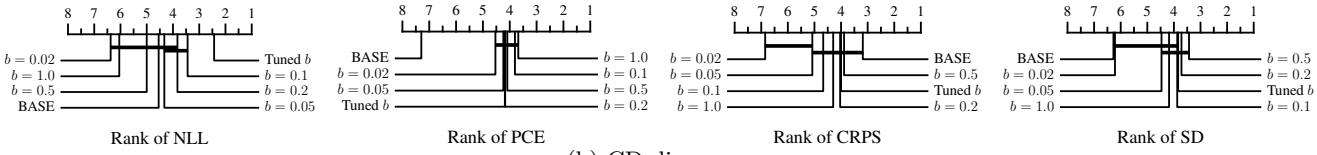
Figure 9: Comparison of `QRTC`, where the calibration map has been computed on calibration datasets of different sizes.

C IMPACT OF THE BANDWIDTH HYPERPARAMETER b

We evaluate the effect of tuning the bandwidth hyperparameter b in QRT. In Figure 10, 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, with respect to BASE.



(b) CD diagrams

Figure 10: Comparison of QRTC with different values of the hyperparameter b.

D KERNEL DENSITY ESTIMATION ON A FINITE DOMAIN

We provide more motivation and details regarding the calibration map $\Phi_\theta^{\text{REFL}}$ discussed in Section 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_θ^{KDE} and ϕ_θ^{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.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:

$$\Phi_\theta^{\text{TRUNC}}(x) = \begin{cases} {}^{F(x)-F(a)}/_{F(b)-F(a)} & \text{if } x \in [a, b] \\ 0 & \text{if } x < a \\ 1 & \text{if } x > b \end{cases} \tag{18}$$

and the resulting PDF is:

$$\phi_\theta^{\text{TRUNC}}(x) = \begin{cases} {}^{f(x)}/_{F(b)-F(a)} & \text{if } x \in [a, b] \\ 0 & \text{if } x \notin [a, b]. \end{cases} \tag{19}$$

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 11.

D.2 Proposed approach: Reflected distribution

To remedy this problem, we define a new PDF $\phi_\theta^{\text{REFL}}$ 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:

$$\Phi_\theta^{\mathrm{REFL}}(x) = \begin{cases} F(x) - F(2a - x) + 1 - F(2b - x) & \text{if } x \in [a, b] \\ 0 & \text{if } x < a \\ 1 & \text{if } x > b \end{cases} \tag{20}$$

and the corresponding PDF is defined by:

$$\phi_\theta^{\mathrm{REFL}}(x) = \begin{cases} f(x) + f(2a - x) + f(2b - x) & \text{if } x \in [a, b] \\ 0 & \text{if } x \notin [a, b]. \end{cases} \tag{21}$$

Figure 11 compares four methods to estimate the calibration map from PIT realizations $Z_1, \dots, Z_N$. The method $\Phi_\theta^{\mathrm{EMP}}$ was introduced in Section 2 and corresponds to the empirical CDF, which is not smooth. In contrast, the methods $\Phi_\theta^{\mathrm{KDE}}$, $\Phi_\theta^{\mathrm{TRUNC}}$ and $\Phi_\theta^{\mathrm{REFL}}$ offer smooth estimations. This figure shows that $\Phi_\theta^{\mathrm{REFL}}$ is closer to the empirical CDF than $\Phi_\theta^{\mathrm{TRUNC}}$ and the value of the corresponding PDF $\phi_\theta^{\mathrm{REFL}}$ is not overestimated, suggesting the superiority of this estimator.

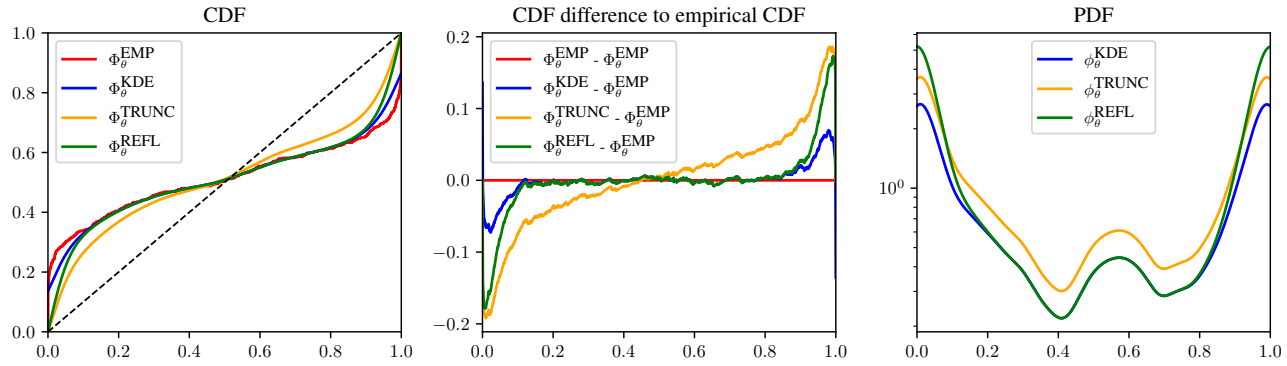
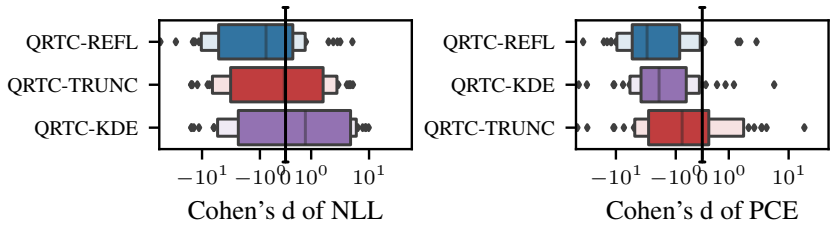
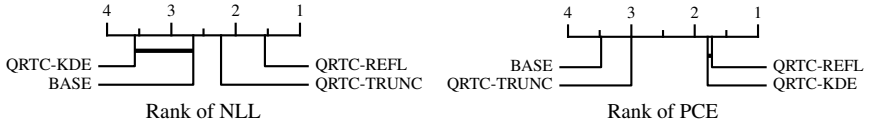


Figure 11: Comparison of different methods to estimate the calibration map. In this example, 512 PITs have been sampled from a beta distribution $Z \sim \mathrm{Beta}(0.2, 0.2)$ and the calibration map is estimated using $\Phi_\theta^{\mathrm{KDE}}$ with $b = 0.1$ (Equation (6) in the main text).

