Flexible Model Aggregation for Quantile Regression

Rasool Fakoor[1], Taesup Kim[2], Jonas Mueller[3], Alexander Smola[1], and Ryan J. Tibshirani[1,4]

[1]Amazon Web Services, [2]Seoul National University, [3]Cleanlab, [4]Carnegie Mellon University

Abstract

Quantile regression is a fundamental problem in statistical learning motivated by a need to quantify uncertainty in predictions, or to model a diverse population without being overly reductive. For instance, epidemiological forecasts, cost estimates, and revenue predictions all benefit from being able to quantify the range of possible values accurately. As such, many models have been developed for this problem over many years of research in statistics, machine learning, and related fields. Rather than proposing yet another (new) algorithm for quantile regression we adopt a meta viewpoint: we investigate methods for aggregating any number of conditional quantile models, in order to improve accuracy and robustness. We consider weighted ensembles where weights may vary over not only individual models, but also over quantile levels, and feature values. All of the models we consider in this paper can be fit using modern deep learning toolkits, and hence are widely accessible (from an implementation point of view) and scalable. To improve the accuracy of the predicted quantiles (or equivalently, prediction intervals), we develop tools for ensuring that quantiles remain monotonically ordered, and apply conformal calibration methods. These can be used without any modification of the original library of base models. We also review some basic theory surrounding quantile aggregation and related scoring rules, and contribute a few new results to this literature (for example, the fact that post sorting or post isotonic regression can only improve the weighted interval score). Finally, we provide an extensive suite of empirical comparisons across 34 data sets from two different benchmark repositories.

1 Introduction

Consider the problem of assessing the height of a child. Common practice is to consult a growth chart, such as the ones provided by the CDC (Kuczmarski, 2000) and to review the distribution of heights, as relevant to the age and sex of the child. In doing so, the medical practitioner performs quantile regression, conditioning their estimates on covariates (age, sex) to obtain a conditional height distribution. While it would be possible to employ more standard regression methods for this problem, which would deliver an estimate of the mean height as a function of the covariates, quantile regression provides something more: it gives us a sense of what to expect in the spread of the response variable (height) as a function of the covariates.

The development of tools for conditional quantile estimation has a rich history in both econometrics and statistics. These tools are widely-used for quantifying uncertainty, and also for characterizing heterogeneous outcomes across diverse populations. As this is an important problem, many methods to arrive at quantile estimates abound. It stands to reason that a combination of different techniques can improve on the accuracy offered by individual base estimates. Precisely in this vein, the current paper considers the problem of *model aggregation*, i.e., the task of combining any number of quantile regression models into a unified estimator.

To fix notation, let $Y \in \mathbb{R}$ be a response variable of interest, and $X \in \mathcal{X}$ be an input feature vector used to predict Y. A generic way to approach uncertainty quantification is to estimate the conditional distribution of $Y|X = x$. However, this can be a formidable challenge, especially in high dimensions ($\mathcal{X} = \mathbb{R}^d$, where d is large). A simpler alternative is to estimate conditional quantiles of $Y|X = x$ across a discrete set of quantile levels $\mathcal{T} \subseteq [0, 1]$, that is, to estimate $g^*(x; \tau) = F^{-1}_{Y|X=x}(\tau)$ for $\tau \in \mathcal{T}$. Here, for a random variable Z with a cumulative distribution function (CDF) F_Z, we denote its level τ quantile by

$$F_Z^{-1}(\tau) = \inf\{z : F_Z(z) \geq \tau\}.$$

For example, we might choose $\mathcal{T} = \{0.01, 0.02, \ldots, 0.99\}$ to finely characterize the spread of $Y|X = x$.

The aggregation problem can be motivated as follows. Suppose we have a number of conditional quantile estimates, for instance, through a set of different quantile regression methods, or various teams submitting their estimates to a consensus board or as entries in a prediction competition. In all of these cases, we need an automated strategy to determine which estimate(s) of which quantile level(s) should be combined into a consensus model.

More formally, suppose we have a collection $\{\hat{g}_j\}_{j=1}^p$ of conditional quantile models, parametrized by a set of common quantile levels $\mathcal{T}$. Each model $\hat{g}_j$, which we refer to as a *base model*, provides an estimate of the true conditional quantile function g^*. It is convenient to view g^*, and each $\hat{g}_j$, as functions from $\mathcal{X}$ to $\mathbb{R}^m$, so that $g^*(x)$ outputs a vector of dimension $m = |\mathcal{T}|$, the number of quantile levels. We denote the components of this vector by $g^*(x;\tau)$ for $\tau \in \mathcal{T}$, and similarly for each $\hat{g}_j(x)$. Given $p \times m$ estimates of m quantile levels by p base models, each one a function of $x \in \mathcal{X}$, we will study ensemble estimates, of the generic form:

$$\hat{g} = H(\hat{g}_1, \ldots, \hat{g}_p) \, : \, \mathcal{X} \to \mathbb{R}^m.$$

In this paper, we focus on linear aggregation procedures H, though we allow the aggregation weights in these linear combinations to be themselves functions of input feature values x, as in:

$$\hat{g}_w(x) = \sum_{j=1}^p w_j(x) \cdot \hat{g}_j(x), \quad x \in \mathcal{X}. \tag{1}$$

This form may seem overly restrictive. That said, each term $\hat{g}_j$ on its own can be quite powerful. Moreover, as we show later, a sufficiently flexible parametrization for the weight functions can provide all the modeling power that we need. In particular, we will consider various aggregation strategies in which each weight $w_j(x)$ is a scalar, vector, or matrix. The product "$\cdot$" between $w_j(x)$ and $\hat{g}_j(x)$ in (1) is to be interpreted accordingly (more below).

Our main purpose in what follows is to provide a guided tour of how one might go about fitting quantile aggregation models of the form (1), of varying degrees of flexibility, and to walk through some of the major practical considerations that accompany fitting and evaluating such models. The aggregation strategies that we consider can be laid out over the following two dimensions.

1. *Coarse* versus *medium* versus *fine*: this dimension determines the resolution for the parametrization of the weights in (1).
   - A coarse aggregator uses one weight w_j per base model $\hat{g}_j$, and we accordingly interpret "$\cdot$" in (1) as a scalar-vector product.
   - A medium aggregator uses one weight w_j^τ per base model $\hat{g}_j$ and quantile level τ, and we interpret "$\cdot$" in (1) as a Hadamard (elementwise) product between vectors. This allows us to place a higher amount of weight for a given model in the tails versus the center the distribution.
   - A fine aggregator uses one weight $w_j^{\tau,\nu}$ per base model $\hat{g}_j$, output quantile level τ (for the output quantile), and input quantile level ν (from a base model), and we interpret "$\cdot$" in (1) as a matrix-vector product. This allows us to use all of the quantiles from all base models in order to form an estimate at a single quantile level for the aggregate model (e.g., the aggregate median is informed by all quantiles from all base models, not just their medians).
2. *Global* versus *local*: this dimension determines whether or not the weights in (1) depend on x. A global aggregator uses constant weights, $w_j(x) = w_j$ for all $x \in \mathcal{X}$, whereas a local aggregator allows these to vary locally (and typically smoothly) in x.

Apart from model frameworks, the considerations we give the most attention to revolve around ensuring *quantile noncrossing*: $\hat{g}(x;\tau) \leq \hat{g}(x;\tau')$ for any x and $\tau < \tau'$; and *calibration*: $\hat{g}(x;\tau') - \hat{g}(x;\tau)$ contains the response variable with probability $\approx \tau' - \tau$, at least in some average sense over $x \in \mathcal{X}$. Finally, we approach all of this work through the lens of the deep learning, designing methodology to be compatible with standard deep learning optimization toolkits so as to leverage their convenience of implementation and scalability.

Before delving into any further details, we present some of our main empirical results in Figure 1. Here and henceforth we use the term *deep quantile aggregation* (DQA) to refer to the aggregation model in our

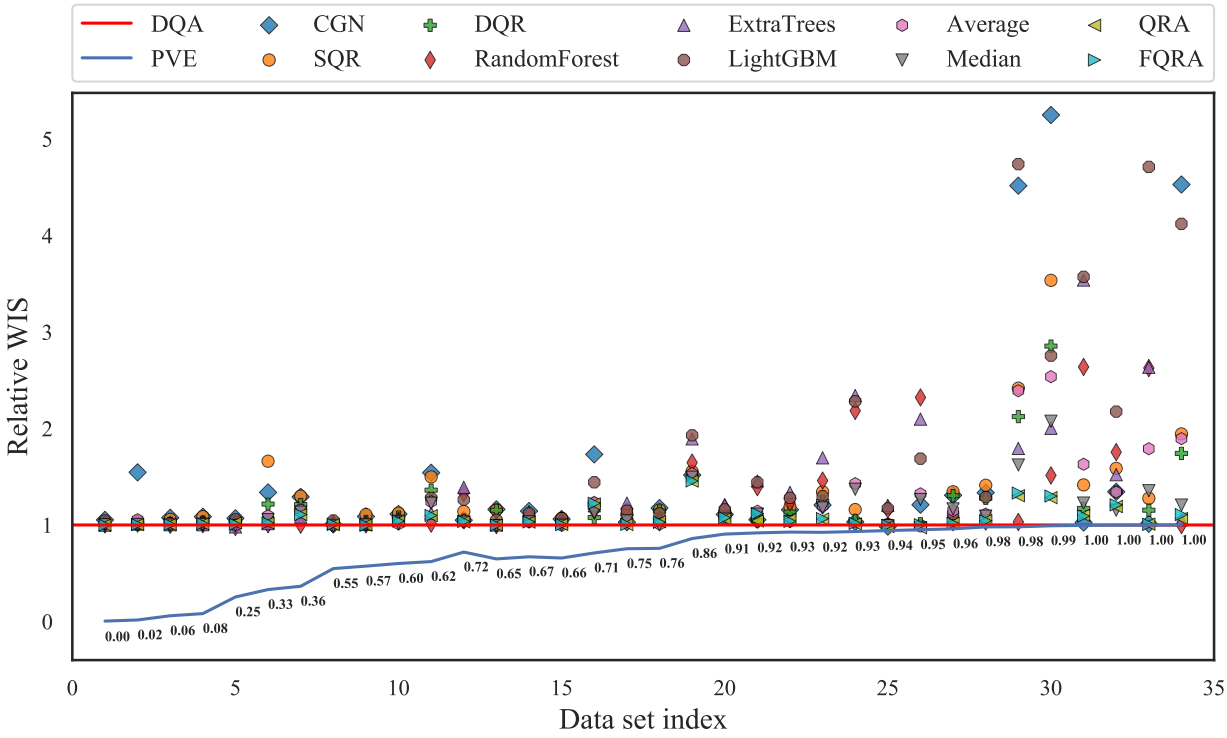

Figure 1: *Weighted interval score (WIS), averaged over out-of-sample predictions, for deep quantile aggregation (DQA) and various quantile regression methods. DQA is the most flexible aggregation model of the form (1) that we consider in this paper. The comparison is made over 34 data sets, ordered along the x-axis by proportion of variance explained (PVE), as in (2). The y-axis displays the WIS of each method relative to DQA, so that 1 indicates equal performance to DQA, and a number greater than 1 indicates worse performance than DQA. We can see that DQA performs very well overall, especially for larger PVE values (problems with higher signal-to-noise ratios).*

framework with the greatest degree of flexibility, a local fine aggregator where the local weights are fit using a deep neural network, and we use an adaptive noncrossing penalty, along with gradient propogation of the min-max sweep isotonization operator, to ensure quantile noncrossing (Section 4 provides more details). The figure compares DQA and various other quantile regression methods across 34 data sets (Section 6 gives more details). The error metric we use is *weighted interval score* (WIS), averaged over out-of-sample predictions (lower WIS is better; more details are given in the next section).

Shown on the y-axis is the relative WIS of each quantile regression method to DQA, where 1 indicates equal WIS performance to DQA, and a number greater than 1 indicates worse performance than DQA. Each point on the x-axis represents a data set, and we order them by increasing *proportion of variance explained* (PVE), measured with respect to DQA $\hat{g}$ (as a proxy for g^*) and test samples (X_i^*, Y_i^*), $i = 1, \ldots, m$ by:

$$\mathrm{PVE}(\hat{g}) = 1 - \frac{\sum_{i=1}^m (Y_i^* - \hat{g}(X_i^*; 0.5))^2}{\sum_{i=1}^m (Y_i^* - \bar{Y}^*)^2}, \tag{2}$$

where $\hat{g}(x; 0.5)$ denotes the estimate of the conditional median of $Y|X = x$ from DQA, and $\bar{Y}^* = \frac{1}{m}\sum_{i=1}^m Y_i^*$ is the sample mean of the test responses. The PVE curve itself is drawn in blue on the figure. The bottom four methods, according to the legend ordering, represent different aggregators, and the rest are individual base models (some are highly nonlinear and flexible themselves) that are fed into each aggregation procedure. As we can see, DQA performs very well across the board, and particularly for higher PVE values (which we can think of as problems that have higher signal-to-noise ratios, presenting a greater promise for the flexibility embodied by DQA), it can provide huge improvements on the base models and other aggregators. We should be clear that DQA is not uniformly better than all methods on all data sets—some points in the figure lie below 1. We provide an alternate visualization in in Appendix B.3 in which this is more clearly visible.

