Beyond Pinball Loss: Quantile Methods for Calibrated Uncertainty Quantification

Youngseog Chung [1 2] **Willie Neiswanger** [3] **Ian Char** [2] **Jeff Schneider** [1]

1. Introduction and Preliminaries

Uncertainty quantification (UQ) in machine learning typically refers to the task of quantifying the confidence of a given prediction. This measure of certainty can be crucial in a variety of downstream applications, including Bayesian optimization (Mockus et al., 1978; Shahriari et al., 2015), model-based reinforcement learning (Garcia and Fernández, 2012; Chua et al., 2018; Malik et al., 2019; Yu et al., 2020), and in high-stakes predictions where mistakes incur large costs (Wexler, 2017; Rudin, 2019).

While the common goal of UQ is to describe predictive distributions over outputs for given inputs, the representation of the distributional prediction varies across methods. For example, some methods assume a parametric distribution and return parameter estimates (Lakshminarayanan et al., 2017; Detlefsen et al., 2019; Zhao et al., 2020), while others return density function estimates, as is common in Bayesian methods (Rasmussen, 2003; Koller and Friedman, 2009; Blundell et al., 2015; Hernández-Lobato and Adams, 2015; Maddox et al., 2019; Liu et al., 2019). Alternatively, many methods represent predictive uncertainty with quantile estimates (Jeon et al., 2016; Pearce et al., 2018; Tagasovska and Lopez-Paz, 2019; Fasiolo et al., 2020; Salem et al., 2020).

Quantiles provide an attractive representation for uncertainty because they can be used to model complex distributions without parametric assumptions, are interpretable with units in the target output space, allow for easy construction of prediction intervals, and can be used to efficiently sample from the predictive distribution via inverse transform sampling (Koenker and Hallock, 2001; Hao et al., 2007). Learning the quantile for a single quantile level is a well studied problem in quantile regression (QR) (Koenker and Hallock, 2001; Koenker, 2005), which typically involves optimizing the so-called *pinball loss*, a tilted transformation of the absolute value function. Given a target y, a prediction $\hat{y}$, and quantile level $\tau \in (0, 1)$, the pinball loss ρ_τ is defined as

$$\rho_\tau(y, \hat{y}) = (\hat{y} - y)(\mathbb{I}\{y \leq \hat{y}\} - \tau). \tag{1}$$

By training for all quantiles simultaneously, recent works have made concrete steps in incorporating QR methods to form competitive UQ methods which output the full predictive distribution (Tagasovska and Lopez-Paz, 2019; Rodrigues and Pereira, 2020).

In this work, we highlight some limitations of the pinball loss and propose several methods to address these shortcomings. Specifically, we explore the following[1]:

- **Model agnostic QR.** Optimizing the pinball loss often restricts the choice of model family for which we can provide UQ. We propose an algorithm to learn all quantiles simultaneously by utilizing methods from conditional density estimation. This algorithm is agnostic to model class and can be applied to *any* regression model.
- **Explicitly balancing calibration and sharpness.** While the pinball loss, as a proper scoring rule, targets both calibration and sharpness, the balance between these two quantities is made implicitly, which may result in a poor optimization objective. We propose a tunable loss function that targets calibration and sharpness separately, and allows the end-user to set an *explicit* balance.

Preliminaries. In this work, bold upper case letters **X, Y** denote random variables; lower case letters x, y, denote their values; and calligraphic upper case letters $\mathcal{X}, \mathcal{Y}$ denote sets of possible values. We use $x \in \mathcal{X}$ to denote the input feature vector and $y \in \mathcal{Y}$ to denote the corresponding target. Additionally, we consider the regression setting where $\mathcal{Y} \subset \mathbb{R}$ and $\mathcal{X} \subset \mathbb{R}^n$. We use $\mathbb{F}_{\mathbf{X}}, \mathbb{F}_{\mathbf{Y}|x}, \mathbb{F}_{\mathbf{Y}}$ to denote the true cumulative distribution of the subscript random variable. For any $x \in \mathcal{X}$, we assume there exists a true conditional distribution $\mathbb{F}_{\mathbf{Y}|x}$ over $\mathcal{Y}$, and we assume $\mathbb{Q}_p(x)$ denotes the true p^{th} quantile of this distribution, i.e. $\mathbb{F}_{\mathbf{Y}|x}(\mathbb{Q}_p(x)) = p$. Any estimates of the true functions, $\mathbb{F}, \mathbb{Q}_p$, will be denoted with a hat, $\hat{\mathbb{F}}, \hat{\mathbb{Q}}_p$. We will specifically refer to any family

[1]Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA [2]Machine Learning Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA [3]Computer Science Department, Stanford, California, USA. Correspondence to: Youngseog Chung <youngsec@cs.cmu.edu>.

Presented at the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning., Copyright 2021 by the author(s).

[1]Code is available at https://github.com/YoungseogChung/calibrated-quantile-uq

of estimates for $\mathbb{Q}_p$, with $p \in (0,1)$, as a "quantile model", denoted $\hat{\mathbb{Q}} : \mathcal{X} \times (0,1) \to \mathcal{Y}$. Unless otherwise noted, we will always consider the *conditional* problem of estimating quantities in the target space $\mathcal{Y}$, conditioned on a value $x \in \mathcal{X}$.

One evaluation metric that many recent works have focused on is the notion of *calibration* and *sharpness* (Gneiting et al., 2007; Guo et al., 2017; Kuleshov et al., 2018; Song et al., 2019; Tran et al., 2020; Zhao et al., 2020; Fasiolo et al., 2020; Cui et al., 2020). Broadly speaking, sharpness quantifies the concentration of distributional predictions (Gneiting et al., 2007), while calibration requires that the probability of observing the target random variable below a predicted p^{th} quantile is equal to the *expected probability* p, for all $p \in (0,1)$. The most common form of calibration is **average calibration** (often referred to simply as "calibration" (Kuleshov et al., 2018; Cui et al., 2020)). Where $\hat{\mathbb{Q}}_p(x)$ is the estimated p^{th} quantile of $\mathbf{Y}|x$, average calibration requires $p = \mathbb{E}_{x\sim\mathbb{F}_\mathbf{X}}[\mathbb{F}_{\mathbf{Y}|x}(\hat{\mathbb{Q}}_p(x))]$, $\forall p \in (0,1)$. Denoting $\mathbb{E}_{x\sim\mathbb{F}_\mathbf{X}}[\mathbb{F}_{\mathbf{Y}|x}(\hat{\mathbb{Q}}_p(x))]$ as p^{obs}_{avg}, and given a dataset $D = \{(x_i, y_i)\}_{i=1}^N$ we can make the estimate $\hat{p}^{\text{obs}}_{avg}(D,p) = \frac{1}{N}\sum_{i=1}^N \mathbb{I}\{y_i \le \hat{\mathbb{Q}}_p(x_i)\}$. The degree of error in average calibration is commonly measured by *expected calibration error* (Guo et al., 2017; Cui et al., 2020; Tran et al., 2020), $\text{ECE}(D, \hat{\mathbb{Q}}) = \frac{1}{m}\sum_{j=1}^m \left|\hat{p}^{\text{obs}}_{avg}(D, p_j) - p_j\right|$.

Recent works have suggested a notion of calibration stronger than average calibration, called adversarial group calibration (Zhao et al., 2020). This form of calibration requires average calibration for *any subset of $\mathcal{X}$ with non-zero measure*. Denote $\boldsymbol{X}_\mathcal{S}$ as a random variable that is conditioned on being in the set $\mathcal{S}$. **Adversarial group calibration** requires that $p = \mathbb{E}_{x\sim\mathbb{F}_{\mathbf{X}_\mathcal{S}}}[\mathbb{F}_{\mathbf{Y}|x}(\hat{\mathbb{Q}}_p(x))]$, $\forall p \in (0,1)$ and for all measurable $\mathcal{S} \subset \mathcal{X}$. With a finite dataset, we can measure a proxy of adversarial group calibration by measuring the average calibration within all subsets of the dataset with sufficiently many points.