Figure 12 compares `QRTC` where the calibration map of the post-hoc model has been estimated using either $\Phi_\theta^{\mathrm{KDE}}$, $\Phi_\theta^{\mathrm{TRUNC}}$ and $\Phi_\theta^{\mathrm{REFL}}$. 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, with respect to `BASE`.

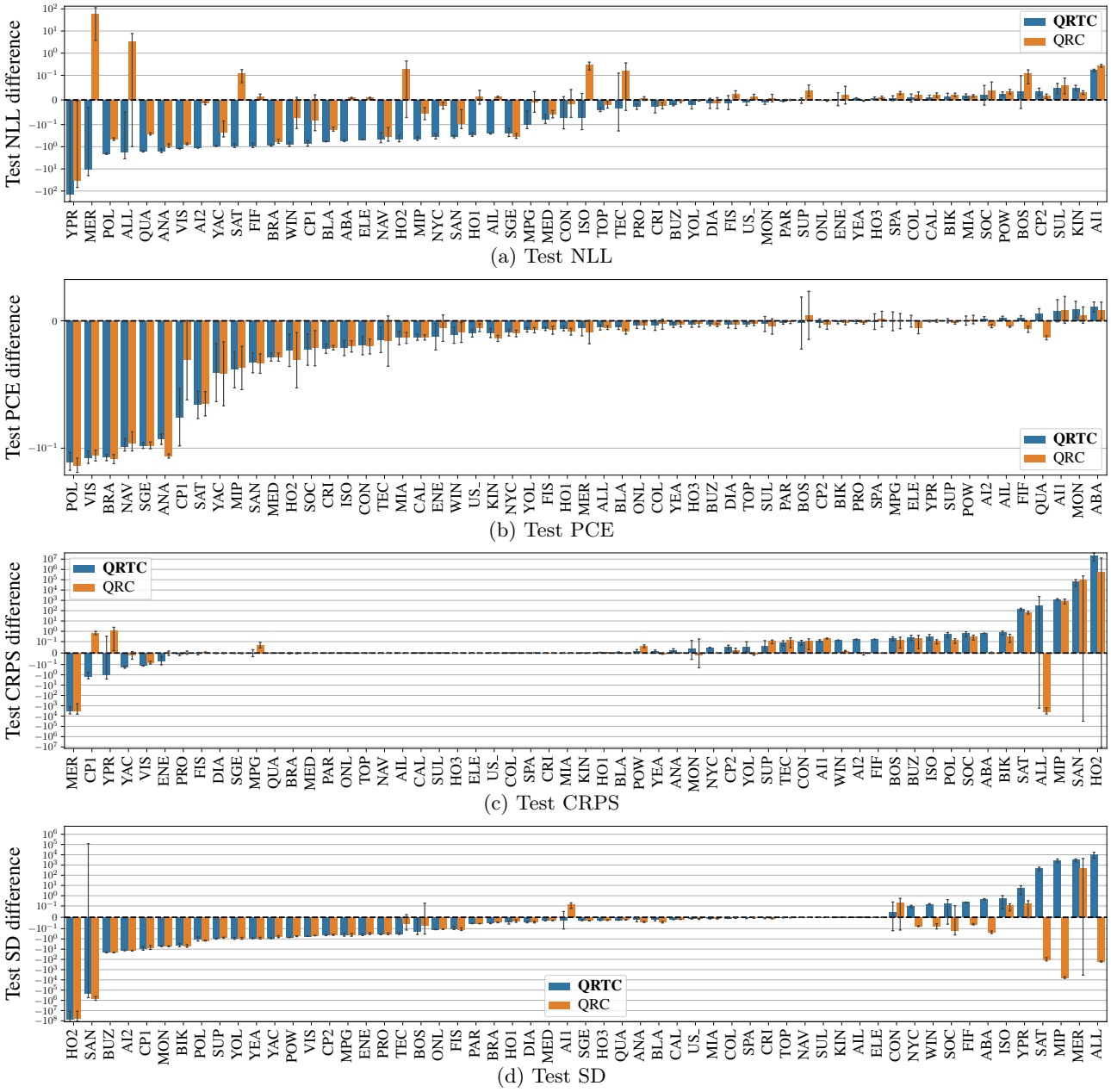


(b) CD diagrams

Figure 12: 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 $\left(\Phi_\theta^{\mathrm{KDE}}\right)^{-1}$ returns values outside the interval $[0, 1]$, we can not correctly sample from the model.

E DETAILED METRICS ON INDIVIDUAL DATASETS

For a more comprehensive view, Figure 13 presents a diagram analogous to Figure 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 5.1. 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

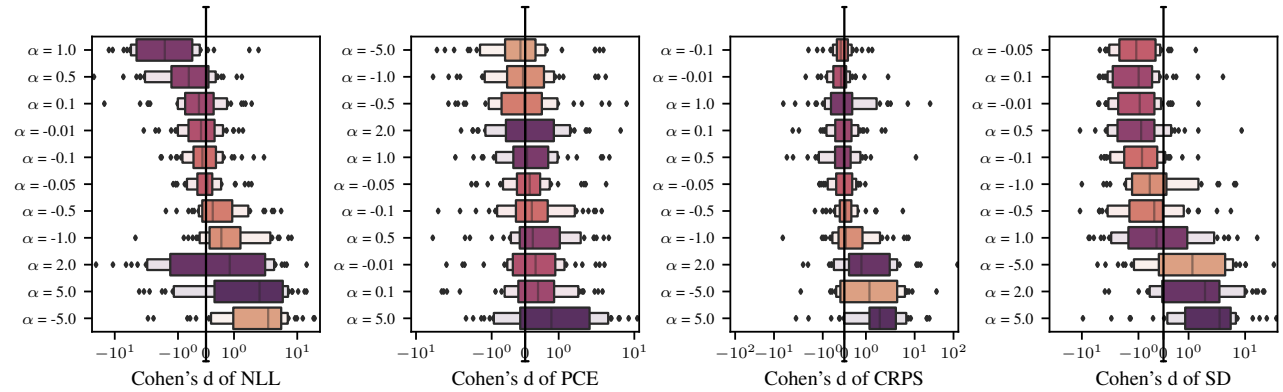
Figure 13: Comparison of QRTC and QRC with respect to BASE by showing the difference between the compared methods and BASE according to a given metric, in average over 5 runs.