1.1 Related work

Quantile regression. Statistical modeling of quantiles dates back to Galton in the 1890s, however, many facts about quantiles were known long before (Hald, 1998). The modern view on conditional quantile models was pioneered by Koenker's work on *quantile regression* in the 1970s (Koenker and Bassett, 1978); e.g., see Koenker and Hallock (2001); Koenker (2005) for nice overviews. This has remained a topic of great interest, with developments in areas such as kernel machines (Takeuchi et al., 2006), additive models (Koenker, 2011), high-dimensional regression (Belloni and Chernozhukov, 2011), and graphical models (Ali et al., 2016), just to name a few. Important developments in distribution-free calibration using quantile regression were given in Romano et al. (2019); Kivaranovic et al. (2020) (which we will return to later). The rise of deep learning has spurred on new progress in quantile regression with neural networks, e.g., Hatalis et al. (2017); Dabney et al. (2018); Xie and Wen (2019); Tagasovska and Lopez-Paz (2019); Benidis et al. (2020).

Model aggregation. Ensemble methods occupy a central place in machine learning (both in theory and in practice). Seminal work on this topic arose in the 1990s on Bayesian model averaging, bagging, boosting, and stacking; e.g., see Dietterich (2000) for a review. While the machine learning literature has mostly focused on ensembling point predictions, distributional ensembles have a long tradition in statistics, with a classic reference being Stone (1961). Combining distributional estimates is also of great interest in the forecasting community, see Raftery et al. (2005); Timmermann (2006); Ranjan and Gneiting (2010); Gneiting and Ranjan (2013); Gneiting and Katzfuss (2014); Kapetanios et al. (2015); Rasp and Lerch (2018); Cumings-Menon and Shin (2020) and references therein.

To the best of our knowledge, the majority of work here has focused on combining probability densities, and there has been less systematic practical exploration of how to best combine quantile functions, especially from a flexible (nonparametric) perspective. Nowotarski and Weron (2015) proposed an aggregation method they call *quantile regression averaging* (QRA), which simply performs quantile linear regression on the output of individual quantile-parametrized base models. Variants of QRA have since been developed, for example, factor QRA (FQRA) (Maciejowska et al., 2016) which applies PCA to reduce dimensionality in the space of base model outputs before fitting the QRA aggregator. As perhaps evidence for the dearth of sophisticated aggregation models[1], simple quantile-averaging-type approaches have won various distributional forecasting competitions (Gaillard et al., 2016; Smyl and Hua, 2019; Browell et al., 2020). Lastly, we note that the study of quantile averaging actually dates back work by to Vincent (1912), and hence some literature refers to this method as *Vincentization*. See also Ratcliff (1979) for relevant historical discussion.

1.2 Outline

An outline for the remainder of this paper is as follows.

- Section 2 presents background material on quantile regression, error metrics, and quantile aggregation. This is primarily a review of relevant facts from the literature, but we do contribute a few small new results, in Propositions 2 and 5.
- Section 3 gives the framework for aggregation methods that we consider in this paper (parametrized by coarse/medium/fine weights on one axis, and global/local weights on the other, as explained earlier in the introduction).
- Section 4 investigates methods for ensuring quantile noncrossing while fitting aggregation models, both through explicit penalization, and use of differentiable isotonization operators.
- Section 5 discusses the use of conformal prediction (specifically, conformal quantile regression and CV+) to improve calibration post-aggregation.
- Section 6 provides a broad empirical evaluation of the proposed aggregation methods alongside various other aggregators and base models. Code to reproduce our all of experimental results is available at: `https://github.com/amazon-research/quantile-aggregation`.

[1] To be fair, this could also be a reflection of the intrinsic difficulty of the forecasting problems in these competitions; for a hard problem (low PVE), simple aggregators can achieve competitive performance with more complex ones, as seen in Figure 1.

2 Background

We cover important background material that will help to understand our contributions presented later.

2.1 Quantile regression and scoring rules

Quantile regression. We begin by recalling the definition of the *pinball loss*, also called the tilted-ℓ_1 loss, at a given quantile level $\tau \in [0,1]$. To measure the error of a level τ quantile estimate q against an observation Z, the pinball loss is defined by (Koenker and Bassett, 1978):

$$\psi_\tau(Z-q) = \begin{cases} \tau|Z-q| & Z-q \geq 0 \\ (1-\tau)|Z-q| & Z-q < 0. \end{cases} \tag{3}$$

For a continuously-distributed random variable Z, the expected pinball loss $\mathbb{E}[\psi_\tau(Z-q)]$ is minimized over q at the population-level quantile $q^*_\tau = F_Z^{-1}(\tau)$. This motivates estimation of q^*_τ given samples Z_i, $i = 1,\dots,n$ by minimizing the sample average of the pinball loss:

$$\underset{q}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^n \psi_\tau(Z_i - q).$$

Quantile regression does just this, but applied to the conditional distribution of $Y|X$. Given samples (X_i, Y_i), $i = 1,\dots,n$, it estimates the true quantile function $g^*(x;\tau) = F^{-1}_{Y|X=x}(\tau)$ by solving:

$$\underset{g\in\mathcal{G}}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^n \psi_\tau(Y_i - g(X_i)),$$

over some class of functions $\mathcal{G}$ (possibly with additional regularization in the above criterion). For example, quantile linear regression takes $\mathcal{G}$ to be the class of all linear functions of the form $g(x) = x^\mathsf{T}\beta$.

As we can see, quantile linear regression is quite a natural extension of ordinary linear regression—and quantile regression a natural extension of nonparametric regression more generally—where the focus moves from the conditional mean to the conditional quantile, but otherwise remains the same. The pinball loss (3) is convex in q but not differentiable at zero (unlike the squared loss, associated with mean estimation), which makes optimization slightly harder.

To model multiple quantile levels simultaneously, we can simply use *multiple* quantile regression, where we solve

$$\underset{g\in\mathcal{G}}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^n \sum_{\tau\in\mathcal{T}} \psi_\tau(Y_i - g(X_i;\tau)) \tag{4}$$

over a discrete set $\mathcal{T} \subseteq [0,1]$ of quantile levels. We will generally drop the qualifier "multiple", and refer to (4) as quantile regression.

Scoring rules. Another appeal of the pinball loss and the quantile regression framework is its connection to proper scoring rules from the forecasting literature, which we detail in what follows.

With Y as our response variable of interest, let P be a predicted distribution (forecast distribution) and S be a score function, which applied to P and Y, produces $S(P,Y)$. Adopting the notation of Gneiting and Raftery (2007), we write $S(P,Q) = \mathbb{E}_Q S(P,y)$, where $\mathbb{E}_Q$ denotes the expectation under $Y \sim Q$. Assuming that S is negatively oriented (lower values are better), recall that S is said to be *proper* if $S(P,Q) \geq S(Q,Q)$ for all P and Q. As Gneiting and Raftery put it: "In prediction problems, proper scoring rules encourage the forecaster to make careful assessments and to be honest."

For $Y \in \mathbb{R}$, denote by F the cumulative distribution function (CDF) associated with P. The *continuous ranked probability score* (CRPS) is defined by (Matheson and Winkler, 1976):

$$\text{CRPS}(F,Y) = \int_{-\infty}^{\infty} (F(y) - 1\{Y \leq y\})^2 \, \mathsf{d}y. \tag{5}$$

This is a well-known proper score, and is popular in various forecasting communities (e.g., in the atmospheric sciences), in part because it is considered more robust than the traditional log score. We also remark that CRPS is equivalent to the Cramér-von Mises divergence between F and the empirical CDF $1\{Y \leq \cdot\}$ based on just a single observation Y. As such, it is intimately connected to kernel scores, and more generally, to integral probability metrics (IPMs) for two-sample testing. For more, see Section 5 of Gneiting and Raftery (2007).

The following reveals an interesting relationship between CRPS (5) and the pinball loss function (3):

$$\int_{-\infty}^{\infty} (F(y) - 1\{Y \leq y\})^2 \, \mathsf{d}y = 2 \int_0^1 \psi_\tau(Y - F^{-1}(\tau)) \, \mathsf{d}\tau. \tag{6}$$

This appears to have been first noted by Laio and Tamea (2007). Their argument uses integration by parts, but it ignores a few subtleties, so we provide a self-contained proof of (6) in Appendix A.1. We can see from (6) that CRPS is equivalent (up to the constant factor of 2) to an integrated pinball loss, over all quantile levels $\tau \in [0, 1]$. This is quite interesting because these two error metrics are motivated from very different perspectives, not to mention different parametrizations (CDF space versus quantile space).

A natural approximation to the integrated the pinball loss is given by discretizing, as in:

$$\sum_{\tau \in \mathcal{T}} \psi_\tau(Y - F^{-1}(\tau)),$$

for a discrete set of quantile levels $\mathcal{T} \subseteq [0, 1]$.[2] The interesting connections now continue, in that the above is equivalent to what is known as the *weighted interval score* (WIS), a proper scoring rule for forecasts parametrized by a discrete set of quantiles. We assume the underlying quantile levels are symmetric around 0.5. The collection of predicted quantiles $F^{-1}(\tau)$, $\tau \in \mathcal{T}$ can then be reparametrized as a collection of central prediction intervals $(\ell_\alpha, u_\alpha) = (F^{-1}(\alpha/2), F^{-1}(1 - \alpha/2))$, $\alpha \in \mathcal{A}$ (each interval here is parametrized by its exclusion probability), and WIS is defined by:

$$\mathrm{WIS}(F^{-1}, Y) = \sum_{\alpha \in \mathcal{A}} \Big\{ \alpha(u_\alpha - \ell_\alpha) + 2 \cdot \mathrm{dist}(Y, [\ell_\alpha, u_\alpha]) \Big\}, \tag{7}$$

where $\mathrm{dist}(a, S)$ is the distance between a point a and set S (the smallest distance between a and an element of S). This scoring metric appears to have been first proposed in Bracher et al. (2021), though the *interval score* (an individual summand in (7)) dates back to Winkler (1972). WIS measures an intuitive combination of sharpness (first term in each summand) and calibration (second term in each summand). The equivalence between WIS and pinball loss is now as follows:

$$\sum_{\alpha \in \mathcal{A}} \Big\{ \alpha(u_\alpha - \ell_\alpha) + 2 \cdot \mathrm{dist}(Y, [\ell_\alpha, u_\alpha]) \Big\} = 2 \sum_{\tau \in \mathcal{T}} \psi_\tau(Y - F^{-1}(\tau)), \tag{8}$$

where $\mathcal{T} = \cup_{\alpha \in \mathcal{A}} \{\alpha/2, 1 - \alpha/2\}$. This is the result of simple algebra and is verified in Appendix A.2.

In summary, we have shown that in training a quantile regression model by optimizing pinball loss as in (4), we are already equivalently optimizing for WIS (7), and approximately optimizing for CRPS (5), where the quality of this approximation improves as the number of discrete quantile levels increases.

2.2 Noncrossing constraints and post hoc adjustment

An important consideration in fitting conditional quantile models is ensuring *quantile noncrossing*, that is, ensuring that the fitted estimate $\hat{g}$ satisfies:

$$\hat{g}(x; \tau) \leq \hat{g}(x; \tau') \quad \text{for all } x \text{ and } \tau < \tau'. \tag{9}$$

Two of the most common ways to approach quantile noncrossing are to use noncrossing constraints during estimation, or to use some kind of post hoc adjustment rule. In the former approach, we first specify a set $\mathcal{X}_0$

[2] We have omitted the adjustment of the summands for spacing between discrete quantile levels; note that this only contributes a global scale factor for evenly-spaced quantile levels.

at which we want to enforce noncrossing, and then solve a modified version of problem (4):

$$\begin{aligned} \underset{g \in \mathcal{G}}{\text{minimize}} \quad & \frac{1}{n} \sum_{i=1}^{n} \sum_{\tau \in \mathcal{T}} \psi_\tau(Y_i - g(X_i;\tau)) \\ \text{subject to} \quad & g(x;\tau) \leq g(x;\tau') \quad \text{for all } x \in \mathcal{X}_0 \text{ and } \tau < \tau', \end{aligned} \tag{10}$$

as considered in Takeuchi et al. (2006); Dette and Volgushev (2008); Bondell et al. (2010), among others (and even earlier in Fung et al. 2002 in a different context). The simplest choice is to take $\mathcal{X}_0 = \{X_i\}_{i=1}^n$, so as to enforce noncrossing at the training feature values; but in a transductive setting where we have unlabeled test feature values at training time, these could naturally be included in $\mathcal{X}_0$ as well.