2. Methods

Utilizing Conditional Density Estimation for Model Agnostic QR. One drawback of many existing quantile-based UQ methods is that their training procedure requires differentiable models. In fact, most UQ methods require a specific class of models because of their modeling structure or their loss objective (e.g. Gaussian processes (Rasmussen, 2003), dropout (Gal and Ghahramani, 2016), latent variable models (Koller and Friedman, 2009), simultaneous pinball loss (Tagasovska and Lopez-Paz, 2019), and NLL-based losses (Lakshminarayanan et al., 2017)). This model restriction can be especially unfavorable in practical settings. A domain expert with an established point prediction model and compute infrastructure may want to add UQ without much additional overhead.

To address these issues, we can consider the following model-agnostic procedure. Instead of optimizing a designated loss function, we can consider splitting the given problem into two parts: estimate conditional quantiles directly from data, then regress onto these estimates. The benefit of this method is that, granted we can estimate the conditional quantiles accurately, we can use any regression model to predict these quantile estimates. Further, this regression task directly targets the goal of producing the true conditional quantiles. This procedure, which we refer to as *Model Agnostic QR* (MAQR), is outlined in Algorithm 1.

MAQR is based on the key assumption that nearby points in $\mathcal{X}$ will have similar conditional distributions, i.e. if $x_j \approx x_k$ then $\mathbb{F}_{\mathbf{Y}|x_j} \approx \mathbb{F}_{\mathbf{Y}|x_k}$. Given this smoothness assumption, we can group neighboring points to estimate the conditional density at each locality over $\mathcal{X}$, with locality determined by the hyperparameter d_N (Algorithm 2, line 2). We then construct an empirical CDF with the group of neighboring points, and conditional quantile estimates are produced with this empirical CDF. These estimates are collected into D, which is ultimately used as the training set for the quantile model $\hat{g}$. In practice, we perform these steps with *residuals*, by first estimating a mean function $\hat{f}$ (Algorithm 1, line 1), which produces more accurate empirical CDFs. This method takes advantage of the fact that accurate point prediction models often already exist in many application settings.

Algorithm 1 is a specific implementation of a more general model-agnostic algorithm, in which we directly estimate conditional quantiles from the data with tools from conditional density estimation. We note that using KDEs for conditional density estimation is a well studied problem with theoretical guarantees (Stute et al., 1986; Hyndman et al., 1996; Holmes et al., 2012). In the case distance in $\mathcal{X}$ is measured using a uniform kernel with mild assumptions on the bandwidth, Algorithm 1 falls under the guarantees stated by Stute et al. (1986) (see Appendix D.1).

Explicitly balancing calibration and sharpness with the combined calibration loss. While MAQR can produce strong results, its performance can suffer in high-dimensional settings, where nonparametric conditional density estimation methods falter. Neural networks (NNs) have shown good performance in high dimensional settings, given their high capacity to approximate complex functions and recent advances in fast gradient-based optimization. We therefore propose a loss-based approach to estimating conditional quantiles for NNs and other differentiable models.

The most common loss function to use for such a procedure is the pinball loss. While this loss function has some nice properties, such as being a so-called *proper scoring rule* (Gneiting and Raftery, 2007), we note that the balance between calibration and sharpness implied by the pinball loss is arbitrary and depends on the expressivity of the model

Algorithm 1 MAQR

Input: Train data $\{x_i, y_i\}_{i=1}^N$, trained regression model $\hat{f}(x)$
Calculate residuals $\epsilon_i = y_i - \hat{f}(x_i),\ i \in [N]$
Initialize $D \leftarrow \varnothing$
for $i = 1$ **to** N **do**
  $D_i \leftarrow$ CONDQUANTILESESTIMATORS(D, i) (Algorithm 2)
  D: $D \leftarrow D \cup D_i$
end for
Use D to fit a regression model $\hat{g}$
  $\hat{g} : (x, p) \mapsto \epsilon$
Output: $\hat{f} + \hat{g},\quad k = 1, \ldots, m$

Algorithm 2 CONDQUANTILESESTIMATORS

Input: Dataset $\{x_i, \epsilon_i\}_{i=1}^N$, point index $k \in [N]$
$E_{k,d_N} \leftarrow \{\epsilon_i : \text{dist}(x_k, x_i) \leq d_N, i \in [N]\}$
Construct an empirical CDF with E_{k,d_N} to produce $\hat{\mathbb{F}}_{\mathbf{E}|x_k} : \epsilon \mapsto p \in [0, 1]$
Initialize $D \leftarrow \varnothing$
for each ϵ_j in E_{k,d_N} **do**
  Query $\hat{\mathbb{F}}_{\mathbf{E}|x_k}$ at ϵ_j for $\hat{p}_{k,j} = \hat{\mathbb{F}}_{\mathbf{E}|x_k}(\epsilon_j)$
  D: $D \leftarrow D \cup \{x_k, \hat{p}_{k,j}, \epsilon_j\}$
end for
Output: D

class. With highly expressive models, this balance can be heavily skewed towards sharpness. We flesh out this argument more clearly in Appendix B.

To address this failing of the pinball loss, we propose objectives separately for calibration and sharpness, Then, we combine the two objectives into a single loss function that provides an explicit balance between calibration and sharpness which can be chosen by the end user.

We first consider calibration of a quantile prediction, $\hat{\mathbb{Q}}_p \in \mathcal{Y}$ for quantile level $p \in (0, 1)$. Here, we omit conditioning on x for clarity. For this prediction to be average calibrated, exactly a p proportion of the true density should lie below $\hat{\mathbb{Q}}_p$, i.e. $P(Y \leq \hat{\mathbb{Q}}_p) = p$. While calibration (e.g. $|p_{avg}^{\text{obs}} - p|$) is a non-differentiable objective, by inducing a truncated distribution based on the current level of calibration, we can construct the following calibration objective, which is minimized if and only if the prediction is average calibrated:

$$\mathcal{C}(\hat{\mathbb{Q}}, p) = \mathbb{I}\{\hat{p}_p < p\} * \mathbb{E}[Y - \hat{\mathbb{Q}}_p | Y > \hat{\mathbb{Q}}_p] * P(Y > \hat{\mathbb{Q}}_p) + \mathbb{I}\{\hat{p}_p > p\} * \mathbb{E}[\hat{\mathbb{Q}}_p - Y | \hat{\mathbb{Q}}_p > Y] * P(\hat{\mathbb{Q}}_p > Y),$$

where $\hat{p}_p = P(Y \leq \hat{\mathbb{Q}}_p)$. The empirical calibration objective, $\mathcal{C}(D, \hat{\mathbb{Q}}, p)$ is defined in Equation 2 (Appendix A).

Note 1: *Intuition of the calibration objective.* For any given p, consider the case when the quantile estimate $\hat{\mathbb{Q}}_p$ is below the true p^{th} quantile $\mathbb{Q}_p$. Since $\hat{\mathbb{Q}}_p < \mathbb{Q}_p \iff \hat{p}_p < p$, this implies that too much density lies above $\hat{\mathbb{Q}}_p$. In this case, $\mathcal{C}(\hat{\mathbb{Q}}, p)$ reduces to $\mathbb{E}[Y - \hat{\mathbb{Q}}_p | Y > \hat{\mathbb{Q}}_p] * P(Y > \hat{\mathbb{Q}}_p)$. $\hat{\mathbb{Q}}_p$ is pulled higher with the expectation of the truncated distribution that places $\hat{\mathbb{Q}}_p$ at the lower bound of the support. In the opposite case, when $\hat{\mathbb{Q}}_p > \mathbb{Q}_p$, $\hat{\mathbb{Q}}_p$ is pulled lower by the same logic.

Note 2: *Is the proposed calibration objective a proper scoring rule?* Strictly speaking, the calibration objective is a non-decomposable function, hence deviates from the standard convention of proper scoring rules (Gneiting and Raftery, 2007), which can be "decomposed" into scores for individual examples (x_i, y_i). This simply arises from the fact that measuring average calibration (i.e. $\hat{p}_{avg}^{\text{obs}}$) is non-decomposable. Proper scoring rules are defined such that an optimum of the *expected score* (or *risk*, if we consider the score as a *loss function*) occur at the true distribution quantity. While an example level *loss* or *score* does not exist due to non-decomposability, we can still show the (expectation-level) score (i.e. $\mathcal{C}(\hat{\mathbb{Q}}, p)$) is minimized by the true distribution and hence enjoys the optimum property of proper scoring rules.