F RESULTS WITH DIFFERENT VALUES OF α

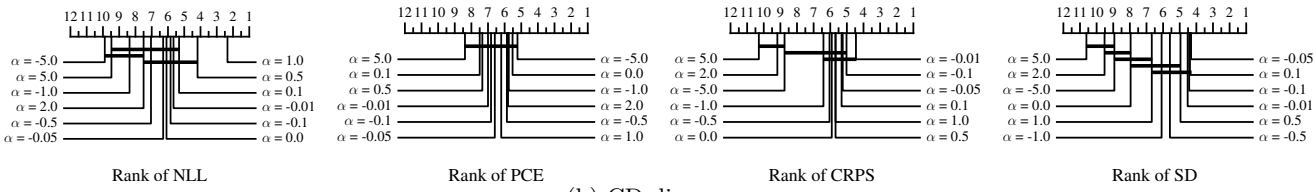
We provide detailed results regarding the hyperparameter α of Algorithm 1 in the main text. As discussed in Section 3.3, it is possible to design an algorithm unifying QRTC, QRC and QREGC, where the methods only differ by the hyperparameter α. We assume here that Quantile Recalibration is applied (C = True) and only consider variations of the hyperparameter α. As discussed previously, a value of $\alpha = 1$ corresponds to `QRTC`, $\alpha = 0$ corresponds to `QRC` and tuning α in order to minimize PCE(F_θ) corresponds to `QREGC` with regularization strength $\lambda = -\alpha$.

In Figure 14, we provide results with different values of α. Values of α 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, $\alpha = 1$, corresponding to the NLL decomposition of the recalibrated model, obtains the best NLL, and is significantly better than other values of α. In terms of CRPS, there is no significant differences for values of α between 0 and 1.



(a) Boxplots of Cohen's d of different metrics on all datasets, with respect to `QRC`.



(b) CD diagrams

Figure 14: Comparison of different values of the hyperparameter α.

G 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 3.

The setup mirrors the illustrative example presented in Section 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 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 15 and 17, 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 16 and Figure 18, 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.

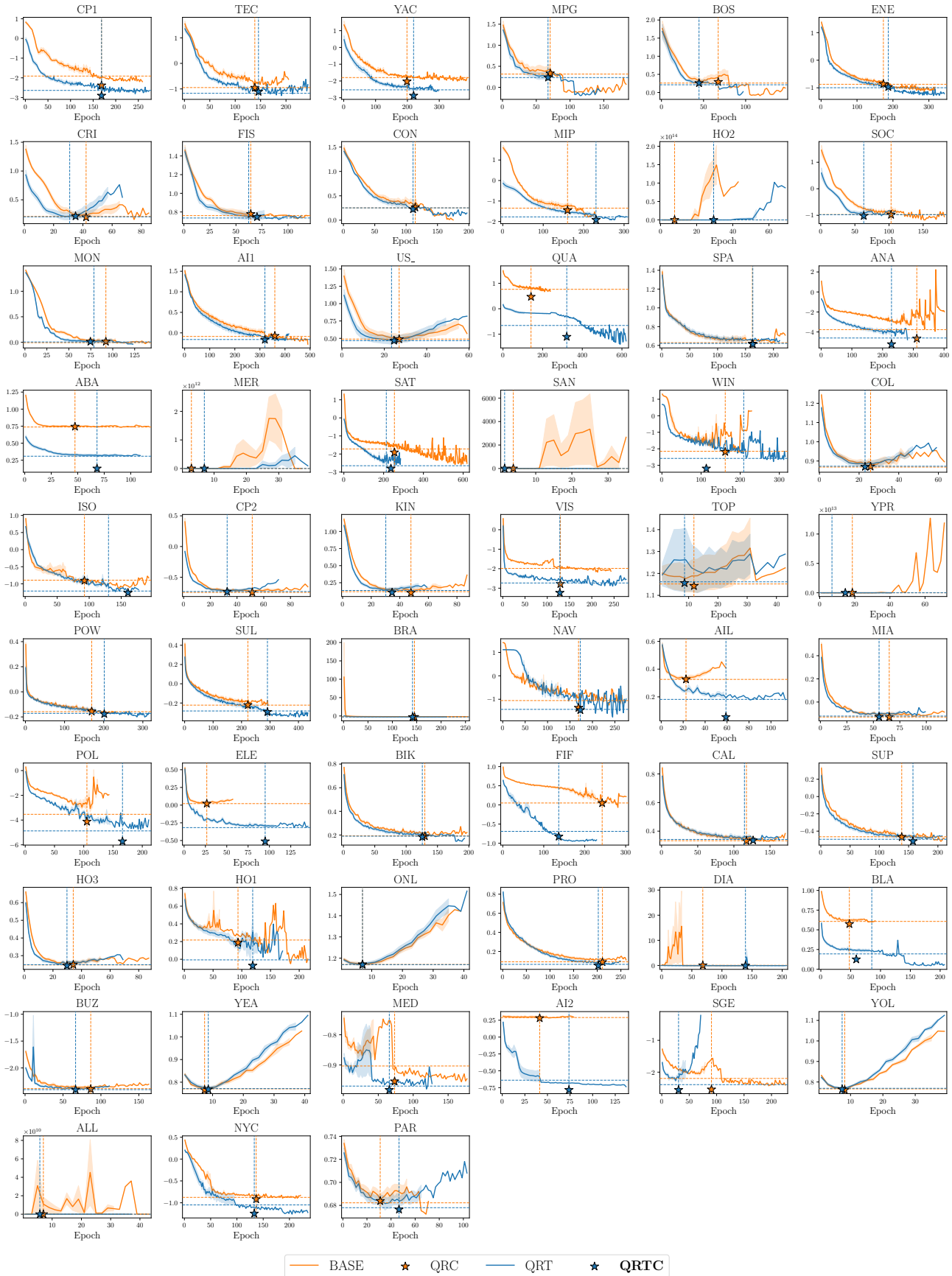


Figure 15: NLL on the validation dataset per epoch.

Figure 16: PCE on the validation dataset per epoch.