The latter strategy, post hoc adjustment, has been studied in Chernozhukov et al. (2010); Kuleshov et al. (2018); Song et al. (2019) (and even earlier in Le et al. 2006 in a different context). In this approach, we solve the original multiple quantile regression problem (4), but then at test time, at any input feature value $x \in \mathcal{X}$, we output

$$\tilde{g}(x) = \mathcal{S}(\hat{g}(x)), \tag{11}$$

where $\mathcal{S} : \mathbb{R}^m \to \mathbb{K}^m$ is a user-chosen isotonization operator. Here, recall $m = |\mathcal{T}|$ is the number of discrete quantile levels, and $\mathbb{K}^m = \{v \in \mathbb{R}^m : v_i \leq v_{i+1},\ i = 1, \ldots, m-1\}$ denotes the isotonic cone in m dimensions. Two widely-used isotonization operators are *sorting*:

$$\texttt{Sort}(v) = (v_{(1)}, \ldots, v_{(m)}), \tag{12}$$

where we use the classic order statistic notation (here $v_{(i)}$ denotes the i^{th} largest element of v), and *isotonic projection*:

$$\texttt{IsoProj}(v) = \underset{u \in \mathbb{K}^m}{\text{argmin}} \ \|v - u\|_2. \tag{13}$$

The set $\mathbb{K}^m$ is a closed convex cone, which means the ℓ_2 projection operator onto $\mathbb{K}^m$ is well-defined (the minimization in (13) has a unique solution), and well-behaved (it is nonexpansive, i.e., Lipschitz continuous with Lipschitz constant $L = 1$, and therefore almost everywhere differentiable). Furthermore, there are fast linear-time algorithms for isotonic projection, such as the famous *pooled adjacent violators algorithm* (PAVA) of Barlow et al. (1972). Note that while $\texttt{Sort}(v) \in \mathbb{K}^m$, this does not mean that $\|\texttt{Sort}(v) - v\|_2$ is the shortest distance between v and $\mathbb{K}^m$. As such, the solutions to (12) and (13) differ in general.

The constrained approach (10) has the advantage that the constraints offer a form of regularization and for this reason, may help improve accuracy over post hoc techniques. It has the disadvantage of increasing the computational burden (depending on the nature of the model class $\mathcal{G}$, these constraints can actually be highly nontrivial to incorporate), and of only ensuring noncrossing over some prespecified finite set $\mathcal{X}_0$. By comparison, the post hoc approach (11) is computationally trivial (in the case of sorting (12) and isotonic projection (13)), and by construction, ensures noncrossing at each $x \in \mathcal{X}$.

Moreover, the post hoc approach is simple enough that it is possible to prove some general guarantees about its effect. The following is a transcription of some important results along these lines from the literature, for sorting and isotonic projection.

Proposition 1 (Chernozhukov et al. 2010; Robertson et al. 1998)**.** *Let $\mathcal{T} \subseteq [0,1]$ be an arbitrary finite set, and denote by*

$$g^*(x) = \big\{g^*(x;\tau)\big\}_{\tau \in \mathcal{T}} \quad \text{and} \quad \hat{g}(x) = \big\{\hat{g}(x;\tau)\big\}_{\tau \in \mathcal{T}}$$

an arbitrary true and estimated conditional quantile function at a point x (that is, $g^*(x)$ and $\hat{g}(x)$ are arbitary vectors in $\mathbb{K}^m$ and $\mathbb{R}^m$, respectively, where $m = |\mathcal{T}|$). Then the following holds for the post-adjusted estimate $\tilde{g}(x)$ in (11).

(i) If $\mathcal{S} = \texttt{Sort}$, the sorting operator (12), then the ℓ_p norm error between the estimate and $g^*(x)$ can only improve for any $p \geq 1$, that is, $\|\tilde{g}(x) - g^*(x)\|_p \leq \|\hat{g}(x) - g^*(x)\|_p$ for any $p \geq 1$. Moreover, if sorting is nontrivial: $\tilde{g}(x) \neq \hat{g}(x)$, and $p > 1$, then the ℓ_p error inequality is strict.

(ii) If $\mathcal{S} = \texttt{IsoProj}$, the isotonic projection operator (13), then the same result holds as in part (i).

Part (i) of this proposition is due to Chernozhukov et al. (2010) and is an application of the classical rearrangement inequality (Hardy et al., 1934). Part (ii) is due to Robertson et al. (1998).

In our ensemble setting, metrics on predictive accuracy are more relevant. Towards this end, the next proposition contributes a new but small result on post hoc adjustment and WIS.

Proposition 2. *As in the last proposition, let $\mathcal{T} \subseteq [0,1]$ be an arbitrary finite set, and $\hat{g}(x) = \{\hat{g}(x;\tau)\}_{\tau\in\mathcal{T}}$ be an estimate of the conditional quantile function at a point x. Then the following holds for the post-adjusted estimate $\tilde{g}(x)$ in (11), and for any $y \in \mathbb{R}$.*

(i) If $\mathcal{S} = \texttt{Sort}$, the sorting operator (12), then the pinball loss can only improve:

$$\sum_{\tau\in\mathcal{T}} \psi_\tau(y - \tilde{g}(x;\tau)) \leq \sum_{\tau\in\mathcal{T}} \psi_\tau(y - \hat{g}(x;\tau)).$$

When $\mathcal{T}$ is symmetric around 0.5, this means $\mathrm{WIS}(\tilde{g}(x), y) \leq \mathrm{WIS}(\hat{g}(x), y)$ as well (by the equivalence between pinball loss and WIS in (8)). Moreover, if sorting is nontrivial: $\tilde{g}(x) \neq \hat{g}(x)$, then the pinball or WIS improvement is strict.

(ii) If $\mathcal{S} = \texttt{IsoProj}$, the isotonic projection operator (13), then the same result holds as in part (i).

The proof of Proposition 2 elementary and given in Appendix A.3. Interestingly, the proof makes use of particular algorithms for sorting and isotonic projection (bubble sort and PAVA, respectively).

Note that, as the inequalities in Propositions 1 and 2 hold pointwise for each $x \in \mathcal{X}$, they also hold in an average (integrated) sense, with respect to an arbitrary distribution on $\mathcal{X}$. Later in Section 4, we compare and combine these and other noncrossing strategies, through extensive empirical evaluations. Analogous to noncrossing constraints in (10), we consider a crossing penalty (which is similar, but more computationally efficient); and as for rules like sorting and isotonic projection, we evaluate them not only post hoc, but also as layers in training.

2.3 Quantile versus probability aggregation

When it comes to combining uncertainty quantification models, we have a number of options: we can either average over probabilities or over quantiles. These strategies are quite different, and often lead to markedly different outcomes. This subsection provides some theoretical background comparing the two strategies. We see it as important to review this material because it is a useful guide to thinking about quantile aggregation, which is likely less familiar to most readers (than probability aggregation).

In this subsection only, the term *average* refers to a weighted linear combination where the weights are nonnegative and sum to 1. For each $j = 1, \ldots, p$, let F_j be a cumulative distribution function (CDF); $f_j = F_j'$ be its probability density function; let $Q_j = F_j^{-1}$ denote the corresponding quantile function; and $q_j = Q_j'$ the quantile density function. A standard fact that relates these objects:

$$q_j(u) = \frac{1}{f_j(Q_j(u))} \quad \text{and} \quad f_j(v) = \frac{1}{q_j(F_j(v))}. \tag{14}$$

The first fact can be checked by differentiating $Q_j(F_j(v)) = v$, applying the chain rule, and reparametrizing via $u = F_j(v)$. The second follows similarly via $F_j(Q_j(u)) = u$.

We compare and contrast two ways of averaging distributions. The first way is in probability space, where we define for weights $w_j \geq 0$, $j = 1, \ldots, p$ such that $\sum_{j=1}^p w_j = 1$,

$$F = \sum_{j=1}^p w_j F_j.$$

The associated density is simply $f = \sum_{j=1}^p w_j f_j$ since differentiation is a linear operator. The second way to average is in quantile space, defining

$$\bar{Q} = \sum_{j=1}^p w_j Q_j,$$

where now $\bar{q} = \sum_{j=1}^p w_j q_j$ is the associated quantile density, again by linearity of differentiation. Denote the CDF and probability density associated with the quantile average by $\bar{F} = \bar{Q}^{-1}$, and $\bar{f} = \bar{F}'$. Note that from (14), we can reason that $\bar{f}$ is a highly *nonlinear* function of f_j, $j = 1, \ldots, p$.

A simple example can go a long way to illustrate the differences between the distributions resulting from probability and quantile averaging. In Figure 2, we compare these two ways of averaging on a pair of normal distributions with different means and variances. Here we see that probability averaging produces the familiar mixture of normals, which is bimodal. The result of quantile averaging is very different: it is always unimodal, and instead of interpolating between the tail behaviors of f_1 and f_2 (as f does), it appears that *both* tails of $\bar{f}$ are generally thinner than those of f_1 and f_2.

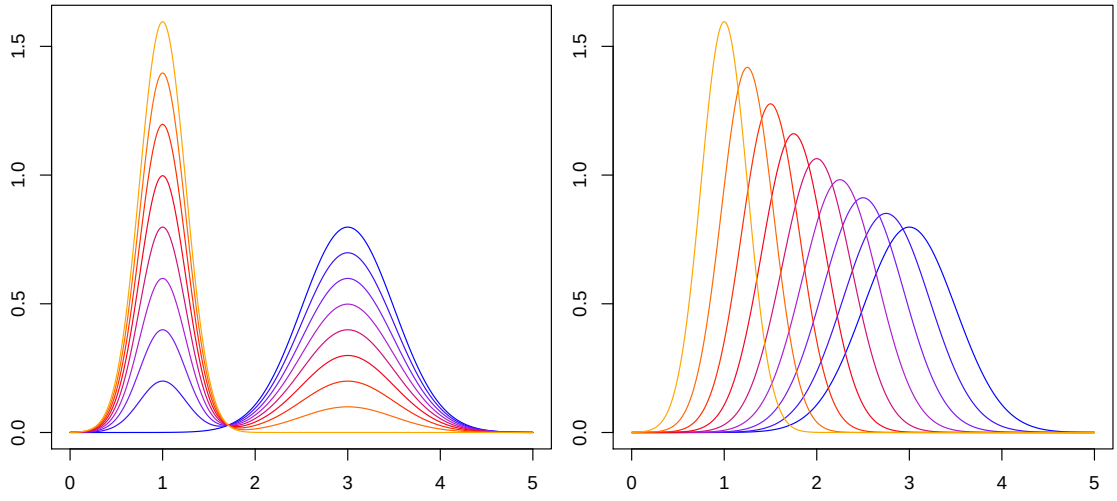

Figure 2: *Densities that result from a probability average (left) or quantile average (right) of two normal distributions* $N(1, 0.25^2)$ *and* $N(3, 0.5^2)$*, as the weight on the first density varies from 1 (orange) to 0 (blue).*

It seems that quantile averaging is doing something that is both like translation and scaling in probability density space. Next we explain this phenomenon precisely by recalling a classic result.

Shape preservation. An aggregation procedure H is said to be *shape-preserving* if, for any location-scale family $\mathcal{P}$ (such as Normal, t, Laplace, or Cauchy) whose elements differ only by scale and location parameters, we have

$$F_j \in \mathcal{P},\ j = 1, \ldots, p \implies F = H(F_1, \ldots, F_p) \in \mathcal{P}.$$

Probability averaging is clearly not shape-preserving, however, interestingly, quantile averaging is: if each F_j a member of the same location-scale family with a base CDF L, then we can write $F_j(v) = L((v - \theta_j)/\sigma_j)$, thus $Q_j(u) = \theta_j + \sigma_j L^{-1}(u)$, so $\bar{Q}$ is still of the form $\theta + \sigma L^{-1}$ and $\bar{F}$ is also in the location-scale family. The next proposition collects this and related results from the literature.

Proposition 3 (Thomas and Ross 1980; Genest 1992)**.**

(i) Quantile averaging is shape-preserving.