Proposition 1. *For any quantile level $p \in (0, 1)$, the true quantile function $\mathbb{Q}_p$ minimizes the calibration objective, $\mathcal{C}(\hat{\mathbb{Q}}, p)$. Further, on a finite dataset D, the empirical calibration objective, $\mathcal{C}(D, \hat{\mathbb{Q}}, p)$, is minimized by an average calibrated solution on D, i.e. when $\hat{p}_{avg}^{obs}(D, p) = p$.*

Proof: The proof is given in Appendix C.1.

Note 3: *Non-zero gradients for miscalibrated predictions* $\hat{\mathbb{Q}}_p$. We can further show that for a miscalibrated quantile prediction, the gradients of $\mathcal{C}$ are always non-zero. When $\hat{p}_p < p$, $\partial\mathcal{C}(\hat{\mathbb{Q}}_p, p)/\partial\hat{\mathbb{Q}}_p = -P(Y > \hat{\mathbb{Q}}_p) < 0$. Thus increasing $\hat{\mathbb{Q}}_p$ decreases the objective $\mathcal{C}$. Similarly, when $\hat{p}_p > p$, $\partial\mathcal{C}(\hat{\mathbb{Q}}_p, p)/\partial\hat{\mathbb{Q}}_p = P(Y < \hat{\mathbb{Q}}_p) > 0$, and an analogous argument follows (proof in Appendix C.2).

Average calibration by itself is not a sufficient condition for meaningful UQ, hence we also desire *sharp* quantile models, with more-concentrated (less flat) distributions. We can induce this property in quantile predictions by predicting the $(1-p)^{\text{th}}$ quantile $\hat{\mathbb{Q}}_{1-p}(x_i)$ alongside each prediction $\hat{\mathbb{Q}}_p(x_i)$ and penalizing the width between the quantile predictions: $\mathcal{P}(\hat{\mathbb{Q}}, p) = \mathbb{E}\left[\left|\hat{\mathbb{Q}}_p - \hat{\mathbb{Q}}_{1-p}\right|\right]$. The empirical sharpness objective, $\mathcal{P}(D, \hat{\mathbb{Q}}, p)$ is defined in Equation 3 (Appendix A). It is important to note that the true underlying distribution will not have 0 sharpness if there is significant noise, and sharpness should be optimized subject to calibra-

	SQR	*mPAIC*	***Cali***	***MAQR***
Concrete	$9.3 \pm 1.5(\underline{7.0} \pm 1.0)$	$6.2 \pm 0.5(14.2 \pm 0.8)$	$5.6 \pm 0.8(17.3 \pm 1.5)$	$\mathbf{5.3} \pm 0.4(16.0 \pm 0.4)$
Power	$2.6 \pm 0.4(13.4 \pm 0.2)$	$5.2 \pm 0.4(13.5 \pm 0.3)$	$2.0 \pm 0.1(\underline{13.1} \pm 0.1)$	$\mathbf{1.6} \pm 0.3(19.9 \pm 0.2)$
Wine	$4.2 \pm 0.2(29.5 \pm 0.4)$	$10.3 \pm 0.3(37.7 \pm 0.5)$	$4.2 \pm 0.4(\underline{26.0} \pm 0.8)$	$\mathbf{2.7} \pm 0.2(39.3 \pm 0.5)$
Yacht	$9.4 \pm 0.9(\underline{1.0} \pm 0.1)$	$10.8 \pm 2.3(2.6 \pm 0.4)$	$8.3 \pm 0.6(2.0 \pm 0.4)$	$\mathbf{6.8} \pm 2.1(2.4 \pm 0.3)$
Naval	$9.7 \pm 1.6(3.5 \pm 0.4)$	$3.1 \pm 0.5(63.0 \pm 1.8)$	$5.9 \pm 0.7(3.0 \pm 0.2)$	$\mathbf{2.3} \pm 0.2(\underline{1.7} \pm 0.1)$
Energy	$9.8 \pm 0.8(\underline{2.0} \pm 0.1)$	$10.4 \pm 0.5(4.3 \pm 0.2)$	$5.8 \pm 0.4(3.6 \pm 0.3)$	$\mathbf{3.5} \pm 1.0(3.2 \pm 0.1)$
Boston	$9.0 \pm 0.8(\underline{9.3} \pm 0.7)$	$8.7 \pm 1.3(12.3 \pm 0.7)$	$8.5 \pm 1.5(10.9 \pm 0.6)$	$\mathbf{6.2} \pm 1.8(10.9 \pm 0.8)$
Kin8nm	$4.4 \pm 0.1(\underline{11.4} \pm 0.2)$	$6.6 \pm 0.4(17.0 \pm 0.5)$	$3.5 \pm 0.3(13.7 \pm 0.7)$	$\mathbf{1.8} \pm 0.4(17.1 \pm 0.1)$

Figure 1. **Average calibration (measured by ECE) and sharpness in parentheses.** The best mean ECE for each dataset has been bolded and the best mean sharpness has been underlined (all values multiplied by 100 for readability

tion. Therefore, we should only penalize sharpness when the data suggests our quantiles are too dispersed, i.e. when $\left|p_{avg}^{\text{obs}}(p) - p_{avg}^{\text{obs}}(1-p)\right|$, the *observed coverage* between the pair of quantiles $\hat{\mathbb{Q}}_p(x_i)$ and $\hat{\mathbb{Q}}_{1-p}(x_i)$, is greater than $|2p-1|$, the *expected coverage*. Combining the calibration and sharpness terms, we have the **combined calibration loss**

$$\mathcal{L}(D, \hat{\mathbb{Q}}, p) = (1-\lambda)\mathcal{C}(D, \hat{\mathbb{Q}}, p) + \lambda\mathcal{P}(D, \hat{\mathbb{Q}}, p).$$

The hyperparameter $\lambda \in [0, 1]$ sets the explicit balance between calibration and sharpness. Note that setting $\lambda = 0$ may not always be desirable, since optimizing $\mathcal{C}(D, \hat{\mathbb{Q}}, p)$ alone may converge to quantiles of the marginal distribution, $\mathbb{F}_{\mathbf{Y}}$. Further, in certain downstream applications that utilize UQ, a sharper prediction, even at the cost of worse calibration, may result in higher utility, and λ can be tuned according to the utility function of the application.

In our experiments, we tune λ by cross-validating with adversarial group calibration as it is the strictest notion of calibration that can be estimated with a finite dataset. To further encourage adversarial group calibration, we also draw carefully constructed batches during training. We describe this batching procedure in detail in Appendix E.1.

3. Experiments

We demonstrate the performance of our proposed methods on the standard 8 UCI datasets (Asuncion and Newman, 2007). To assess the predictions, we report average calibration and sharpness. We also evaluate on adversarial group calibration, centered interval calibration, check score, and interval score in Appendix F due to space restrictions. We describe each of these evaluation metrics and how each are calculated in Appendix E.3, and we include details on the experiment setup and hyperparameters in Appendix E.2.

We provide a comparison against current state-of-the-art UQ methods for which computing the above metrics is tractable. *SQR* (Tagasovska and Lopez-Paz, 2019) is an NN model that optimizes the pinball loss for a batch of random quantile levels $p \sim U(0, 1)$. *mPAIC* (Zhao et al., 2020) is an extension of probabilistic neural networks (Nix and Weigend, 1994; Lakshminarayanan et al., 2017) that optimizes a combination of the standard Gaussian NLL and a loss that induces calibration. While additional quantile and PI based UQ methods exist, they are mostly designed to output a single quantile level or interval coverage level, which makes measuring calibration extremely expensive (training up to 100 models separately). All results report the mean and error across 5 trials. Error bars and shaded bands in plots indicate ± 1 standard error.

Figure 1 displays average calibration-sharpness for all 8 UCI datasets. *MAQR* produces the best average calibrated models on 7 of the 8 UCI datasets. Adversarial group calibration (Figure 4, Appendix F) also indicates *MAQR* tends to achieve the lowest calibration error across *any* random subgroup of *any* size with more than one point (6 out of 8 datasets). While not quite as good as *MAQR*, *Cali* (combined calibration loss) achieves competitive average calibration on 7 out of 8 UCI datasets. Notably, *SQR* tends to produce the sharpest predictions across all datasets (often at the cost of worse calibration). The check score, interval score, and centered interval calibration in the tables in Appendix F also rank *MAQR*'s prediction as best on the UCI datasets. This is surprising since *SQR* trains neural networks of the same capacity to *explicitly minimize* the check score. This suggests that the distribution predicted by *MAQR* is fundamentally different as it utilizes direct estimates of the conditional distribution, while other methods all optimize a specific loss function.