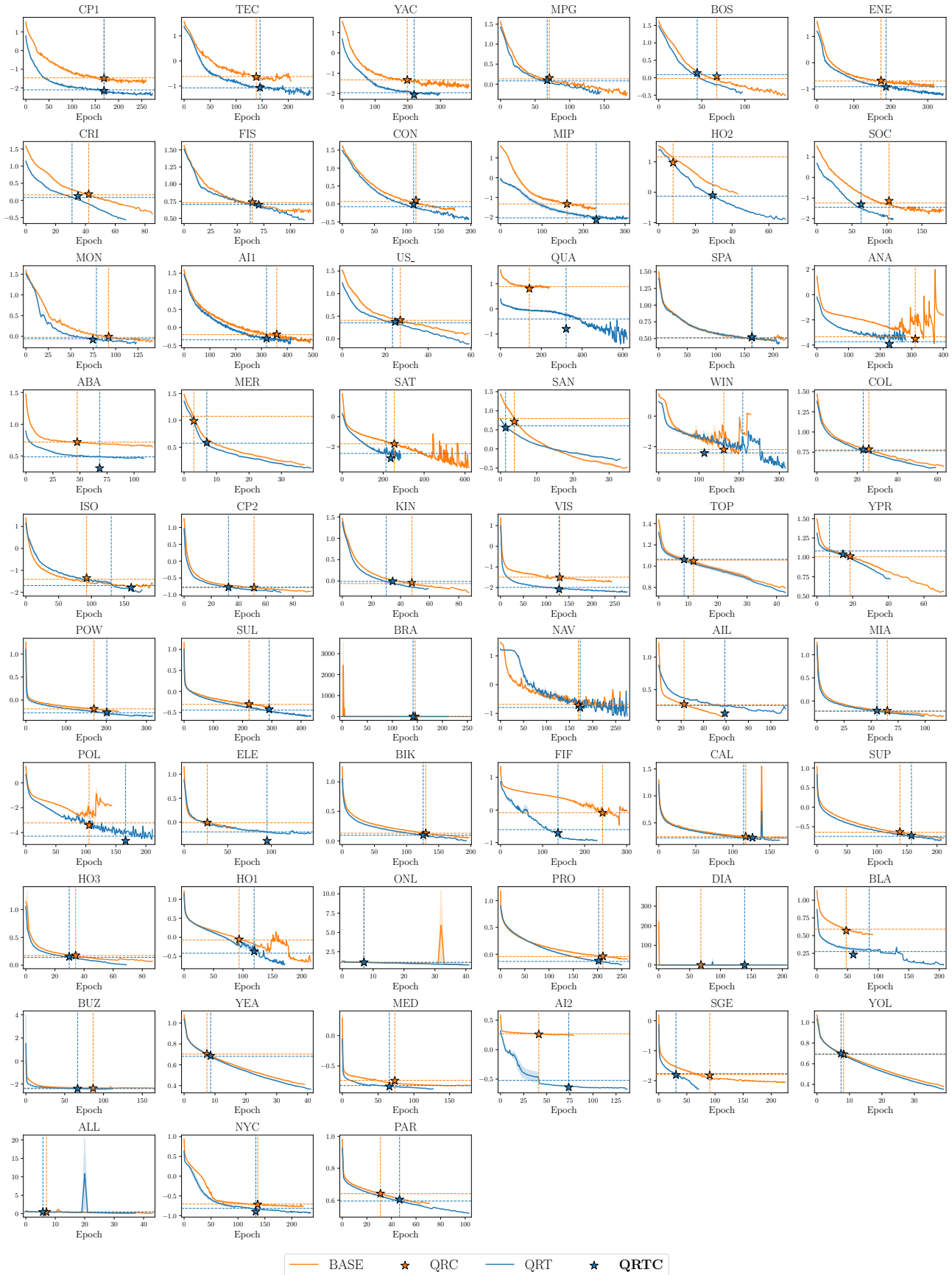


Figure 17: NLL on the training dataset per epoch.

Figure 18: PCE on the training dataset per epoch.

H 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 5.1 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 3 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 3 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 19 and 20, 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 to the discreteness level, we provide an isotonic regression estimate in red.

Figure 19 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. (2019) trained a model based on NLL on the `wine_quality` dataset for which the output variable only takes 7 distinct values.

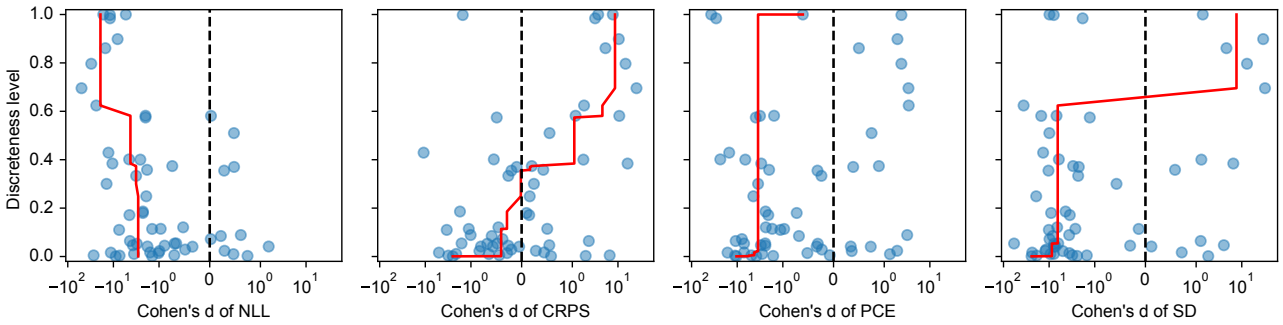


Figure 19: Cohen's d of different metrics compared to the discreteness level of a dataset for the `QRTC` model relative to the `BASE` model.

Figure 20 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 don't observe a notable trend in terms of PCE.