(ii) Location-scale families $\mathcal{P}$ are the only ones with respect to which quantile averaging is a closed operation (meaning $F_j \in \mathcal{P}$, $j = 1, \ldots, p$ implies $\bar{F} \in \mathcal{P}$).

(iii) Quantile averaging is the only aggregation procedure H, of those satisfying (for h not depending on u):

$$H(F_1, \ldots, F_p)^{-1}(u) = h(Q_1(u), \ldots, Q_p(u)),$$

that is shape-preserving.

Part (ii) is due to Thomas and Ross (1980). Part (iii) is due to Genest (1992), following from an elegant application of Pexider's equation.

The parts of Proposition 3, taken together, suggest that quantile averaging is somehow "tailor-made" for shape preservation in a location-scale family—which can be seen as either a pro or a con, depending on the application one has in mind. To elaborate, suppose that in a quantile regression ensembling application, each base model outputs a normal distribution for its predicted distribution at each x (with different means and variances). If the normal assumption is warranted (i.e., it actually describes the data generating distribution) then we would want our ensemble to retain normality, and quantile averaging would do exactly this. But if the normal assumption is used only as a working model, and we are looking to combine base predictions as a way to construct some flexible and robust model, then the shape-preserving property of quantile averaging would be problematic. In general, to model arbitrary distributions without imposing strong assumptions, we are therefore driven to use linear combinations of quantiles that allow *different aggregation weights to be used for different quantile levels*, of the form $\bar{Q}(u) = \sum_{j=1}^p w_j(u) Q_j(u)$.

Moments and sharpness. We recall an important result about moments of the distributions returned by probability and quantile averages. For a distribution G, we denote its uncentered moment of order $k \geq 1$ by $m_k(G) = \mathbb{E}_G[X^k]$, where the expectation is under $X \sim G$.

Proposition 4 (Lichtendahl et al. 2013)**.**

(i) A probability and quantile average always have equal means: $m_1(F) = m_1(\bar{F})$.

(ii) A quantile average is always sharper than a probability average: $m_k(\bar{F}) \leq m_k(F)$ *for any even* $k \geq 2$.

Note that sharpness is only a desirable property if it does not come at the expense of calibration. With this in mind, the above result cannot be understood as a pro or con of quantile averaging without any context on calibration. That said, the relative sharpness of quantile averages to probability averages is an important general phenomenon to be aware of.

Tail behavior. Lastly, we study the action of quantile averaging on the tails of the subsequent probability density. Simply starting from $\bar{q} = \sum_{j=1}^p w_j q_j$, differentiating, and using (14), we get

$$\frac{1}{\bar{f}(\bar{Q}(u))} = \sum_{j=1}^p \frac{w_j}{f_j(Q_j(u))}.$$

That is, the probability density $\bar{f}$ at the level u quantile is a (weighted) *harmonic mean* of the densities f_j at their respective level u quantiles. Since harmonic means are generally (much) smaller than arithmetic means, we would thus expect $\bar{f}$ to have thinner tails than f. The next result formalizes this. We use $g(v) = o(h(v))$ to mean $g(v)/h(v) \to 0$ as $v \to \infty$, and $g(v) \asymp h(v)$ to mean $g(v)/h(v) \to c \in (0, \infty)$ as $v \to \infty$.

Proposition 5. *Assume that* $p = 2$, $f_2(v) = o(f_1(v))$, *and the weights* w_1, w_2 *are nontrivial (they lie strictly between 0 and 1). Then the density from probability averaging satisfies* $f(v) \asymp f_1(v)$. *Assuming further that* f_1 *is log-concave, the density from quantile averaging satisfies* $\bar{f}(v) = o(f_1(v))$.

The proof is based on (14), and is deferred to Appendix A.4. The assumption that f_1 is log-concave for the quantile averaging result is stronger than it needs to be (as is the restriction to $p = 2$), but is used to simplify exposition. Proposition 5 reiterates the importance of allowing for level-dependent weights in a linear combination of quantiles. For applications in which there is considerable uncertainty about extreme events (especially ones in which there is disagreement in the degree of uncertainty between individual base models), we would not want an ensemble to de facto inherit a particular tail behavior—whether thin or thick—but want to endow the aggregation procedure with the ability to adapt its tail behavior as needed.

3 Aggregation methods

In what follows, we describe various aggregation strategies for combining multiple conditional quantile base models into a single model. Properly trained, the ensemble should be able to account for the strengths and

weaknesses of the base models and hence achieve superior accuracy to any one of them. This of course will only be possible if there is enough data to statistically identify such weaknesses and enough flexibility in the aggregation model to adjust for them.

3.1 Out-of-sample base predictions

To avoid overfitting, a standard model ensembling scheme uses *out-of-fold predictions* from all base models when learning ensemble weights; see, e.g., Van der Laan et al. (2007); Erickson et al. (2020). As in cross-validation, here we randomly partition the training data $\{(X_i, Y_i)\}_{i=1}^n$ into K disjoint and equally-sized folds (all of our experiments use $K = 5$), where the folds $\{\mathcal{I}_k\}_{k=1}^K$ form a partition of the index set $\{1, \dots, n\}$. For each fold $\mathcal{I}_k$, we retrain each base model on the other folds $\{\mathcal{I}_1, \dots, \mathcal{I}_K\} \setminus \{\mathcal{I}_k\}$ to obtain an out-of-sample prediction at each X_i, $i \in \mathcal{I}_k$, denoted $\hat{g}_j^{-k(i)}(X_i)$ (where we use $k(i)$ for the index of the fold containing the i^{th} data point, and the superscript $-k(i)$ on the base model estimate indicates that the model is trained on folds excluding the i^{th} data point). When fitting the ensemble weights, we only consider quantile predictions from each model that are out-of-sample:

$$\hat{g}_w(X_i) = \sum_{j=1}^p w_j(X_i) \cdot \hat{g}_j^{-k(i)}(X_i), \quad i = 1, \dots, n. \tag{15}$$

Once the ensemble weights have been learned, we make predictions at any new test point $x \in \mathcal{X}$ via (1), where the base models $\{\hat{g}_j\}_{j=1}^p$ have been fit to the full training set $\{(X_i, Y_i)\}_{i=1}^n$.

3.2 Global aggregation weighting schemes

For now, we assume that each weight is constant with respect to $x \in \mathcal{X}$, i.e., $w_j(x) = w_j$ for all $j = 1, \dots, p$. This puts us in the category of *global* aggregation procedures; we will turn to *local* aggregation procedures in the next subsection. All strategies described below can be cast in the following general form. We fit the global aggregation weights w by solving the optimization problem:

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \frac{1}{n} \sum_{i=1}^n \sum_{\tau \in \mathcal{T}} \psi_\tau \bigg(Y_i - \sum_{j=1}^p w_j \cdot \hat{g}_j^{-k(i)}(X_i; \tau) \bigg) \\ \text{subject to} \quad & Aw = 1, \quad w \geq 0, \end{aligned} \tag{16}$$

Note that each base model prediction used in (16) is an out-of-fold prediction, as in (15). Moreover, A is a linear operator that encodes the unit-sum constraint on the weights (further details on the parametrization for each case is described below):

$$Aw = \begin{cases} \sum_{j=1}^p w_j & \text{coarse case} \\ \big\{ \sum_{j=1}^p w_j^\tau \big\}_{\tau \in \mathcal{T}} & \text{medium case} \\ \big\{ \sum_{j=1}^p \sum_{\nu \in \mathcal{T}} w_j^{\tau,\nu} \big\}_{\tau \in \mathcal{T}} & \text{fine case.} \end{cases}$$

Lastly, in the second line of (16), we use 1 to denote the vector of all 1s (of appropriate dimension), and the constraint $w \geq 0$ is to be interpreted elementwise.

Within this framework, we can consider various weighted ensembling strategies of increasing flexibility (akin to coffee grind sizes). These were covered briefly in the introduction, and we expand on them below.

- *Coarse.* To each base model $\hat{g}_j$, we allocate a weight w_j, shared over all quantile levels. With p base models, a coarse aggregator learns p weights, satisfying $\sum_{j=1}^p w_j = 1$. The product "$\cdot$" in (16) is just a scalar-vector product, so that ensemble prediction for level τ is

  $$\sum_{j=1}^p \big(w_j \cdot \hat{g}_j(x) \big)_\tau = \sum_{j=1}^p w_j \, \hat{g}_j(x; \tau).$$

  This appears to be the standard way to build weighted ensembles, including quantile ensembles, see, e.g., Nowotarski and Weron (2018); Browell et al. (2020); Zhang et al. (2020); Uniejewski and Weron

(2021). However, it has clear limitations in the quantile setting, as outlined in Section 2.3. To recap, coarsely-aggregated distributions will generally have thinner tails than the thickest of the base model tails (Proposition 5), and if all base models produce quantiles according to some common location-scale family, then the coarsely-aggregated distribution will remain in this family (Proposition 3). Medium and fine strategies, described next, are free of such restrictions.

- *Medium.* To each base model $\hat{g}_j$ and quantile level τ, we allocate a weight w_j^τ. With p base models and m quantile levels, a medium aggregator learns $p \times m$ weights, satisfying $\sum_{j=1}^p w_j^\tau = 1$ for each quantile level τ. The weight assigned to each base model is a vector $w_j \in \mathbb{R}^p$, and the product "$\cdot$" in (16) is the Hadamard (elementwise) product between vectors, so that ensemble prediction for level τ is

$$\sum_{j=1}^p \big(w_j \cdot \hat{g}_j(x)\big)_\tau = \sum_{j=1}^p w_j^\tau \, \hat{g}_j(x;\tau).$$

  This approach enables the ensemble to account for the fact that different base models may be better or worse at predicting certain quantile levels. It has been considered in quantile aggregation by, e.g., Nowotarski and Weron (2015); Maciejowska et al. (2016); Lima and Meng (2017); Tibshirani (2020).

- *Fine.* To each base model $\hat{g}_j$, output (ensemble) quantile level τ, and input (base model) quantile level ν, we allocate a weight $w_j^{\tau,\nu}$. Thus with p base models and m quantile levels, a fine aggregator learns $p \times m \times m$ weights, satisfying $\sum_{j=1}^p \sum_{\nu \in \mathcal{T}} w_j^{\tau,\nu} = 1$ for each quantile level τ. The weight assigned to each base model is a matrix $w_j \in \mathbb{R}^{m \times m}$, and "$\cdot$" in (16) is a matrix-vector product, so the ensemble prediction for level τ is

$$\sum_{j=1}^p \big(w_j \cdot \hat{g}_j(x)\big)_\tau = \sum_{j=1}^p \sum_{\nu \in \mathcal{T}} w_j^{\tau,\nu} \, \hat{g}_j(x;\nu).$$

  This strategy presumes that for a given quantile level τ, base model estimates for other quantile levels $\nu \neq \tau$ could also be useful for aggregation purposes. Note that this is particularly pertinent to a setting which some base models are poorly calibrated or produce unstable estimates for one particular quantile level. To the best of our knowledge, this type of aggregation is not common and has not been thoroughly studied in the literature.

As a way of comparing the flexibility offered by the three strategies, denote the ensemble prediction at x by $\hat{g}_w(x) = \sum_{j=1}^p \big(w_j \cdot \hat{g}_j(x)\big)_\tau$, and note that (where we abbreviate $\text{range}_{s \in S} \, a_s = [\min_{s \in S} a_s, \max_{s \in S} b_s]$):

$$\hat{g}_w(x;\tau) \in \begin{cases} \text{range}_{j=1,\ldots,p} \; \hat{g}_j(x;\tau) & \text{coarse and medium cases} \\ \text{range}_{j=1,\ldots,p,\, \nu \in \mathcal{T}} \; \hat{g}_j(x;\nu) & \text{fine case.} \end{cases}$$

In the medium and fine cases, and any element in the respective ranges above is achieveable (by varying the weights appropriately). In the coarse case, this is not true, and $\hat{g}_w$ is restricted by the shape of the individual quantile functions (recall Section 2.3).

Furthermore, it is not hard to see that problem (16) is equivalent to a linear program (LP), and can be solved with any standard convex optimization toolkit. While conceptually tidy, solving this LP formulation can be overly computationally intensive at scale. In this paper, rather than relying on LP formulations, we fit global aggregation weights by optimizing (16) via stochastic gradient descent (SGD) methods, which are much more scalable to large data sets through their use of subsampled mini-batches and backpropagation in deep learning toolkits that can leverage hardware accelerators (GPUs). Outside of computational efficiency, the use of SGD also provides implicit regularization to avoid overfitting (which we can further control using early stopping). To circumvent the simplex constraints in (16), we reparametrize the weights $\{w_j\}_{j=1}^p$ using a softmax layer applied to (unconstrained) parameters $\{\phi_j\}_{j=1}^p$, which for the coarse case we can write as:

$$w = \texttt{SoftMax}(\phi) = \left\{ \frac{e^{\phi_j}}{\sum_{\ell=1}^p e^{\phi_\ell}} \right\}_{j=1}^p.$$

The other cases (medium and fine) follow similarly.