Conclusion. In this work, we have proposed methods to improve quantile estimates for calibrated uncertainty quantification in regression. We assert that the pinball loss may not be an adequate objective to optimize in order to achieve calibration, and that our proposed methods provide better means of learning calibrated conditional quantiles. We have also extended the scope of regression models on which quantile-based UQ can be applied by developing a model-agnostic method. This can be of practical interest to users that have specific training infrastructure or preexisting regression procedures since these procedures can be leveraged to quantify uncertainty without additional overhead.

References

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pages 4754–4765, 2018.

Peng Cui, Wenbo Hu, and Jun Zhu. Calibrated reliable regression using maximum mean discrepancy. *Advances in Neural Information Processing Systems*, 33, 2020.

Nicki S Detlefsen, Martin Jørgensen, and Søren Hauberg. Reliable training and estimation of variance networks. *arXiv preprint arXiv:1906.03260*, 2019.

Matteo Fasiolo, Simon N Wood, Margaux Zaffran, Raphaël Nedellec, and Yannig Goude. Fast calibrated additive quantile regression. *Journal of the American Statistical Association*, pages 1–11, 2020.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

Javier Garcia and Fernando Fernández. Safe exploration of state and action spaces in reinforcement learning. *Journal of Artificial Intelligence Research*, 45:515–564, 2012.

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

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):243–268, 2007.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

Lingxin Hao, Daniel Q Naiman, and Daniel Q Naiman. *Quantile regression*. Number 149. Sage, 2007.

José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869, 2015.

Michael P Holmes, Alexander G Gray, and Charles Lee Isbell. Fast nonparametric conditional density estimation. *arXiv preprint arXiv:1206.5278*, 2012.

Rob J Hyndman, David M Bashtannyk, and Gary K Grunwald. Estimating and visualizing conditional densities. *Journal of Computational and Graphical Statistics*, 5(4): 315–336, 1996.

Soyoung Jeon, Christopher J Paciorek, and Michael F Wehner. Quantile-based bias correction and uncertainty quantification of extreme event attribution statements. *Weather and Climate Extremes*, 12:24–32, 2016.

Koenker. *Quantile Regression (Econometric Society Monographs; No. 38)*. Cambridge university press, 2005.

Roger Koenker and Kevin F Hallock. Quantile regression. *Journal of economic perspectives*, 15(4):143–156, 2001.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. *arXiv preprint arXiv:1807.00263*, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

Jeremiah Liu, John Paisley, Marianthi-Anna Kioumourtzoglou, and Brent Coull. Accurate uncertainty estimation and decomposition in ensemble learning. In *Advances in Neural Information Processing Systems*, pages 8950–8961, 2019.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pages 13153–13164, 2019.

Ali Malik, Volodymyr Kuleshov, Jiaming Song, Danny Nemer, Harlan Seymour, and Stefano Ermon. Calibrated model-based deep reinforcement learning. *arXiv preprint arXiv:1906.08312*, 2019.

Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of bayesian methods for seeking the extremum. *Towards global optimization*, 2(117-129): 2, 1978.

David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, volume 1, pages 55–60. IEEE, 1994.

Tim Pearce, Mohamed Zaki, Alexandra Brintrup, and Andy Neely. High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. *arXiv preprint arXiv:1802.07167*, 2018.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

Filipe Rodrigues and Francisco C Pereira. Beyond expectation: deep joint mean and quantile regression for spatiotemporal problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Tárik S Salem, Helge Langseth, and Heri Ramampiaro. Prediction intervals: Split normal mixture from quality-driven deep ensembles. In *Conference on Uncertainty in Artificial Intelligence*, pages 1179–1187. PMLR, 2020.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

Hao Song, Tom Diethe, Meelis Kull, and Peter Flach. Distribution calibration for regression. In *International Conference on Machine Learning*, pages 5897–5906. PMLR, 2019.

Winfried Stute et al. On almost sure convergence of conditional empirical distribution functions. *The Annals of Probability*, 14(3):891–901, 1986.

Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning. In *Advances in Neural Information Processing Systems*, pages 6414–6425, 2019.

Kevin Tran, Willie Neiswanger, Junwoong Yoon, Qingyang Zhang, Eric Xing, and Zachary W Ulissi. Methods for comparing uncertainty quantifications for material property predictions. *Machine Learning: Science and Technology*, 1(2):025006, 2020.

Rebecca Wexler. When a computer program keeps you in jail: How computers are harming criminal justice. *New York Times*, 13, 2017.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *arXiv preprint arXiv:2005.13239*, 2020.

Shengjia Zhao, Tengyu Ma, and Stefano Ermon. Individual calibration with randomized forecasting. *arXiv preprint arXiv:2006.10288*, 2020.

Appendix

A. Additional Definitions

- Empirical Calibration Objective, $\mathcal{C}(D, \hat{\mathbb{Q}}, p)$:

$$\begin{aligned}\mathcal{C}(D, \hat{\mathbb{Q}}, p) = \mathbb{I}\{\hat{p}^{\text{obs}}_{avg} < p\} * \frac{1}{N}\sum_{i=1}^{N}\left[(y_i - \hat{\mathbb{Q}}_p(x_i))\mathbb{I}\{y_i > \hat{\mathbb{Q}}_p(x_i)\}\right] \\ + \mathbb{I}\{\hat{p}^{\text{obs}}_{avg} > p\} * \frac{1}{N}\sum_{i=1}^{N}\left[(\hat{\mathbb{Q}}_p(x_i) - y_i)\mathbb{I}\{\hat{\mathbb{Q}}_p(x_i) > y_i\}\right].\end{aligned} \tag{2}$$

- Empirical Sharpness Objective, $\mathcal{P}(D, \hat{\mathbb{Q}}, p)$:

$$\mathcal{P}(D, \hat{\mathbb{Q}}, p) = \frac{1}{N}\sum_{i=1}^{N}\begin{cases}\hat{\mathbb{Q}}_{1-p}(x_i) - \hat{\mathbb{Q}}_p(x_i) & (p \leq 0.5) \\ \hat{\mathbb{Q}}_p(x_i) - \hat{\mathbb{Q}}_{1-p}(x_i) & (p > 0.5).\end{cases} \tag{3}$$

B. Highlighting Problems with Pinball Loss

In this appendix, we flesh out the argument that a problem with the pinball loss is that it depends on the model class and can often be skewed towards sharpness. In their seminal work on probabilistic forecasts, Gneiting and Raftery (2007) contend that the goal of probabilistic forecasting is to "maximize the sharpness of the predictive distribution subject to calibration", i.e. calibration should be first achieved and then sharpness optimized. We show that common machine learning methods that use the pinball loss objective may in fact lead to an arbitrary and miscalibrated UQ.

B.1. Toy Experiment

In Figure 2 we show experimentally via a synthetic dataset how pinball loss can become detached from calibration. The details of this experiment are as follows:

Dataset: The synthetic dataset is based on the Boston UCI dataset. A NN model, μ_θ, was trained on the train split of the Boston dataset to predict the mean by optimizing the MSE loss (same architecture and training details as described in Appendix E.2). Afterwards, all points, $\{(x_i, y_i)\}_{i=1}^N$, in the full Boston dataset were re-labelled with the prediction of the mean model, $(x_i, \mu_\theta(x_i))$. Then, uniform noise, ϵ_i was added to these re-labelled mean values to create the observations, $\tilde{y}_i$, i.e. $\tilde{y}_i = \mu_\theta(x_i) + \epsilon_i$. The uniform noise was 0 mean, with width of the support equal to 5% of the range of the mean values, i.e $\epsilon_i \sim \text{Unif}[-0.025 * (\max_i y_i - \min_i y_i), 0.025 * (\max_i y_i - \min_i y_i)]$. Thus synthetic dataset is $\{(x_i, \tilde{y}_i)\}_{i=1}^N$.