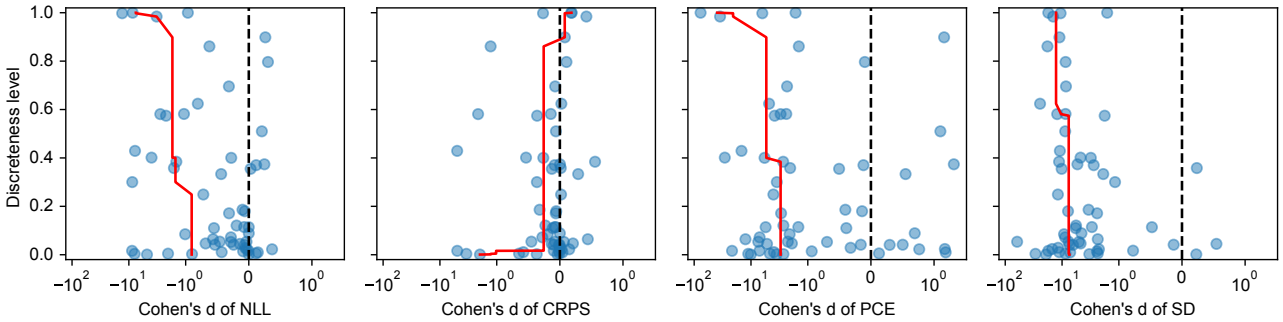
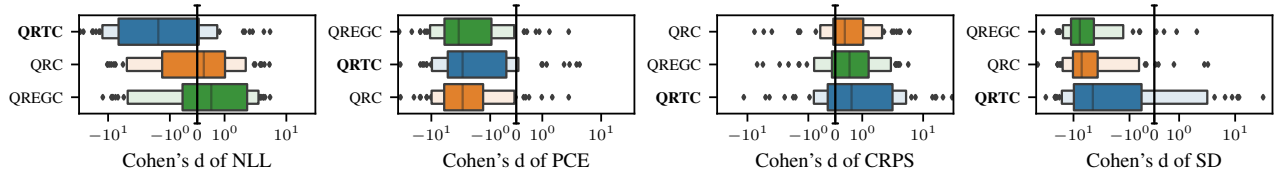


Figure 20: Cohen's d of different metrics compared to the discreteness level of a dataset for the QRC model relative to the BASE model.

I RESULTS ON ALL DATASETS

As discussed in Section 5.1, we provide the full results including the datasets with a high discreteness level. Despite the potential issues discussed in Appendix H, the conclusions drawn in Section 5 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 Appendix E.



(a) Letter-value plots showing Cohen's d for different metrics with respect to Base.

(b) Critical difference diagrams for different metrics.

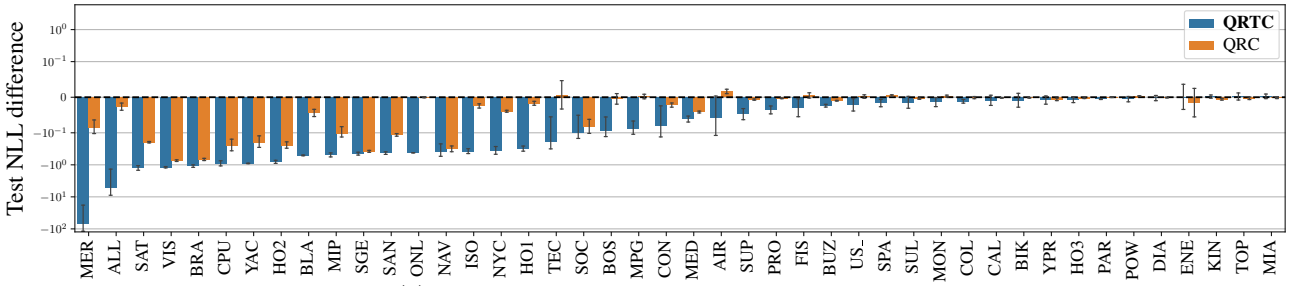
Figure 21: Same setup than the main experiments (Figure 3 in the main text), with all the datasets.

J RESULTS WHERE BASE DOES NOT HAVE ACCESS TO CALIBRATION DATA

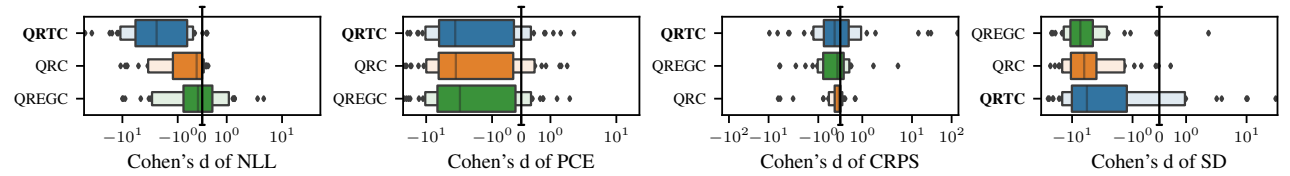
As discussed in Section 5.2, we aimed to provide a fair comparison between BASE and the post-hoc methods by training it on a larger dataset than the post-hoc methods QRTC, QRC and QREGC in all of our experiments. Since the post-hoc methods benefit from the calibration data during the post-hoc step, all 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 model than BASE and performs an additional post-hoc step.

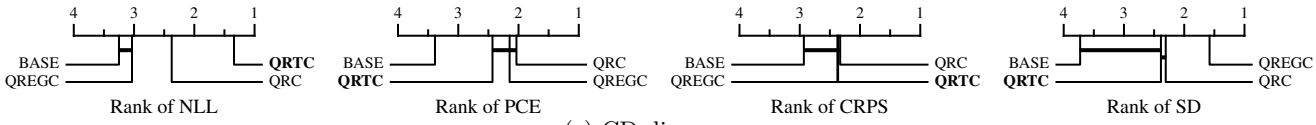
In Figure 22(a), QRC shows that the post-hoc step never degrades NLL and sometimes results in a notable NLL improvement, which suggests that a post-hoc step on additional calibration data is always beneficial. Figure 22(c) shows that QRC results in a significant NLL improvement compared to BASE, and 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 with respect to BASE.



(c) CD diagrams

Figure 22: Same setup than the main experiments (Figure 3 in the main text), except that BASE is not trained on the calibration data.

K COMPUTATIONAL TIME

In Table 2, 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 3.4, this disparity can be attributed to the extra computational overhead associated with the computation of the calibration map, i.e., $-\frac{1}{B}\sum_{i=1}^{B}\log\phi_{\theta}^{\mathrm{REFL}}(Z_i)$.