Yet another advantage of using SGD and deep learning toolkits to solve (16) is that this extends more fluidly to the setting of local aggregation weights, which we described next.

3.3 Local aggregation via neural networks

To fit *local* aggregation weights that vary with $x \in \mathcal{X}$, we use a neural network approach. For concreteness, we describe the procedure in the context of fine aggregation weights, and the same idea applies to the other cases (coarse and medium) as well. We will refer to the local-fine approach, described below, as *deep quantile aggregation* or DQA.

To fit weight functions $w_j^{\tau,\nu} : \mathcal{X} \to \mathbb{R}$, we model these jointly over all base models $j \in \{1, 2, \ldots, p\}$ and all pairs of quantile levels $(\tau, \nu) \in \mathcal{T} \times \mathcal{T}$ using a single neural network G. All but the last layer of G forms a feature extractor $f_\theta : \mathcal{X} \to \mathbb{R}^d$ that maps input feature values $x \in \mathcal{X}$ to a hidden representation $h \in \mathbb{R}^d$. In our experiments, we take f_θ to be a standard feed-forward network parametrized by θ. The last layer of G is defined, for each output quantile level $\tau \in \mathcal{T}$, by a matrix $W^\tau \in \mathbb{R}^{pm \times d}$ that maps the hidden representation h, followed by a softmax operation (to ensure the simplex constraints are met), to fine aggregation weights $w^\tau(x) \in \mathbb{R}^{pm}$, as in:

$$w^\tau(x) = G(x; \theta, W^\tau) = \texttt{SoftMax}\big(W^\tau f_\theta(x)\big), \quad \tau \in \mathcal{T}. \tag{17}$$

The parameters θ and $\{W^\tau\}_{\tau \in \mathcal{T}}$ can be jointly optimized via SGD, applied to the optimization problem:

$$\underset{\theta,\, \{W^\tau\}_{\tau \in \mathcal{T}}}{\text{minimize}} \; \frac{1}{n} \sum_{i=1}^n \sum_{\tau \in \mathcal{T}} \psi_\tau \bigg(Y_i - \sum_{j=1}^p \sum_{\nu \in \mathcal{T}} G(x; \theta, W^\tau)_{j,\nu} \, \hat{g}_j^{-k(i)}(X_i; \nu) \bigg). \tag{18}$$

Note that there are no explicit constraints because the softmax parametrization in (17) implicitly fulfills the appropriate constraints.

Why would we want to move from using global aggregation weights (16), which do not depend on the feature values x, to using local aggregation weights (17), (18), which do? Aside from the general perspective of increasing model flexibility, local weights can be motivated by the observation that, in any given problem setting, it may well be the case that that certain constituent models perform better (output more accurate quantile estimates) in some parts of the feature space, while others perform better in other parts. In such cases, a local aggregation scheme will be able to adapt its weights accordingly, whereas a global scheme will not, and will be stuck with attributing some global level of preference between the models.

3.4 Interpreting the local aggregation scheme

We can interpret our proposal in (17), (18) as a *mixture of experts* ensemble (Jacobs et al., 1991) where the gating network G emits predictions about which "experts" will be most accurate for any particular x. In the local-fine setting (DQA), the "experts" correspond to estimates of individual quantiles from individual base models, and we define a separate gating scheme for each target quantile level τ.

We can also view our local-fine aggregator through the lens of an *attention* mechanism (Bahdanau et al., 2015), which adaptively attends to the different quantile estimates from each base model, and uses a different attention map for each τ and each x. In terms of the architecture design, we use a representation of the tuple (τ, x) both as a *key* and as a *query*. The individual quantiles $\hat{g}_j(x; \tau)$ are then used as *values* in the (query, key, value) mechanism commonly used.

Visualizing these attention maps—see Figures 3 and 4—can help address natural questions, e.g., for any particular x and τ, which base estimates are most useful? Or, how do estimates at different quantile levels interact with each other to achieve accuracy in the ensemble? While this conveys only *qualitative* information, we note that interpretations such as these would be far more difficult to obtain for nonlinear aggregators like stacked ensembles (Wolpert, 1992).

3.5 New base model: deep quantile regression

Finally, we remark that the optimization in (18) can be used to itself define a standalone neural network quantile regressor, which we refer to as *deep quantile regression* or DQR. This is given by taking $p = 1$ in (18) with a trivial base model that outputs $\hat{g}_1(x; \tau) = 1$, for any x and τ. DQR can be useful as a base model (to feed into an aggregation method like DQA), and will be used as a point of comparison and/or illustration at various points in what follows. For example, Figure 5 illustrates DQR on a real data set.

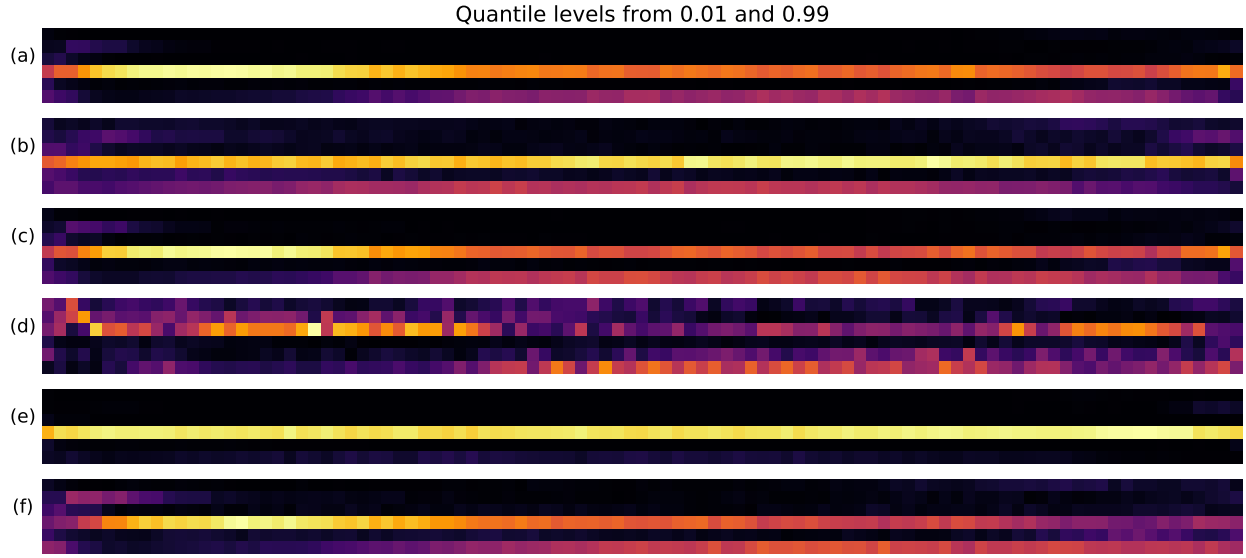

Figure 3: *Heatmaps of aggregation weights fit by the local-medium aggregator to the concrete data set, at 6 different input feature values. Each heatmap in (a)–(f) corresponds to a particular input feature value x. Its rows correspond to the $p = 6$ base models, and columns to the $m = 99$ quantile levels. A black color means that a given base prediction is essentially ignored by the aggregator. We can see that for different x_i the algorithm chooses different base models but also that it then mostly uses the estimates of these base models in a consistent fashion across quantile levels.*

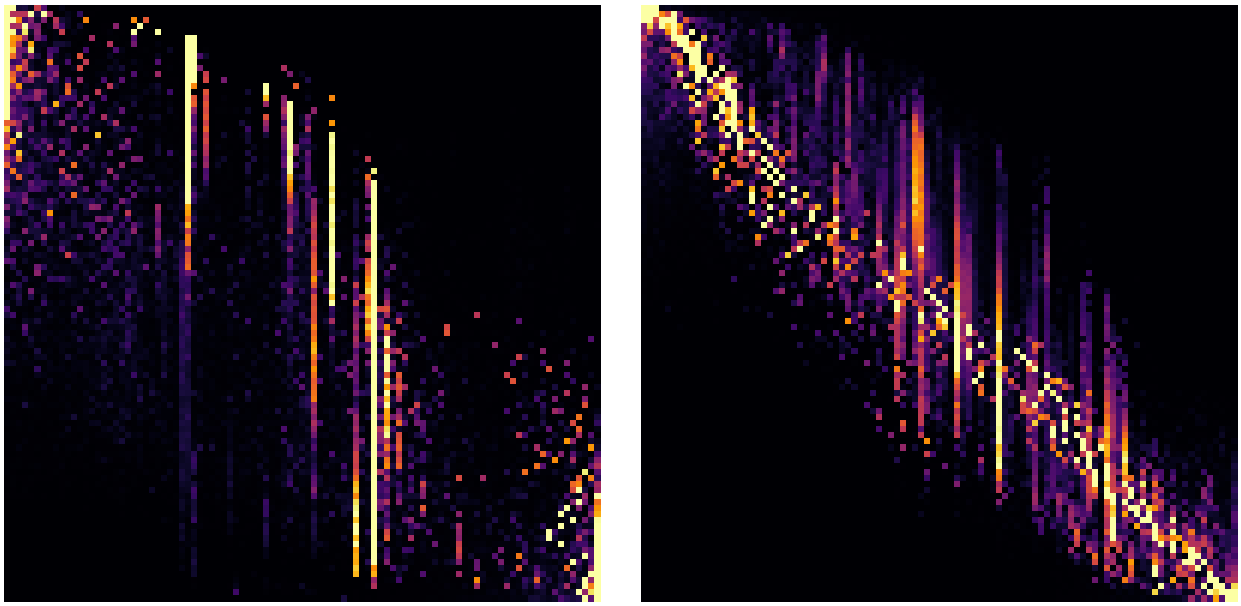

Figure 4: *Heatmaps of aggregation weights fit by the local-fine (DQA) aggregator to the concrete data set (left) and the power data set (right). In the current fine setting, we have one $m \times m = 99 \times 99$ weight matrix per base model; the heatmaps have therefore been averaged over the $p = 6$ base models. Their rows correspond to the output quantiles (for the aggregator), and the columns to the input quantiles (from the base model). The right heatmap displays clear concentration around the diagonal: here each input quantile is on average the most reliable estimate used to define its corresponding output quantile. This is much less true on the left, which also has more pronounced vertical "streaks": output quantiles use input quantiles from many neighboring quantile levels. The left heatmap also has notably more dispersion in the tails.*

4 Quantile noncrossing

Here we discuss methods for enforcing quantile noncrossing in the predictions produced by the aggregation methods from the last section. We combine and extend the ideas introduced in Section 2.2.

4.1 Crossing penalty and quantile buffering

As an alternative to enforcing hard noncrossing constraints during training, as in (10), we consider a crossing penalty of the form:

$$\rho(g) = \sum_{x \in \mathcal{X}_0} \sum_{\tau < \tau'} \big(g(x;\tau) - g(x;\tau') + \delta_{\tau,\tau'} \big)_+, \tag{19}$$

where $a_+ = \max\{a, 0\}$, and $\delta_{\tau,\tau'} \geq 0$, for $\tau < \tau'$ are constants (not depending on x) that encourage a buffer between pairs of quantiles at different quantile levels. In the simplest case, we can reduce this to just a single margin $\delta_{\tau,\tau'} = \delta$ across all quantile levels, that we can tune as a hyperparameter (e.g., using a validation set). We will cover a slightly more advanced strategy shortly, for fitting adaptive margins which vary with quantile levels (as well as the training data at hand).

The advantage of using a penalty (19) over constraints is primarily computational: we can add it to the criterion defining the aggregation model, and we can still use SGD for optimization. Note that the penalty is applied at the ensemble level, to $g = \hat{g}_w = \sum_{j=1}^p w_j \cdot \hat{g}_j$; to be precise, the global optimization (16) becomes

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \frac{1}{n} \sum_{i=1}^n \sum_{\tau \in \mathcal{T}} \psi_\tau \bigg(Y_i - \sum_{j=1}^p w_j \cdot \hat{g}_j^{-k(i)}(X_i;\tau) \bigg) + \lambda \, \rho \bigg(\sum_{j=1}^p w_j \cdot \hat{g}_j \bigg) \\ \text{subject to} \quad & Aw = 1, \quad w \geq 0, \end{aligned} \tag{20}$$

where $\lambda \geq 0$ is a hyperparameter to be tuned. The modification of the local optimization (18) to include the penalty term is similar. Figure 5 gives an illustration of the crossing penalty in action.