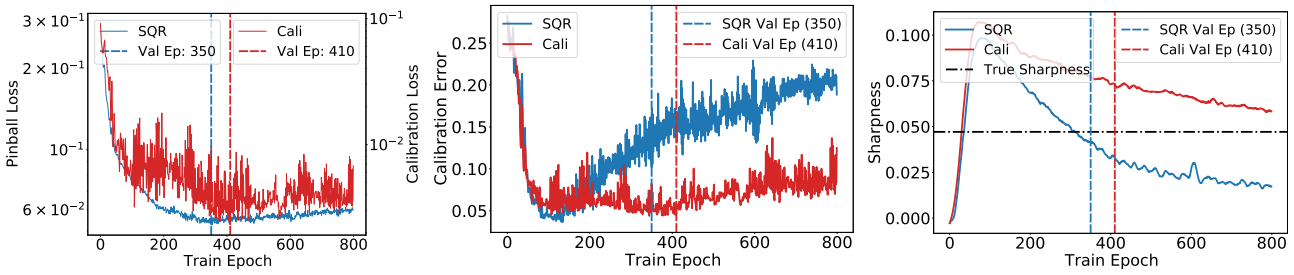
Procedure: The training procedure follows exactly that of the main experiments, which is described in detail in Appendix E.2. The only difference is that the models were trained for the full 1000 epochs, instead of halting training according to the validation loss.

We first note in Figure 2 (a) and (b) that even while the pinball loss decreases on the test set, test calibration worsens (while sharpness improves). Further, at the best validation epoch, optimizing the pinball loss converges to a solution that is sharper than the true noise level. Note that a UQ that is sharper than the true noise level will *never* be calibrated (meanwhile, a less sharp prediction *can still be calibrated*, e.g. the marginal distribution $\mathbb{F}_{\mathbf{Y}}$). While this may seem like an issue that can simply be addressed with regularization, we demonstrate in Appendix B.2 how that is not the case.

B.2. Regularizing the Pinball Loss

At first glance, regularization may appear to be the answer to what seems like an overfitting problem with the pinball loss, however, the problem is deeper and more specific to the loss function. In this section, we empirically demonstrate the effect of applying regularization when minimizing the pinball loss.

With the *SQR* method (which optimizes the pinball loss simultaneously for random batches of quantile levels), we applied L1, L2, and dropout, by cross-validating the regularization coefficients in $\{2^i, i \in \text{np.linspace}(-13, 1, 40)\}$ for L1 and L2, and dropout probability p in $\{2^i, i \in \text{np.linspace}(-13, -1, 40)\}$, for the pinball loss criterion (i.e. cross-validate among 40



(a) Test Loss Curves (b) Test Calibration (c) Test Sharpness

Figure 2. **(a)** Test loss continues to decrease until the validated epoch. **(b-c)** At the validated epoch, *SQR* (optimizes *pinball loss*) is highly miscalibrated while sharper than the true sharpness level. *Cali* (optimizes proposed *calibration loss*) is better calibrated while less sharp than the true sharpness.

	SQR	*SQR w/Reg*
concrete	$\mathbf{9.3} \pm 1.5(7.0 \pm 1.0)$	$10.9 \pm 1.0(\underline{6.9} \pm 0.6)$
power	$2.6 \pm 0.4(\underline{13.4} \pm 0.2)$	$\mathbf{1.0} \pm 0.1(14.8 \pm 0.1)$
wine	$\mathbf{4.2} \pm 0.2(29.5 \pm 0.4)$	$5.1 \pm 0.8(\underline{26.0} \pm 0.5)$
yacht	$\mathbf{9.4} \pm 0.9(\underline{1.0} \pm 0.1)$	$12.3 \pm 2.6(\underline{1.0} \pm 0.1)$
naval	$\mathbf{9.7} \pm 1.6(\underline{3.5} \pm 0.4)$	$11.5 \pm 2.8(\underline{3.5} \pm 0.3)$
energy	$9.8 \pm 0.8(2.0 \pm 0.1)$	$\mathbf{9.4} \pm 1.3(\underline{1.9} \pm 0.2)$
boston	$\mathbf{9.0} \pm 0.8(9.3 \pm 0.7)$	$11.6 \pm 1.1(\underline{8.6} \pm 0.8)$
kin8nm	$4.4 \pm 0.1(11.4 \pm 0.2)$	$\mathbf{3.5} \pm 0.3(\underline{11.2} \pm 0.2)$

Figure 3. **Applying Regularization with SQR: Average Calibration and Sharpness.** The table shows *SQR*'s mean ECE and sharpness (in parentheses) and their standard error with and without regularization. Among the 3 regularization methods (L1, L2, dropout), the method resulting in the best calibration is shown, for each dataset. The best mean ECE for each dataset has been bolded and the best mean sharpness has been underlined. All values have been multiplied by 100 for readability.

different regularization coefficients, between 2^{-13} and 2^{-1} on the log scale). We show the best calibrated regularization result, across all regularization methods and cross-validation in Figure 3.

Counter-intuitively, regularization tends to further the bias towards sharpness, and upon reflection, this may not be surprising because the range of quantile predictions shrink: given a quantile model $f : \mathcal{X} \times \mathcal{P} \to \mathcal{Y}$, where $\mathcal{P}$ is the space of quantile levels in $(0, 1)$, regularization affects the smoothness not only in $\mathcal{X}$, but also in $\mathcal{P}$, hence for any fixed $x \in \mathcal{X}$, the range in predictions for different quantile level inputs also shrinks.

C. Theoretical Results

C.1. Proof of Proposition 1

Proposition 1. *For any quantile level $p \in (0, 1)$, the true quantile function $\mathbb{Q}_p$ minimizes the calibration objective, $\mathcal{C}(\hat{\mathbb{Q}}, p)$. Further, on a finite dataset D, the empirical calibration objective, $\mathcal{C}(D, \hat{\mathbb{Q}}, p)$, is minimized by an average calibrated solution on D, i.e. when $\hat{p}^{obs}_{avg}(D, p) = p$.*

Proof:
Recall the calibration objective for a quantile level $p \in (0, 1)$,

$$\mathcal{C}(\hat{\mathbb{Q}}, p) = \mathbb{I}\{\hat{p}_p < p\} * \mathbb{E}[Y - \hat{\mathbb{Q}}_p | Y > \hat{\mathbb{Q}}_p] * P(Y > \hat{\mathbb{Q}}_p) + \mathbb{I}\{\hat{p}_p > p\} * \mathbb{E}[\hat{\mathbb{Q}}_p - Y | \hat{\mathbb{Q}}_p > Y] * P(\hat{\mathbb{Q}}_p > Y), \text{where } \hat{p}_p = P(Y \leq \hat{\mathbb{Q}}_p).$$

For the true quantile function $\mathbb{Q}_p$, $P(Y \leq \mathbb{Q}_p) = p$, thus achieves the minimum value of 0 for $\mathcal{C}(\hat{\mathbb{Q}}, p)$, as the two non-negative terms of $\mathcal{C}(\hat{\mathbb{Q}}, p)$ are 0.

Further, recall the empirical calibration objective,

$$\mathcal{C}(D, \hat{\mathbb{Q}}, p) = \mathbb{I}\{\hat{p}_{avg}^{\text{obs}} < p\} * \frac{1}{N}\sum_{i=1}^{N}\Big[(y_i - \hat{\mathbb{Q}}_p(x_i))\mathbb{I}\{y_i > \hat{\mathbb{Q}}_p(x_i)\}\Big] + \mathbb{I}\{\hat{p}_{avg}^{\text{obs}} > p\} * \frac{1}{N}\sum_{i=1}^{N}\Big[(\hat{\mathbb{Q}}_p(x_i) - y_i)\mathbb{I}\{\hat{\mathbb{Q}}_p(x_i) > y_i\}\Big].$$

An average calibrated solution on the dataset D satisfies $\hat{p}_{avg}^{\text{obs}} = p$, thus achieves the minimum value of 0 for $\mathcal{C}(D, \hat{\mathbb{Q}}, p)$, as the two non-negative terms of $\mathcal{C}(D, \hat{\mathbb{Q}}, p)$ are 0.