Table 2: 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

L TABULAR REGRESSION DATASETS

The datasets considered in our study are detailed in Table 3. The table provides information about the benchmark suite, full dataset name, abbreviations, number of training instances (truncated to 53,184 instances, similarly to Dheur and Ben Taieb (2023)), and the number of features. Additionally, the last two columns represent measures of the dataset's discreteness levels, as discussed in Appendix H. Proportions that are superior to 0.5 are highlighted in bold.

Table 3: Detailed properties of all datasets

Group	Dataset	Abbrev.	Nb of training instances	Nb of features	Proportion of top 10 most frequent values	Proportion of duplicated values
uci	CPU	CP1	135	7	0.33	**0.64**
	Yacht	YAC	200	6	0.11	0.21
	MPG	MPG	254	7	0.37	**0.78**
	Energy	ENE	499	9	0.05	0.22
	Crime	CRI	531	104	**0.57**	**0.95**
	Fish	FIS	590	6	0.04	0.12
	Concrete	CON	669	8	0.04	0.10
	Airfoil	AI1	976	5	0.02	0.04
	Kin8nm	KIN	5324	8	0.00	0.00
	Power	POW	6219	4	0.01	**0.65**
	Naval	NAV	7757	17	0.40	**1.00**
	Protein	PRO	31328	9	0.01	**0.80**
oml_297	wine_quality	WIN	4223	11	**1.00**	**1.00**
	isolet	ISO	5068	613	0.40	**1.00**
	cpu_act	CP2	5324	21	**0.51**	**1.00**
	sulfur	SUL	6552	6	0.01	0.09
	Brazilian_houses	BRA	6949	8	0.02	**0.58**
	Ailerons	AIL	8942	33	**0.86**	**1.00**
	MiamiHousing2016	MIA	9069	13	0.12	**0.91**
	pol	POL	9817	26	**0.98**	**1.00**
	elevators	ELE	10936	16	**0.80**	**1.00**
	Bike_Sharing_Demand	BIK	11482	6	0.11	**0.99**
	fifa	FIF	11961	5	**0.70**	**1.00**
	california	CAL	13765	8	0.08	**0.94**
	superconduct	SUP	14201	79	0.05	**0.93**
	house_sales	HO3	14446	15	0.07	**0.87**
	house_16H	HO1	15266	16	0.17	**0.96**
	diamonds	DIA	37075	6	0.02	**0.89**
	medical_charges	MED	53164	3	0.00	0.05
	year	YEA	53164	90	**0.58**	**1.00**
	nyc-taxi-green-dec-2016	NYC	53164	9	0.38	**1.00**
oml_299	analcatdata_supreme	ANA	2633	12	**1.00**	**1.00**
	Mercedes_Benz_Greener_Manufacturing	MER	2735	735	0.02	**0.53**
	visualizing_soil	VIS	5616	5	0.43	**1.00**
	yprop_4_1	YPR	5775	82	0.04	**0.94**
	OnlineNewsPopularity	ONL	27068	73	0.35	**0.99**
	black_friday	BLA	53164	23	0.00	**0.95**
	SGEMM_GPU_kernel_performance	SGE	53164	15	0.00	**0.70**
	particulate-matter-ukair-2017	PAR	53164	26	0.05	**0.92**
oml_269	tecator	TEC	156	124	0.19	0.48
	boston	BOS	328	22	0.18	**0.73**
	MIP-2016-regression	MIP	708	111	0.05	0.17
	socmob	SOC	751	39	0.36	**0.76**
	Moneyball	MON	800	18	0.09	**0.86**
	house_prices_nominal	HO2	711	234	0.11	**0.59**
	us_crime	US_	1295	101	0.37	**0.99**
	quake	QUA	1415	3	**1.00**	**1.00**
	space_ga	SPA	2019	6	0.01	0.00
	abalone	ABA	2715	10	**0.90**	**1.00**
	SAT11-HAND-runtime-regression	SAT	2886	118	0.06	**0.61**
	Santander_transaction_value	SAN	2898	3611	0.30	**0.73**
	colleges	COL	4351	34	0.03	0.44
	topo_2_1	TOP	5775	252	0.04	**0.94**
	Allstate_Claims_Severity	ALL	53164	477	0.00	0.10
	Yolanda	YOL	53164	100	**0.58**	**1.00**
	Buzzinsocialmedia_Twitter	BUZ	53164	70	0.25	**0.98**
	Airlines_DepDelay_10M	AI2	53164	5	**0.62**	**1.00**

M EXAMPLES OF PREDICTIONS

To offer a deeper understanding of the shape of the predictions, Figures 23 to 28 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 23 to 26 are from datasets where `QRTC` outperformed `QRC` in terms of NLL. Notably, within these, `QRTC` exhibits heightened confidence in its predictions for Figures 24 and 25. However, in other datasets, the NLL improvements are more subtle. The Figure 28 represents predictions on a dataset with a high level of discreteness which has not be 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 Appendix H. Overall, the shape of the predictions can vary greatly in function of the dataset.

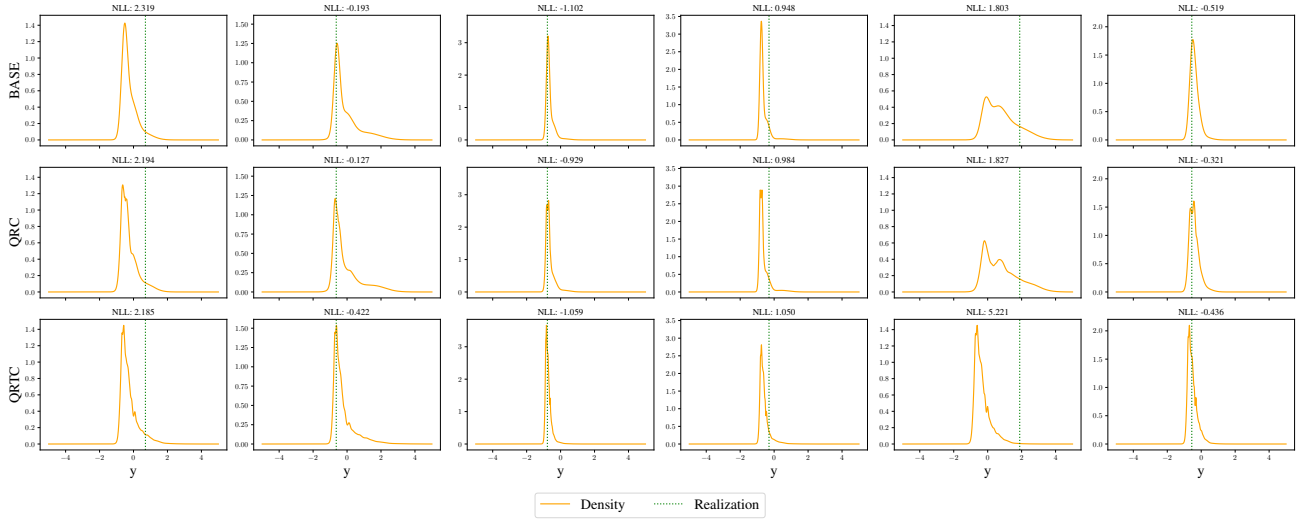


Figure 23: Examples of predictions of `BASE`, `QRC` and `QRTC` on dataset `Allstate_Claims_Severity` (ALL).