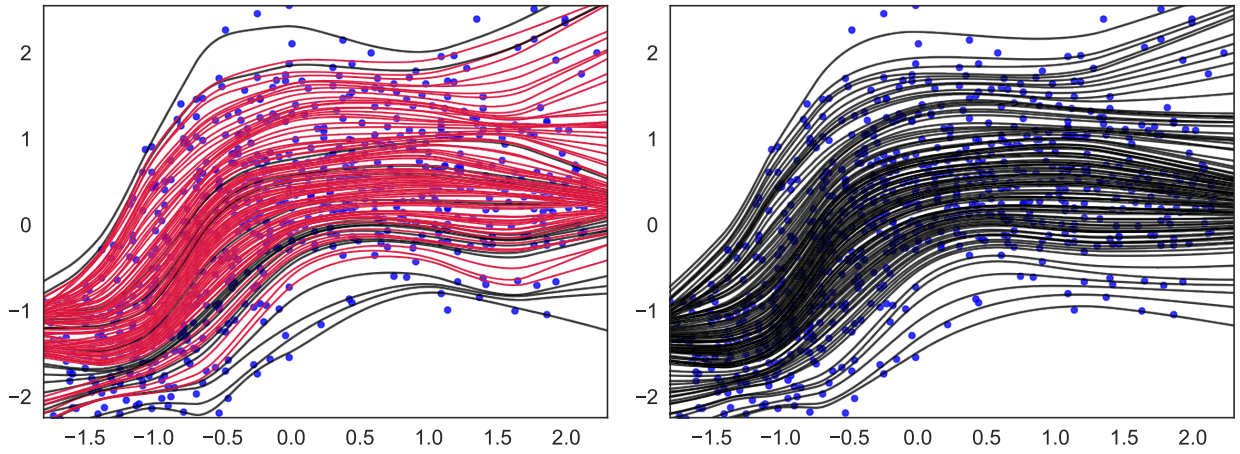

Figure 5: *Conditional quantile estimates fit by a neural network quantile regressor (DQR) to the bone mineral data set, used in Takeuchi et al. (2006), where X is univariate. The left panel displays estimates that were fit without any crossing penalty or isotonization operator, and the right panel displays estimates fit using the crossing penalty (19) and post sorting. (This is the simplest combination among all the methods that we consider to address quantile crossing.) Crossing quantile curves are highlighted in red.*

Adaptive margins. One of the problems with choosing a fixed margin δ (of using the reduction $\delta_{\tau,\tau'} = \delta$ for all τ, τ') is that it may not accurately reflect the scale or shape of the quantile gaps in the data at hand. We propose the following heuristic for tuning margin parameters $\delta_{\tau,\tau'}$, over all pairs of distinct levels $\tau < \tau'$. We begin with any pilot estimator of the conditional mean or median of $Y|X$, call it $\hat{g}_0$, and form residuals

$R_i = Y_i - \hat{g}_0(X_i)$, $i = 1, \ldots, n$. The closer these residuals are to the true error distribution, of $Y - \mathbb{E}(Y|X)$ (or similar for the median), the better. Thus we typically want to use cross-validation residuals for this pilot step. We then take, for each $\tau < \tau'$, the margin to be

$$\delta_{\tau,\tau'} = \delta_0 \big(Q_{\tau'}(\{R_i\}_{i=1}^n) - Q_\tau(\{R_i\}_{i=1}^n) \big)_+. \tag{21}$$

where Q_τ gives the empirical level τ quantile of its argument. The motivation behind the proposal in (21) is that $Q_{\tau'}(\{R_i\}_{i=1}^n) - Q_\tau(\{R_i\}_{i=1}^n)$ is itself the gap between quantile estimates from a basic but properly (monotonically) ordered quantile regressor, namely, that defined by:

$$\hat{g}_0(x;\tau) = \hat{g}_0(x) + Q_\tau(\{R_i\}_{i=1}^n), \quad x \in \mathcal{X},\ \tau \in \mathcal{T}.$$

We note that $\delta_0 \geq 0$ in (21) is a single hyperparameter to be tuned. In our experiments, we take the pilot estimator to be the simple average of base model predictions at the median, $\hat{g}_0(x) = \frac{1}{p}\sum_{j=1}^p \hat{g}_j(x; 0.5)$, and we use the same cross-validation scheme for the base models as in Section 3.1 to produce out-of-fold residuals R_i, $i = 1, \ldots, n$ around $\hat{g}_0$, that are used in (21).

4.2 Isotonization: post hoc and end-to-end

Next we consider different isotonization operators that can be used in conjunction with the crossing penalty in (19). This is important because it will guarantee quantile noncrossing for the ensemble predictions at all points $x \in \mathcal{X}$, as in (9), whereas the crossing penalty alone is not sufficient to ensure this. The added benefit of this approach is that we are able to satisfy the constraint at out-of-sample points without having to invest unreasonable amounts of function complexity into the model, simply by applying isotonization as a form of prior knowledge (see Le et al. 2006 for a detailed general discussion of this aspect).

Generally, an isotonization operator $\mathcal{S}$ can be used in one of two ways. The first is *post hoc*, as explained in Section 2.2, where we simply apply it to quantile predictions after the ensemble $\hat{g} = \sum_{j=1}^p \hat{w}_j \cdot \hat{g}_j$ has been trained, as in (11). The second way is *end-to-end*, where $\mathcal{S}$ gets used a layer in training, and we now solve, instead of (20),

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \frac{1}{n}\sum_{i=1}^n \sum_{\tau\in\mathcal{T}} \psi_\tau \bigg(Y_i - \Big[\mathcal{S}\Big(\sum_{j=1}^p w_j \cdot \hat{g}_j^{-k(i)}(X_i) \Big) \Big]_\tau \bigg) + \lambda\, \rho\Big(\sum_{j=1}^p w_j \cdot \hat{g}_j \Big) \\ \text{subject to} \quad & Aw = 1, \quad w \geq 0, \end{aligned} \tag{22}$$

where we denote by $[\mathcal{S}(v)]_\tau$ the element of $\mathcal{S}(v)$ corresponding to a quantile level τ. The modification for the local aggregation problem is similar. For predictions, we still post-apply $\mathcal{S}$, as in (11).

One might imagine that the end-to-end approach can help predictive accuracy, since in training we are leveraging the knowledge of what we will be doing at prediction time (applying $\mathcal{S}$). Furthermore, by treating $\mathcal{S}$ as a differentiable layer, we can still seamlessly apply SGD for optimization (end-to-end training). Note that sorting (12) and isotonic projection (13) are almost everywhere differentiable (they are in fact almost everywhere linear maps). We will propose an additional isotonization operator shortly that is also almost everywhere differentiable, and particularly convenient to implement in modern deep learning toolkits.

To be clear, the end-to-end approach is not free of downsides; while sorting and isotonic projection are guaranteed to improve WIS when applied post hoc (Proposition 2), the same cannot be said when they are used end-to-end. Thus we give up on a formal guarantee, to potentially gain accuracy in practice.

Min-max sweep. In addition to sorting (12) and isotonic projection (13), we propose and investigate a simple isotonization operator that we call the *min-max sweep*. It works as follows: starting at the median, it makes two outward sweeps: one upward and one downward, where it replaces each value with a cumulative max (upward sweep) or cumulative min (downward sweep). To be more precise, if we have sorted quantile levels $\tau_1 < \cdots < \tau_m$ with $\tau_{m_0} = 0.5$, and corresponding values $v_1, \ldots, v_m$, then we define the *min-max sweep* operator according to:

$$\texttt{MinMaxSweep}(v)_k = \begin{cases} v_k & k = m_0 \\ \max\{v_k, \texttt{MinMaxSweep}(v)_{k-1}\} & k = m_0+1, \ldots, m \\ \min\{v_k, \texttt{MinMaxSweep}(v)_{k+1}\} & k = m_0-1, \ldots, 1. \end{cases} \tag{23}$$

Like sorting and isotonic projection, the min-max sweep is almost everywhere a linear map, and thus almost everywhere differentiable. The primary motivation for it is that it can be implemented highly efficiently in modern deep learning frameworks via cumulative max and cumulative min functions—these are vectorized operations that can be efficiently applied in parallel over a mini-batch, in each iteration of SGD (when it is used in an end-to-end manner). Unlike sorting and isotonic projection, however, it is not guaranteed to improve WIS (Proposition 2) when applied in post.

5 Conformal calibration

Informally, a conditional quantile estimator is said to be *calibrated* provided that its quantile estimates have the appropriated nominal one-sided coverage, for example, an estimated 90% quantile covers the target from above 90% of the time, and the same for the other quantile levels being estimated. Equivalently, we can view this in terms of central prediction intervals: the interval whose endpoints are given by the estimated 5% and 95% quantiles covers the target 90% of the time. Formally, let $\hat{g}$ be an estimator trained on samples (X_i, Y_i), $i = 1, \ldots, n$, with $\hat{g}(x; \tau)$ denoting the estimated level τ quantile of $Y|X = x$. Assume that the set of quantile levels $\mathcal{T}$ being estimated is symmetric around 0.5, so we can write it as $\mathcal{T} = \cup_{\alpha \in \mathcal{A}} \{\alpha/2, 1 - \alpha/2\}$. Then $\hat{g}$ is said to be calibrated, provided that for every $\alpha \in \mathcal{A}$, we have

$$\mathbb{P}\Big(Y_{n+1} \in \Big[\hat{g}(X_{n+1}; \alpha/2),\, \hat{g}(X_{n+1}; 1-\alpha/2)\Big]\Big) = 1 - \alpha, \tag{24}$$

where (X_{n+1}, Y_{n+1}) is a test sample, assumed to be i.i.d. with the training samples (X_i, Y_i), $i = 1, \ldots, n$. To be precise, the notion of calibration that we consider in (24) is *marginal* over everything: the training set and the test sample. We remark that a stronger notion of calibration, which may often be desirable in practice, is calibration *conditional* on the test feature value:

$$\mathbb{P}\Big(Y_{n+1} \in \Big[\hat{g}(X_{n+1}; \alpha/2),\, \hat{g}(X_{n+1}; 1-\alpha/2)\Big] \,\Big|\, X_{n+1} = x\Big) = 1 - \alpha, \quad x \in \mathcal{X}. \tag{25}$$

Conditional calibration, as in (25), is actually impossible to achieve in a distribution-free sense; see Lei and Wasserman (2014); Vovk (2012); Barber et al. (2021a) for precise statements and developments. Meanwhile, marginal calibration (24) is possible to achieve using *conformal prediction* methodology, without assuming anything about the joint distribution of (X, Y). The definitive reference on conformal prediction is the book by Vovk et al. (2005); see also Lei et al. (2018), which helped popularize conformal methods in the statistics and machine learning communities. The conformal literature is by now somewhat vast, but we will only need to discuss a few parts of it that are most relevant to calibration of conditional quantile estimators.

In particular, we very briefly review *conformalized quantile regression* or CQR by Romano et al. (2019) and CV+ by Barber et al. (2021b). We then discuss how these can be efficiently applied in combination to any of the quantile aggregators studied in this paper.

CQR. We first describe CQR in a split-sample setting. Suppose that we have reserved a subset indexed by $\mathcal{I}_1 \subseteq \{1, \ldots, n\}$ as the *proper training set*, and $\mathcal{I}_2 = \{1, \ldots, n\} \setminus \mathcal{I}_1$ as the *calibration set*. (Extending this to a cross-validation setting, via the CV+ method, is discussed next.) In this split-sample setting, CQR from Romano et al. (2019) can be explained as follows.

1. First, fit the quantile estimator $\hat{g}^{\mathcal{I}_1}$ on the proper training set $\{(X_i, Y_i) : i \in \mathcal{I}_1\}$.
2. Then, for each $\alpha \in \mathcal{A}$, compute lower and upper residuals on the calibration set,
$$R^-_{i,\alpha} = \hat{g}^{\mathcal{I}_1}(X_i; \alpha/2) - Y_i \quad \text{and} \quad R^+_{i,\alpha} = Y_i - \hat{g}^{\mathcal{I}_1}(X_i; 1-\alpha/2), \quad i \in \mathcal{I}_2.$$
3. Finally, adjust (or *conformalize*) the original estimates based on residual quantiles, yielding for $\alpha \in \mathcal{A}$ and $x \in \mathcal{X}$,
$$\begin{aligned} \tilde{g}(x; \alpha/2) &= \hat{g}^{\mathcal{I}_1}(x; \alpha/2) - \tilde{Q}_{1-\alpha/2}\big(\{R^-_{i,\alpha} : i \in \mathcal{I}_2\}\big), \\ \tilde{g}(x; 1-\alpha/2) &= \hat{g}^{\mathcal{I}_1}(x; 1-\alpha/2) + \tilde{Q}_{1-\alpha/2}\big(\{R^+_{i,\alpha} : i \in \mathcal{I}_2\}\big), \end{aligned} \tag{26}$$
where $\tilde{Q}_\tau$ is a slightly modified empirical quantile function, giving the empirical level $\lceil \tau(n_2 + 1) \rceil / n_2$ quantile of its argument, with $n_2 = |\mathcal{I}_2|$.