□

C.2. Derivation of Gradients of Calibration Objective

Denote the CDF and PDF of the random variable Y as $\mathbb{F}_Y$ and f_Y.

- When $\hat{p}_p = P(Y \leq \hat{\mathbb{Q}}_p) < p$:

$$\begin{aligned}
\mathcal{C}(\hat{\mathbb{Q}}_p, p) &= \mathbb{E}[Y - \hat{\mathbb{Q}}_p | Y > \hat{\mathbb{Q}}_p] * P(Y > \hat{\mathbb{Q}}_p) \\
&= \Big(\mathbb{E}[Y | Y > \hat{\mathbb{Q}}_p] - \hat{\mathbb{Q}}_p\Big) * P(Y > \hat{\mathbb{Q}}_p) \\
&= \left(\frac{\int_{\hat{\mathbb{Q}}_p}^{\infty} y f_Y(y) dy}{P(Y > \hat{\mathbb{Q}}_p)} - \hat{\mathbb{Q}}_p\right) * P(Y > \hat{\mathbb{Q}}_p) \\
&= \left(\int_{\hat{\mathbb{Q}}_p}^{\infty} y f_Y(y) dy\right) - \Big(\hat{\mathbb{Q}}_p * P(Y > \hat{\mathbb{Q}}_p)\Big) \\
&= \left(\int_{\hat{\mathbb{Q}}_p}^{\infty} y f_Y(y) dy\right) - \Big(\hat{\mathbb{Q}}_p * (1 - F_Y(\hat{\mathbb{Q}}_p))\Big)
\end{aligned}$$

  Note that,

$$\begin{aligned}
\frac{\partial\left(\int_{\hat{\mathbb{Q}}_p}^{\infty} y f_Y(y) dy\right)}{\partial \hat{\mathbb{Q}}_p} &= -\hat{\mathbb{Q}}_p * f_Y(\hat{\mathbb{Q}}_p) \\
\frac{\partial\left(\hat{\mathbb{Q}}_p * (1 - F_Y(\hat{\mathbb{Q}}_p)\right)}{\partial \hat{\mathbb{Q}}_p} &= (1 - F_Y(\hat{\mathbb{Q}}_p) + \hat{\mathbb{Q}}_P * (-f_y(\hat{\mathbb{Q}}_p))
\end{aligned}$$

  Therefore,

$$\begin{aligned}
\frac{\partial \mathcal{C}(\hat{\mathbb{Q}}_p, p)}{\partial \hat{\mathbb{Q}}_p} &= \frac{\partial\left(\int_{\hat{\mathbb{Q}}_p}^{\infty} y f_Y(y) dy\right)}{\partial \hat{\mathbb{Q}}_p} - \frac{\partial\left(\hat{\mathbb{Q}}_p * (1 - F_Y(\hat{\mathbb{Q}}_p)\right)}{\partial \hat{\mathbb{Q}}_p} \\
&= -\hat{\mathbb{Q}}_p * f_Y(\hat{\mathbb{Q}}_p) - (1 - F_Y(\hat{\mathbb{Q}}_p) + \hat{\mathbb{Q}}_P * f_y(\hat{\mathbb{Q}}_p) \\
&= -(1 - F_Y(\hat{\mathbb{Q}}_p) \\
&= -P(Y > \hat{\mathbb{Q}}_p)
\end{aligned}$$

- When $\hat{p}_p = P(Y \leq \hat{\mathbb{Q}}_p) > p$:

$$
\begin{aligned}
\mathcal{C}(\hat{\mathbb{Q}}_p, p) &= \mathbb{E}[\hat{\mathbb{Q}}_p - Y | \hat{\mathbb{Q}}_p > Y] * P(\hat{\mathbb{Q}}_p > Y) \\
&= \Big(\hat{\mathbb{Q}}_p - \mathbb{E}[Y | \hat{\mathbb{Q}}_p > Y]\Big) * P(\hat{\mathbb{Q}}_p > Y) \\
&= \left(\hat{\mathbb{Q}}_p - \frac{\int_{-\infty}^{\hat{\mathbb{Q}}_p} y f_Y(y) dy}{P(\hat{\mathbb{Q}}_p > Y)}\right) * P(\hat{\mathbb{Q}}_p > Y) \\
&= \Big(\hat{\mathbb{Q}}_p * P(\hat{\mathbb{Q}}_p > Y)\Big) - \left(\int_{-\infty}^{\hat{\mathbb{Q}}_p} y f_Y(y) dy\right) \\
&= \Big(\hat{\mathbb{Q}}_p * F_Y(\hat{\mathbb{Q}}_p)\Big) - \left(\int_{-\infty}^{\hat{\mathbb{Q}}_p} y f_Y(y) dy\right)
\end{aligned}
$$

Note that,

$$
\begin{aligned}
\frac{\partial \Big(\hat{\mathbb{Q}}_p * F_Y(\hat{\mathbb{Q}}_p)\Big)}{\partial \hat{\mathbb{Q}}_p} &= F_Y(\hat{\mathbb{Q}}_p) + \hat{\mathbb{Q}}_P * f_Y(\hat{\mathbb{Q}}_p) \\
\frac{\partial \left(\int_{-\infty}^{\hat{\mathbb{Q}}_p} y f_Y(y) dy\right)}{\partial \hat{\mathbb{Q}}_p} &= \hat{\mathbb{Q}}_p * f_Y(\hat{\mathbb{Q}}_p)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\frac{\partial \mathcal{C}(\hat{\mathbb{Q}}_p, p)}{\partial \hat{\mathbb{Q}}_p} &= \frac{\partial \Big(\hat{\mathbb{Q}}_p * F_Y(\hat{\mathbb{Q}}_p)\Big)}{\partial \hat{\mathbb{Q}}_p} - \frac{\partial \left(\int_{-\infty}^{\hat{\mathbb{Q}}_p} y f_Y(y) dy\right)}{\partial \hat{\mathbb{Q}}_p} \\
&= F_Y(\hat{\mathbb{Q}}_p) + \hat{\mathbb{Q}}_P * f_Y(\hat{\mathbb{Q}}_p) - \hat{\mathbb{Q}}_p * f_Y(\hat{\mathbb{Q}}_p) \\
&= F_Y(\hat{\mathbb{Q}}_p) \\
&= P(Y < \hat{\mathbb{Q}}_p)
\end{aligned}
$$

□

D. Model Agnostic Quantile Regression

D.1. Conditional Density Estimation Theory

Theorem 1 (Stute et al., 1986). *Assume $\mathcal{Y} \subset \mathbb{R}$, $\mathcal{X} \subset \mathbb{R}^n$, dist$(x_i, x_j) := |x_i - x_j|_\infty$, and that $\hat{\mathbb{F}}_{\mathbf{E}|x}$ is constructed using the procedure given in line 5 of Algorithm 1 (i.e. $x_i = x$). Further assume that, as $N \to \infty$, $d_N \to 0$ and that $\sum_{N \geq 1} \exp(-\rho N d_N^n) < \infty$, $\forall \rho > 0$. Then, as $N \to \infty$, for almost all $x \in \mathcal{X}$, $\sup_\epsilon[\hat{\mathbb{F}}_{\mathbf{E}|x}(\epsilon) - \mathbb{F}_{\mathbf{E}|x}(\epsilon)] \to 0$ with probability 1.*

This theorem states that in the limit of data, for almost all $x \in \mathcal{X}$, the CDF estimate $\hat{\mathbb{F}}_{\mathbf{E}|x}$ will converge uniformly to the true CDF $\mathbb{F}_{\mathbf{E}|x}$ with probability 1. The dataset, D, will therefore be populated with good estimates of the conditional quantile and quantile level pair for x. In Appendix D.2, we state the general form of Algorithm 1 and also demonstrate how the algorithm is model agnostic.

D.2. General Algorithm for Model Agnostic Quantile Regression

Algorithm 1 is one implementation of a general model-agnostic quantile regression procedure, in which we take direct estimates of the target density and regress onto these estimates. This general framework is state in Algorithm 3

Algorithm 1 implements the KDE step of Algorithm 3 (Line 5 of Algorithm 3) by using a uniform kernel over $\mathcal{X}$ (Line 2 of Algorithm 2) and $\mathcal{Y}$ (Lines 3, 5, 6, 7 of Algorithm 2).