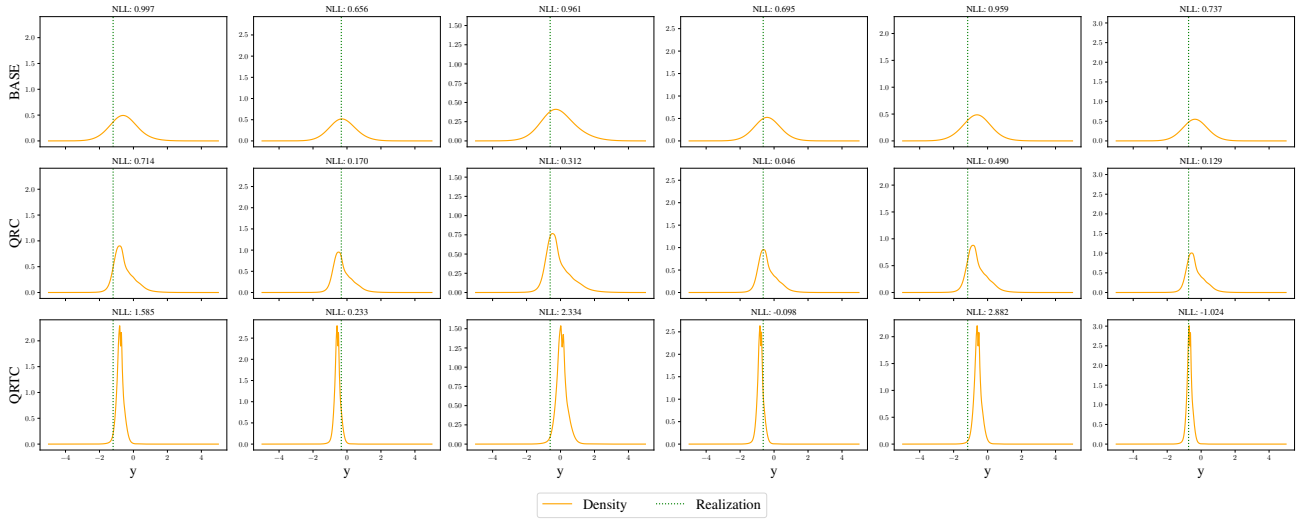


Figure 24: Predictions of `BASE`, `QRC` and `QRTC` on dataset `house_prices_nominal` (HO2).

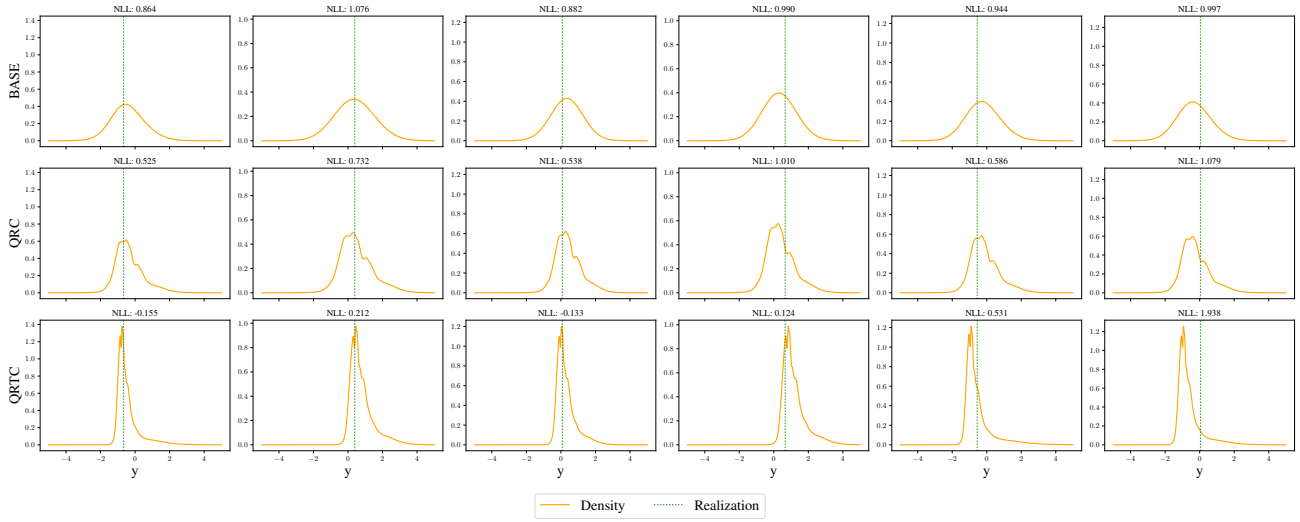


Figure 25: Predictions of `BASE`, `QRC` and `QRTC` on dataset `Mercedes_Benz_Greener_Manufacturing` (MER).