A cornerstone result in conformal prediction theory (see Theorems 1 and 2 in Romano et al. (2019) for the application of this result to CQR) says that for any estimator, its conformalized version in (26) has the finite-sample marginal coverage guarantee

$$\mathbb{P}\Big(Y_{n+1} \in \Big[\tilde{g}(X_{n+1}; \alpha/2),\, \tilde{g}(X_{n+1}; 1-\alpha/2)\Big]\Big) \geq 1-\alpha. \tag{27}$$

In fact, the coverage of the level $1-\alpha$ central prediction interval is also upper bounded by $1-\alpha+1/(n_2+1)$, provided that the residuals are continuously distributed (i.e., there are almost surely no ties).

CV+. Now we describe how to extend CQR to a cross-validation setting. Let $\{I_k\}_{k=1}^K$ denote a partition of $\{1,\dots,n\}$ into disjoint folds. We can map steps 1–3 used to produce the CQR correction in (26) onto the CV+ framework of Barber et al. (2021b), as follows.

1. First, for each fold $k = 1,\dots,K$, fit the quantile estimator $\hat{g}^{-k}$ on all data points but those in the k^{th} fold, $\{(X_i, Y_i) : i \notin \mathcal{I}_k\}$.

2. Then, for each $i = 1,\dots,n$ and each $\alpha \in \mathcal{A}$, compute lower and upper residuals on the calibration fold $k(i)$ (where $k(i)$ denotes the index of the fold containing the i^{th} data point),

$$R^-_{i,\alpha} = \hat{g}^{-k(i)}(X_i; \alpha/2) - Y_i \quad \text{and} \quad R^+_{i,\alpha} = Y_i - \hat{g}^{-k(i)}(X_i; 1-\alpha/2), \quad i \in \mathcal{I}_{k(i)}.$$

3. Finally, adjust (or conformalize) the original estimates based on residual quantiles, yielding for $\alpha \in \mathcal{A}$ and $x \in \mathcal{X}$,

$$\begin{aligned} \tilde{g}(x; \alpha/2) &= \tilde{Q}^-_{1-\alpha/2}\Big(\big\{\hat{g}^{-k(i)}(x; \alpha/2) - R^-_{i,\alpha} : i \in k(i)\big\}\Big), \\ \tilde{g}(x; 1-\alpha/2) &= \tilde{Q}^+_{1-\alpha/2}\Big(\big\{\hat{g}^{-k(i)}(x; 1-\alpha/2) + R^+_{i,\alpha} : i \in k(i)\big\}\Big), \end{aligned} \tag{28}$$

where now $\tilde{Q}^-_\tau, \tilde{Q}^+_\tau$ are modified lower and upper level τ empirical quantiles: for any set Z, we define $\tilde{Q}^+_\tau(Z)$ to be the level $\lceil \tau(|Z|+1) \rceil / |Z|$ empirical quantile of Z, and $\tilde{Q}^-_\tau(Z) = -\tilde{Q}^+(-Z)$.

According to results from Barber et al. (2021b) and Vovk et al. (2018) (see Theorem 4 in Barber et al. (2021b) for a summary), for any estimator, its conformalized version in (28) has the finite-sample marginal coverage guarantee

$$\begin{aligned} \mathbb{P}\Big(Y_{n+1} \in \Big[\tilde{g}(X_{n+1}; \alpha/2),\, \tilde{g}(X_{n+1}; 1-\alpha/2)\Big]\Big) &\geq 1 - 2\alpha - \min\left\{\frac{2(1-1/K)}{n/K+1},\, \frac{1-K/n}{K+1}\right\} \\ &\geq 1 - 2\alpha - \sqrt{2/n}. \end{aligned} \tag{29}$$

We can see that the guarantee from CV+ in (29) is weaker than that in (27) from sample-splitting, though CV+ has the advantage that it utilizes each data point for *both* training and calibration purposes, and can often deliver shorter conformalized prediction intervals as a result. Moreover, Barber et al. (2021b) argue that for stable estimation procedures, the empirical coverage from CV+ is closer to $1-\alpha$ (and provide some theory to support this as well).

Nested implementation for quantile aggregators. We discuss the application of CQR and CV+ to calibrate the quantile aggregation models developed in Section 3 and 4. An additional level of complexity in this application arises because the quantile aggregation model is itself built using cross-validation: recall, as discussed in Section 3.1, that the aggregation weights are trained using out-of-sample (out-of-fold) predictions from the base models. Thus, application of CV+ here requires a *nested* cross-validation scheme, where in the inner loop, the aggregator is trained using cross-validation (for the base model predictions), and in the outer loop, the calibration residuals are computed for the ultimate conformalization step.

In order to make this as computationally efficient as possible, we introduce a particular nesting scheme that reduces the number of times base models are trained by roughly a factor of 2. We first pick a number of folds K for the outer loop, and define folds $\mathcal{I}_k$, $k = 1,\dots,K$. Then for each $k = 1,\dots,K$, we fit the quantile aggregation model $\hat{g}^{-k}_w$ on $\{(X_i, Y_i) : i \notin \mathcal{I}_k\}$ (using any one of the approaches described in Sections 3 and 4),

with the key idea being how we implement the inner cross-validation loop to train the base models: we use folds $\mathcal{I}_\ell$, $\ell \neq k$ for this inner loop, so that we can later avoid having to refit the base models when the roles of k and ℓ are reversed. To see more clearly why this is the case, we describe the procedure in more detail below.

1. For $k = 1, \ldots, K$:
   a. For $\ell \neq k$, train each base model $\hat{g}_j^{-k,\ell}$ on $\{(X_i, Y_i) : i \notin \mathcal{I}_k \cup \mathcal{I}_\ell\}$.
   b. Train the aggregation weights w on $\{(X_i, Y_i) : i \notin \mathcal{I}_k\}$ (using the base model predictions $\hat{g}_j^{-k,\ell}(X_i)$ from Step a), to produce the aggregator $\hat{g}_w^{-k}$.
   c. Compute lower and upper calibration residuals of $\hat{g}_w^{-k}$ on $\{(X_i, Y_i) : i \in \mathcal{I}_k\}$, as in Step 2 of CV+.
2. Conformalize original estimates using residual quantiles, as in Step 3 of CV+.

The computational savings comes from observing that $\hat{g}_j^{-k,\ell} = \hat{g}_j^{-\ell,k}$: the base model we train when k is the calibration fold and ℓ is the validation fold is the same as that we would train when the roles of k and ℓ are reversed. Therefore we only need to train each base model in Step 1a, across all iterations of the outer and inner loops, a total of $K(K-1)/2$ times, as opposed to $K(K-1)$ times if we were ignorant to this fact (or K^2 times, the result of a more typical nested K-fold cross-validation scheme).

6 Empirical comparisons

We provide a broad empirical comparison of our proposed quantile aggregation procedures as well as many different established regression and aggregation methods. We examine 34 data sets in total: 8 from the UCI Machine Learning Repository (Dua and Graff, 2017) and 26 from the AutoML Benchmark for Regression from the Open ML Repository (Vanschoren et al., 2013). These have served as popular benchmark repositories in probabilistic deep learning and uncertainty estimation, see, e.g., Lakshminarayanan et al. (2017); Mukhoti et al. (2018); Jain et al. (2020); Fakoor et al. (2020). To the best of our knowledge, what follows is the most comprehensive evaluation of quantile regression/aggregation methods published to date.

Training, validation, and testing setup. For each of the 34 data sets studied, we average all results over 5 random train-validation-test splits, of relative size 72% (train), 18% (validation), and 10% (test). We adopt the following workflow for each data set.

1. Standardize the feature variables and response variable on the combined training and validation sets, to have zero mean and unit standard deviation.
2. Fit the base models on the training set.
3. Find the best hyperparameter configuration for each base model by minimizing weighted interval score (WIS) on the validation set. Fix these henceforth.
4. Divide the training set into 5 random folds. Record out-of-fold (OOF) predictions for each base model; that is, for fold k, we fit each base model $\hat{g}_j^{-k}$ by training on data from all folds except k, and use it to produce $\hat{g}_j^{-k}(X_i)$ for each i such that $k(i) = k$.
5. Fit the aggregation models on the training set, using the OOF base predictions from the last step.
6. Find the best hyperparameter configuration for each aggregation model by again minimizing validation WIS. Fix these henceforth.
7. Report the test WIS for each aggregation model (making sure to account for the initial standardization step, in the test set predictions).

All base and aggregation models are fit using $m = 99$ levels, $\mathcal{T} = \{0.01, 0.02, \ldots, 0.99\}$; the specific base and aggregation models we consider are described below. The initial standardization step is convenient for hyperparameter tuning. For more details on hyperparameter tuning, see Appendix B.1 and B.2.

Base models. We consider $p = 6$ quantile regression models as base models for the aggregators, described below, along with the abbreviated names we give them henceforth. The first three are neural network models with different architectures and objectives. More details are given in Appendix B.1.

1. Conditional Gaussian network (CGN, Lakshminarayanan et al., 2017).
2. Simultaneous quantile regression (SQR, Tagasovska and Lopez-Paz, 2019);
3. Deep quantile regression (DQR), which falls naturally out of our framework, as explained in Section 3.3.
4. Quantile random forest (RandomForest, Meinshausen, 2006).
5. Extremely randomized trees (ExtraTrees, Geurts et al., 2006).
6. Quantile gradient boosting (LightGBM, Ke et al., 2017).

Aggregation models. We study 4 aggregation models outside of the ones defined in our paper, described below, along with their abbreviated names. More details are given in Appendix B.2.

1. Average: a straight per-quantile average of base model predictions.
2. Median: a straight per-quantile median of base model predictions.
3. Quantile regression averaging (QRA, Nowotarski and Weron, 2015).
4. Factor QRA (FQRA, Maciejowska et al., 2016).

The rest of this section proceeds as follows. We first discuss the performance of the local-fine aggregator from our framework in Section 3, called DQA, to other methods (all base models, and the other aggregators defined outside of this paper). In the subsections that follow, we move on to a finer analysis, comparing the different weighted ensembling strategies to each other, comparing the different isotonization approaches to each other, and evaluating the gains in calibration made possible by conformal prediction methodology.

6.1 Continuation of Figure 1

Recall that in Figure 1, we already described some results from our empirical study, comparing DQA to all other methods. To be precise, for DQA—in Figure 1, here, and henceforth—we use the simplest strategy to ensure quantile noncrossing among the options discussed in Section 4: the crossing penalty along with post sorting (CrossPenalty + PostSort in the notation that we will introduce shortly). To give an aggregate view of the same results, Figure 6 shows the average relative WIS over all data sets. Relative WIS is the ratio of the WIS of given method to that of DQA, hence 1 indicates equal performance to DQA, and larger numbers indicate worse performance. (Standard error bars are also drawn.) These results clearly show the favorable average-case performance of DQA, to complement the per-data-set view in Figure 1. We can also see that QRA and FQRA are runners up in terms of average performance; while they are fairly simple aggregators, based on quantile linear regression, they use inputs here the predictions from flexible base models. Going back to the results in Figure 1, it is worth noting that there are data sets for which QRA and FQRA perform clearly worse than DQA (up to 50% worse, a relative factor of 1.5), but none where they perform better. In Appendix B.3, we provide an alternative visualization of these same results (differences in WIS rather than ratios of WIS).

6.2 Comparing ensembling strategies

Now we compare different ensembling strategies of varying flexibility, as discussed in Section 3. Recall that we have two axes: global or local weighting; and coarse, medium, or fine parameterization. This gives a total of 6 approaches, which we denote by global-coarse, global-medium, and so on. Figure 7 displays the relative WIS of each method to global-fine per data set (left panel), and the average relative WIS over all data sets (right panel). The data sets here and henceforth are indexed by increasing PVE, as in Figure 1. We note that global-fine is like a "middle man" in terms of its flexibility, among the 6 strategies considered. The overall trend in Figure 7 is that greater flexibility leads to better performance, especially for larger PVE values.