Algorithm 3 General Algorithm for Model Agnostic Quantile Regression

Input: Train data $\{x_i, y_i\}_{i=1}^N$
Initialize $D \leftarrow \varnothing$
for $i = 1$ **to** N **do**
  Select a set of quantile levels $\{p_k\}_{k=1}^m,\ p_k \in [0,1]$
  $\hat{q}_{i,p_k} \leftarrow$ KDE estimate of $\mathbb{Q}(x_i, p_k)$, $k = 1, \ldots, m$
  $D \leftarrow D \cup \{x_i, p_k, \hat{q}_{i,p_k}\}_{k=1}^m$
end for
Use D to fit a regression model $\hat{\mathbb{Q}}$
  $\hat{\mathbb{Q}} : (x_i, p_k) \mapsto \hat{q}_{i,p_k}, \quad k = 1, \ldots, m$
Output: $\hat{g}, \quad k = 1, \ldots, m$

It should also be noted that many other conditional KDE methods can be used to construct the dataset D. We refer the reader to Hyndman et al. (1996); Holmes et al. (2012) for a more thorough treatment of methods in conditional KDE.

Lastly, this algorithm is model agnostic because *any* regression model can be used for $\hat{g}$ in Algorithm 3, and for $\hat{f}$ and $\hat{g}$ in Algorithm 1. In our specific implementation of Algorithm 1, we used a neural network for $\hat{g}$ to fit the quantile dataset D, but we can also use other models, such as a random forest or gradient-boosted trees. In particular, we have replaced $\hat{g}$ in Algorithm 1 with a gradient-boosted tree model and observed very similar UQ performance on the UCI datasets as reported in Section 3 (we omit numerical values because they are very similar and otherwise uninformative).

D.3. Algorithm Complexity

Lines 2 and 3 of Algorithm 2 requires calculating the distance between x_k and all other $x_i, 1 \leq i \leq N$, and ordering these distances to construct the empirical CDF. Let the distance calculation between a pair of points take constant C time. Ordering the distance requires sorting the N distances. Hence, Lines 5 and 6 takes $\mathcal{O}(N \log N)$ time.

If K points are in the set E_{k,d_N}, Lines 5, 6, 7 of Algorithm 2 are done for each of the K points. We consider the worst case when each set E_{k,d_N} contains all N points, which costs $\mathcal{O}(N)$.

The above two procedures are done for all N points (Line 4 of Algorithm 1), therefore the for loop from Lines 4 to 7 in Algorithm 1 requires $\mathcal{O}(N^2 \log N)$ time. This loop takes into account creating the dataset D for the quantile model. The rest of the algorithm constitutes fitting a regression model with this dataset D, which we do not analyze here.

We now discuss the space complexity. In Lines 5, 6, 7 of Algorithm 2, we only draw the quantiles at which a discontinuous step occurs in the constructed empirical CDF. For example, if we constructed an empirical CDF with three equally weighted points, we will only draw the quantiles $[1/3, 2/3, 1]$. Following this procedure, in the worst case, for each of the N points, we will construct a CDF with all N points, and hence draw N quantiles to append to the quantile model dataset D. If we consider the space complexity of the mean function and the quantile model to be constant, the algorithm requires $O(N^2)$ space in total.

E. Details on Setup of Main Experiments

E.1. Group Batching

Along with our proposed loss, we also find significant improvements can be made by altering the batching scheme. One condition we can additionally consider during training is *adversarial group calibration* (Zhao et al., 2020). Since adversarial group calibration requires average calibration over any subset of non-zero measure over the domain, this is not fully observable with finite datasets D for all subset sizes. However, for any subset in D with enough datapoints, we can still estimate average calibration over the subset. Hence, we can apply our optimization objectives onto appropriately large subsets to induce adversarial group calibration.

In practice, this involves constructing subsets within the domain and taking gradient steps based on the loss over each subset. In naive implementations of stochastic gradient descent, a random batch is drawn *uniformly* from the training dataset D, and a gradient step is taken according to the loss over this batch. This is also the case in *SQR* (Tagasovska and Lopez-Paz, 2019). The uniform draw of the batch will tend to preserve $\mathbb{F}_{\boldsymbol{X}}$ (the marginal distribution over $\mathcal{X}$), hence optimizing average

calibration over this batch will only induce average calibration of the model. Instead, deliberately grouping the datapoints based on input features, and then batching and taking gradient steps based on these groups, induces better adversarial group calibration.

Implementing group batching can be computationally demanding, because for each threshold setting, one has to make sure to choose a subset with sufficiently many points, and iteratively increase or decrease the dimension thresholds if no subset with sufficiently many points can be found. This computational cost increases significantly with dimension.

Our implementation of group batching for all our experiments were as follows: we sort the datapoints according to a single dimension, then take consecutive sets of size equal to the batch size, and use these sets as the batches to take gradient steps over during an epoch. We repeat the above process by cycling through each dimension for sorting. While this process is very simple and inexpensive and only considers a single dimension in constructing the subsets, this group batching scheme has shown to be very effective in our experiments.

E.2. UCI Experiment Details

Here, we provide details on the setup of the UCI experiments presented in Section 3. For each of the 8 UCI datasets, we split 10% of the data into the test set, and we further split 20% of the remaining 90% of the data for the validation, resulting in a train/validation/test split of proportions $72\%, 18\%, 10\%$. For all tasks and all 8 datasets, the data was preprocessed by centering to zero mean and scaling to unit variance.

We used the same NN architecture across all methods: 2 layers of 64 hidden units with ReLU non-linearities. We used the same learning rate, $1e^{-3}$, and the same batch size, 64, for all methods. For all methods, training was stopped early if the validation loss did not decrease for more than 200 epochs, until a maximum of 10000 epochs. If training was stopped early, the final model was backtracked to the model with lowest validation loss.

Cali (combined calibration loss) has one hyperparameter, λ, which balances the calibration loss and sharpness penalty in the loss function. We tuned λ according to the same grid as above for α, based on the criterion of adversarial group calibration (Zhao et al., 2020). Note that adversarial group calibration will not always favor lower values of λ as $\lambda = 0$ will only target average calibration, which, in the degenerate case, may converge to the marginal distribution of $\mathbb{F}_{\boldsymbol{Y}}$. This state will achieve very poor adversarial group calibration.

Group batching (Appendix E.1) was applied to *Cali*. During training, we alternated between "group batching epochs" and "regular batching epochs" (where batches are drawn uniformly from the training set), and the frequency of group batching epochs was a hyperparameter we tuned with cross-validation in [1, 2, 3, 5, 10, 30, 100], based on the criterion of adversarial group calibration.

MAQR has a two-step training process: we first learn a mean model, then construct a quantile dataset D, then regress onto this dataset with the quantile model. Both the mean model and the quantile model had the same NN architecture as mentioned above. The mean model was trained with the MSE loss according to the same, aforementioned training procedure. For each UCI dataset, we learned one mean model from the first seed, and re-used this mean model for all other seeds. Using this mean model, we then populated the quantile dataset according to the method outlined in Algorithm 2. Algorithm 2 requires one hyperparameter: the distance threshold in $\mathcal{X}$ space. We tuned this hyperparameter by setting the minimum distance required to include k number of points, on average, in constructing an empirical CDF at each training point. We tuned this parameter with cross-validation using the grid $k \in [10, 20, 30, 40, 50]$. The quantile model was trained according to the same training procedure but with one difference: the batch size was set to 1024 because the quantile dataset D could become very large due to many conditional quantile estimates at each training point.

All methods, for all datasets were repeated with 5 seeds: [0, 1, 2, 3, 4].

E.3. Calculation of Evaluation Metrics

To measure the calibration metrics (average, adversarial group, centered interval), we discretized the expected probabilities from 0.01 to 0.99 in 0.01 increments (i.e. $0.01, 0.02, \ldots, 0.97, 0.98, 0.99$) and estimated ECE according to this finite discretization.