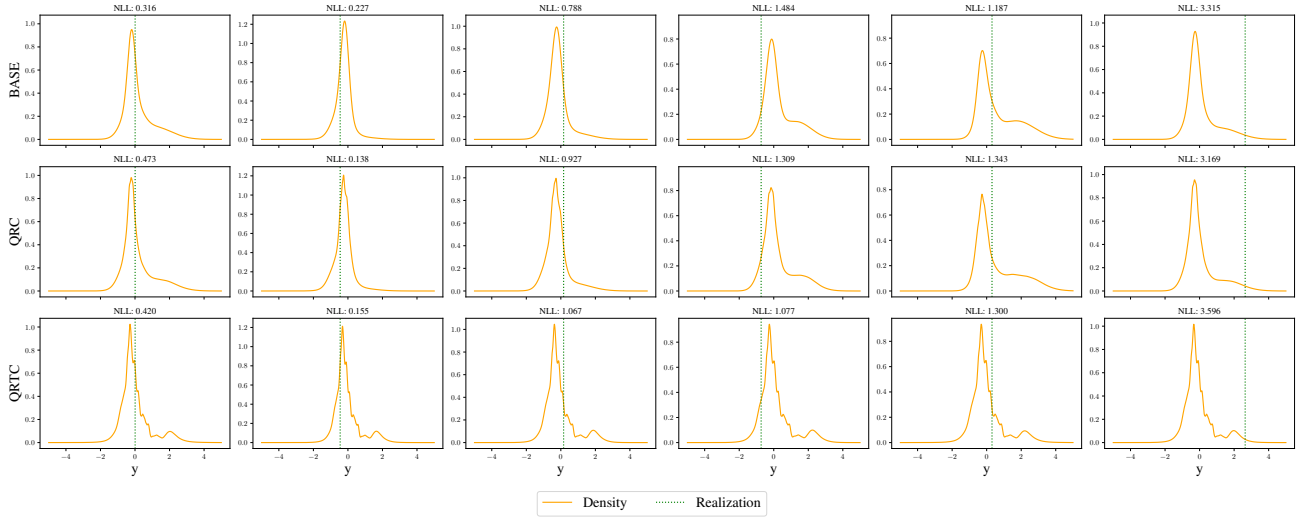


Figure 26: Predictions of `BASE`, `QRC` and `QRTC` on dataset `yprop_4_1` (YPR).

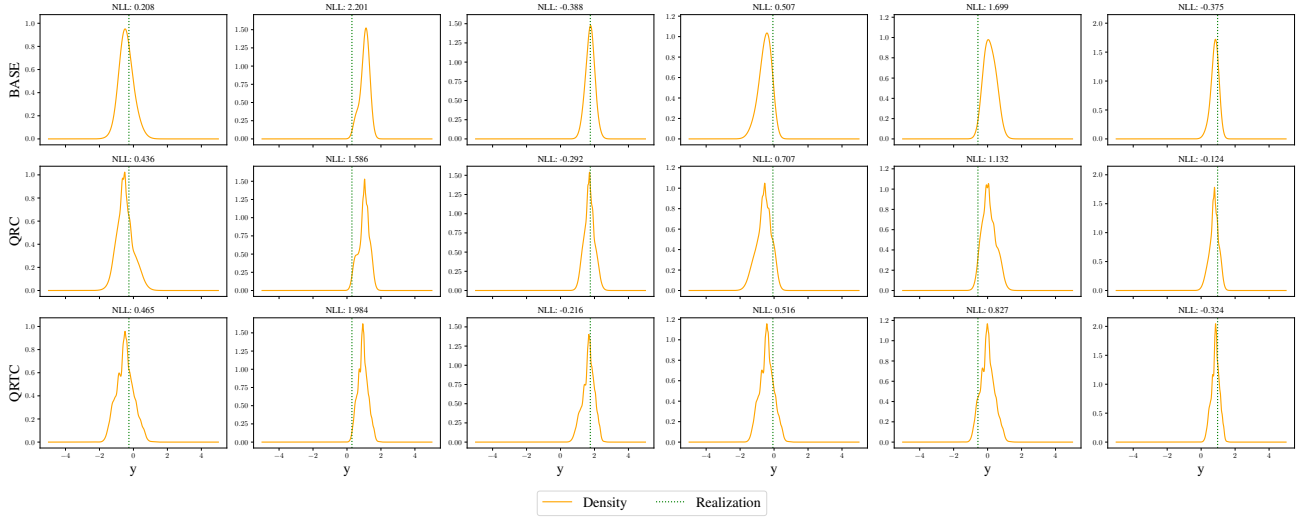


Figure 27: Predictions of `BASE`, `QRC` and `QRTC` on dataset `space_ga` (SPA).

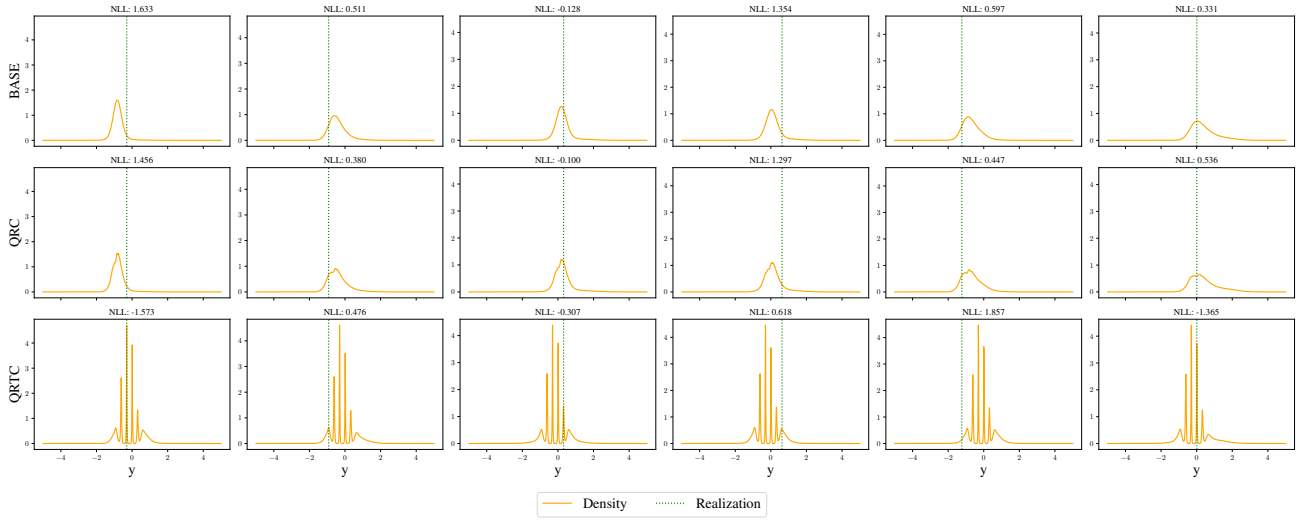


Figure 28: Predictions of `BASE`, `QRC` and `QRTC` on dataset `abalone` (ABA).