To measure centered interval calibration, for each expected probability p, we predict centered $100 \times p\%$ PIs, and calculate

$\hat{p}^{\text{obs}}_{avg}$ as the proportion of test points falling within the PI, i.e. $\hat{p}^{\text{obs}}_{avg}(p)$ here would be calculated as

$$\hat{p}^{\text{obs}}_{avg}(p) \text{ for centered intervals } = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\{\hat{\mathbb{Q}}(x_i, 0.5 - p/2) \leq y_i \leq \hat{\mathbb{Q}}(x_i, 0.5 + p/2)\}.$$

The procedure in which we measure adversarial group calibration is the following. For a given test set, we scale group size between 1% and 100% of the full test set size, in 10 equi-spaced intervals, and for each group size, we draw 20 random groups from the test set and record the worst calibration incurred across these 20 random groups. This is also the method used by Zhao et al. (2020) to measure adversarial group calibration.

Sharpness was measured as the mean width of the 95% centered PI (i.e. width between quantiles at levels $p = 0.025$ and 0.975).

The check score and interval score are proper scoring rules, which is a family of evaluation metrics that summarize the quality of a distribution prediction (Gneiting and Raftery, 2007). The proper scoring rules metrics (check score, interval score) were calculated as the average of the score on the test set.

F. Full Experimental Results

In this section, we provide all of the experimental results from Section 3 for additional metrics. In addition to the check score (which is the same as the pinball loss), we evaluate using adversarial group calibration, interval score, and centered interval calibration (metrics and calculations described in Appendix E.3).

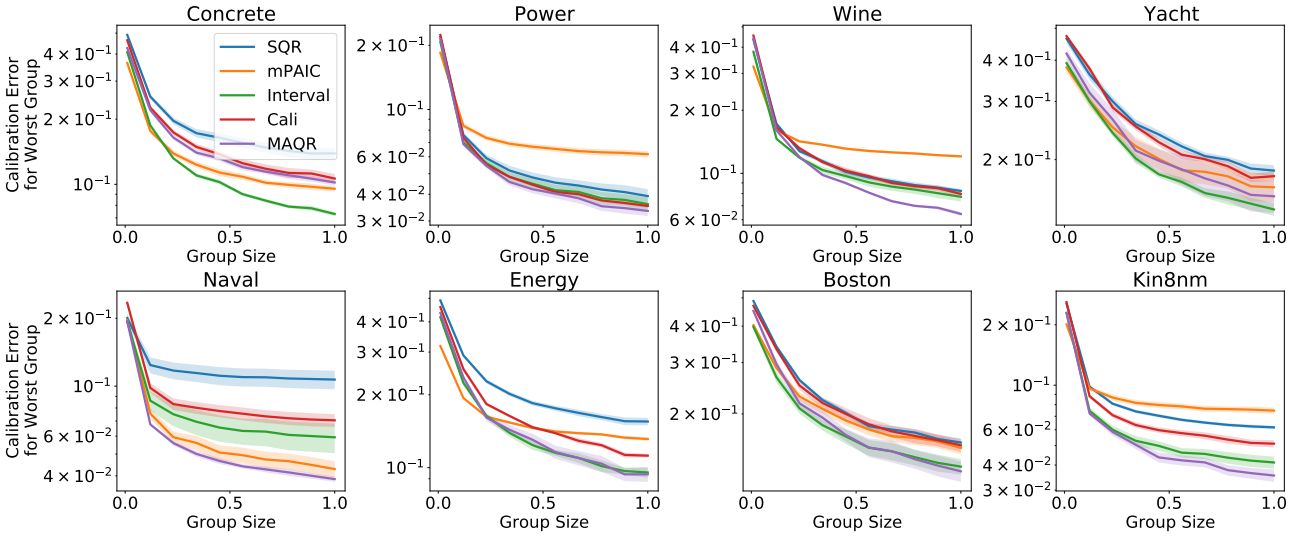


Figure 4. **UCI Adversarial Group Caibration.** The plot displays the worst calibration error incurred for any group of any given size. The mean is plotted along with ± 1 standard error in shades. Group size here refers to the proportion of the test dataset size.

	SQR	*mPAIC*	***Cali***	***MAQR***
concrete	$0.085(0.006)$	0.085 ± 0.005	0.118 ± 0.006	$\mathbf{0.059 \pm 0.008}$
power	$\mathbf{0.057 \pm 0.001}$	0.070 ± 0.001	0.064 ± 0.001	0.058 ± 0.001
wine	0.205 ± 0.008	0.219 ± 0.004	0.210 ± 0.008	$\mathbf{0.191 \pm 0.003}$
yacht	0.012 ± 0.002	0.015 ± 0.002	0.019 ± 0.004	$\mathbf{0.007 \pm 0.001}$
naval	0.070 ± 0.001	0.276 ± 0.004	0.159 ± 0.029	$\mathbf{0.004 \pm 0.000}$
energy	0.014 ± 0.000	0.015 ± 0.001	0.017 ± 0.002	$\mathbf{0.010 \pm 0.001}$
boston	0.088 ± 0.008	0.095 ± 0.008	0.103 ± 0.013	$\mathbf{0.063 \pm 0.016}$
kin8nm	0.078 ± 0.001	0.104 ± 0.003	0.096 ± 0.005	$\mathbf{0.070 \pm 0.001}$

Figure 5. **UCI Check Score.** Full check score results of UCI experiments. Mean score across 5 trials is given, along with ± 1 standard error. The best mean has been bolded. *MAQR* tends to achieve the best check score, which is surprising given that *SQR* utilizes the same model class to optimize the check score directly.

	SQR	*mPAIC*	***Cali***	***MAQR***
concrete	2.038 ± 0.225	1.157 ± 0.069	1.465 ± 0.086	$\mathbf{0.672 \pm 0.118}$
power	0.834 ± 0.022	0.917 ± 0.021	0.699 ± 0.019	$\mathbf{0.592 \pm 0.009}$
wine	3.242 ± 0.166	3.168 ± 0.019	2.498 ± 0.135	$\mathbf{2.052 \pm 0.052}$
yacht	0.314 ± 0.061	0.197 ± 0.036	0.298 ± 0.063	$\mathbf{0.086 \pm 0.016}$
naval	0.097 ± 0.011	3.112 ± 0.053	1.560 ± 0.268	$\mathbf{0.044 \pm 0.001}$
energy	0.290 ± 0.016	0.223 ± 0.017	0.204 ± 0.018	$\mathbf{0.101 \pm 0.006}$
boston	1.833 ± 0.299	1.395 ± 0.176	1.449 ± 0.259	$\mathbf{0.864 \pm 0.287}$
kin8nm	1.241 ± 0.041	1.347 ± 0.031	1.121 ± 0.072	$\mathbf{0.691 \pm 0.015}$

Figure 6. **UCI Interval Score** Full interval score results of UCI experiments. Mean score across 5 trials is given, along with ± 1 standard error. The best mean has been bolded. *MAQR* tends to achieve the best interval score, which is surprising given that *Interval* utilizes the same model class to optimize the interval score directly.

	SQR	*mPAIC*	***Cali***	***MAQR***
concrete	0.186 ± 0.031	0.089 ± 0.005	0.096 ± 0.013	$\mathbf{0.059 \pm 0.020}$
power	0.045 ± 0.004	0.068 ± 0.008	0.037 ± 0.002	$\mathbf{0.010 \pm 0.002}$
wine	0.053 ± 0.006	0.169 ± 0.008	0.065 ± 0.007	$\mathbf{0.045 \pm 0.005}$
yacht	0.135 ± 0.009	0.100 ± 0.020	0.129 ± 0.016	$\mathbf{0.085 \pm 0.024}$
naval	0.128 ± 0.031	0.039 ± 0.003	0.110 ± 0.013	$\mathbf{0.012 \pm 0.002}$
energy	0.174 ± 0.011	0.163 ± 0.009	0.090 ± 0.011	$\mathbf{0.052 \pm 0.018}$
boston	0.163 ± 0.020	$\mathbf{0.050 \pm 0.007}$	0.138 ± 0.028	0.092 ± 0.041
kin8nm	0.070 ± 0.005	0.080 ± 0.002	0.067 ± 0.005	$\mathbf{0.019 \pm 0.008}$

Figure 7. **UCI Centered Interval Calibration** Full centered interval calibration results of UCI experiments. Mean score across 5 trials is given, along with ± 1 standard error. The best mean has been bolded. *MAQR* tends to achieve the best centered interval calibration.