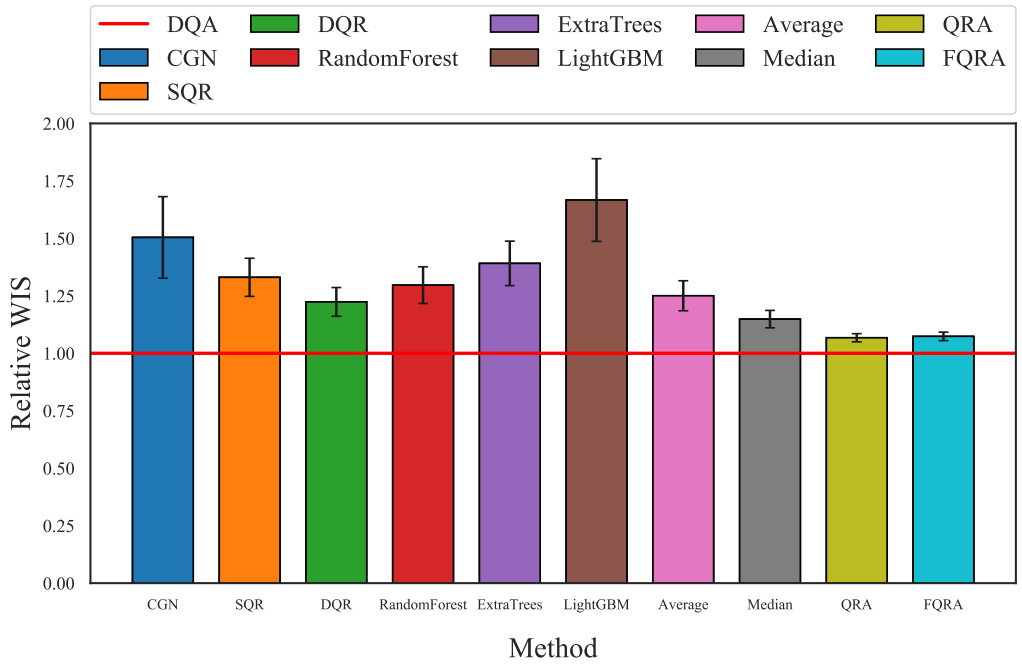

Figure 6: *Average relative WIS over all data sets for deep quantile aggregation (DQA) and various quantile regression methods. (The per-data-set results are given in Figure 1). Numbers larger than 1 indicate a worse average performance than DQA. We see that DQA performs the best overall, and QRA and FQRA are somewhat close runners up.*

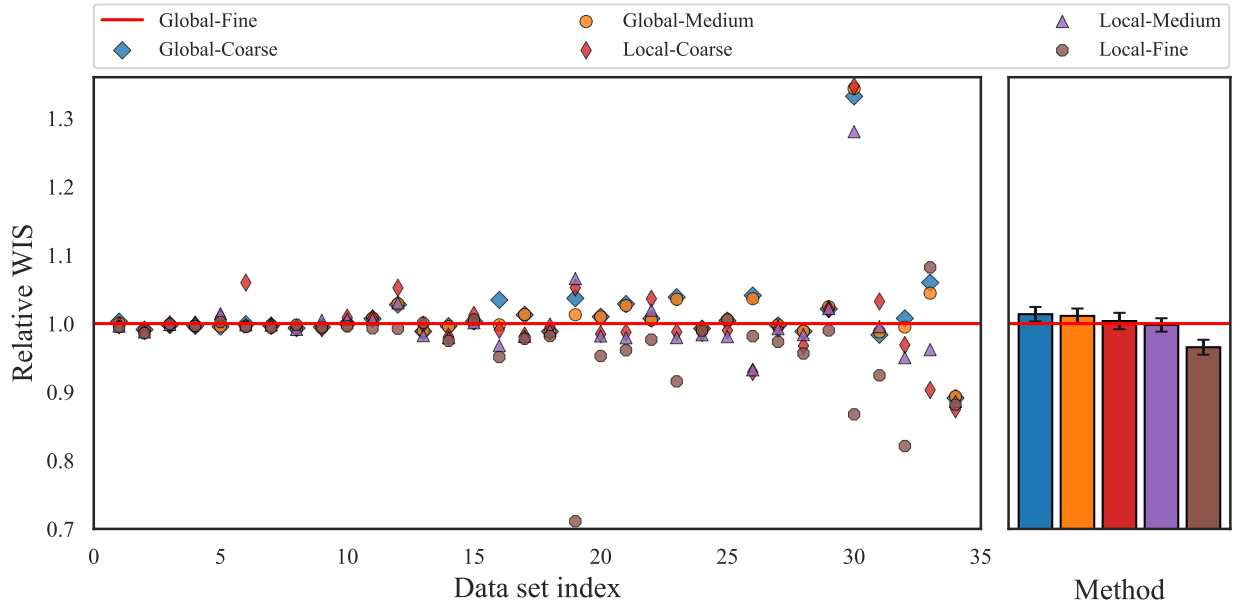

Figure 7: *Comparing the relative WIS of various ensembling strategies to global-fine. The left panel shows the relative WIS per data set (ordered by increasing PVE), and the right panel shows the average relative WIS over all data sets. In general, the more flexible aggregation strategies tend to perform better, and local-fine (DQA) performs the best overall.*

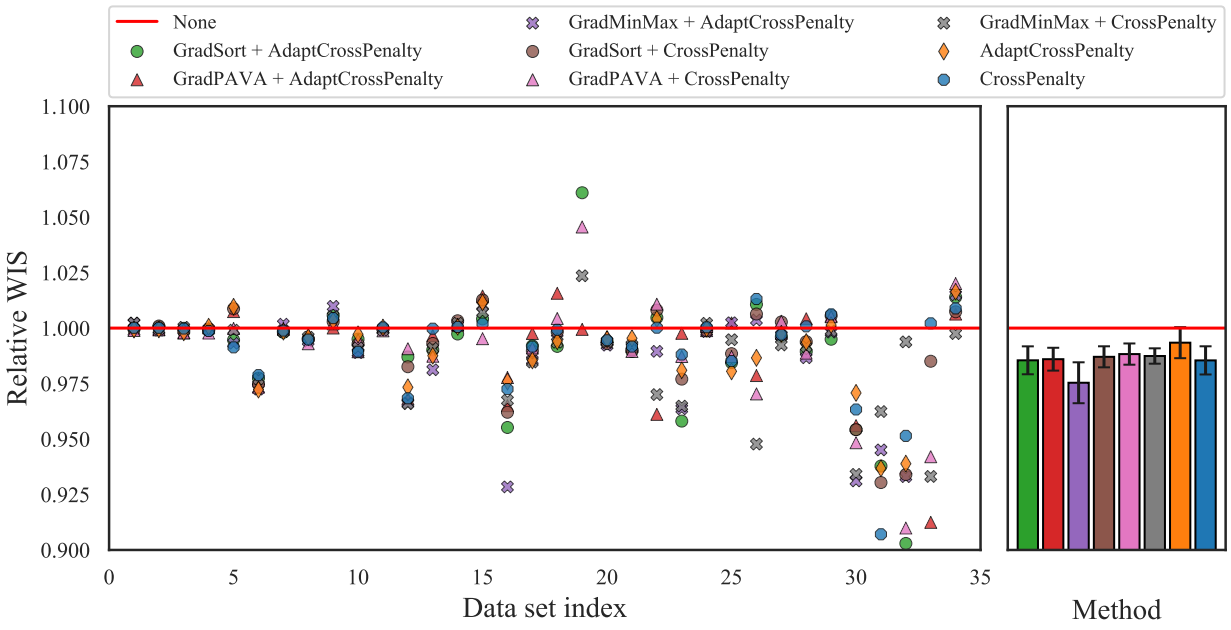

Figure 8: *Comparing the relative WIS of various isotonization strategies on top of pure DQA, with no penalty and no isotonization operator ("none"). The display is as in Figure 7 (individual results per data set on left panel, averaged results on right panel). We can see that all isotonization methods offer consistent but small improvements on "none", with AdaptCrossPenalty + GradMinMax achieving the best average-case performance.*

6.3 Comparing isotonization approaches

We compare the various isotonization approaches discussed in Section 4, applied to DQA. In Figure 8, the legend label "none" refers to DQA without any penalty and any isotonization operator; all WIS scores are computed relative to this method. CrossPenalty and AdaptCrossPenalty, respectively, denote the crossing penalty and adaptive cross penalty from Section 4.1. PostSort and GradSort, respectively, denote the sorting operator applied in post-processing and in training (see (22) for how this works in the global models), as described in Section 4.2. We similarly use PostPAVA, GradPAVA, and so on. Lastly, we use "+" to denote the combination of a penalty and isotonization operator, as in CrossPenalty + PostSort. The main takeaway from Figure 8 is that all of the considered methods improve upon "None". However, it should be noted that these are second-order improvements in relative WIS compare to the much larger first-order improvements in Figures 1 and 6, of DQA over base models and other aggregators—compare the y-axis range in Figure 8 to that in Figures 1 and 6.

A second takeaway from Figure 8 might be to generally recommend AdaptCrossPenalty + GradMinMax for future aggregation problems, which had the best average performance across our empirical suite. That said, given the small relative improvement this offers, we generally stick to using the simpler CrossPenalty + PostSort in the current paper.

Finally, Figure 9 gives an empirical verification of Proposition 2, which recall, showed that sorting and PAVA applied in post-processing can never make WIS worse. We can see that for each of the 34 data sets, combining PostSort or PostPAVA with CrossPenalty (left panel) or AdaptCrossPenalty (right panel) never hurts WIS, and occasionally improves it. This is *not* true for PostMinMax, as we can see that PostMinMax hurts WIS in a few data sets across the panels, the most noticeable instance being data set 19 on the left.

6.4 Evaluating conformal calibration

We examine the CQR-CV+ method from Section 5 applied to DQA, to adjust its central prediction intervals to have proper coverage. Figures 10 and 11 summarize empirical coverage and interval length at the nominal 0.8 coverage level, respectively. (The results for other nominal levels are similar.) Here, the coverage and

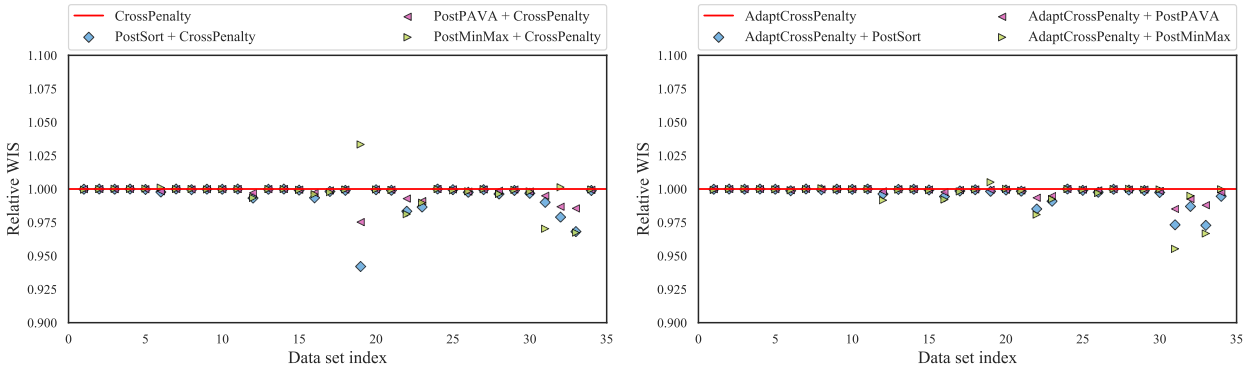

Figure 9: *Comparing post isotonization strategies applied on top of CrossPenalty on the left, and AdaptCrossPenalty on the right. As we can see in either panel, PostSort and PostPAVA can only improve WIS; verifying Proposition 2. This is not true of PostMinMax, the most noticeable example being data set 19 on the left panel.*

length of a quantile regressor $\hat{g}$ at level $1-\alpha$ are defined as

$$\frac{1}{m}\sum_{i=1}^{m} 1\Big\{\hat{g}(X_i^*;\alpha/2) \le Y_i^* \le \hat{g}(x_i;1-\alpha/2)\Big\},$$
$$\frac{1}{m}\sum_{i}^{m}\Big[\hat{g}(X_i^*;1-\alpha/2) - \hat{g}(X_i^*;\alpha/2)\Big],$$

respectively, where (X_i^*, Y_i^*), $i = 1,\dots,m$ denotes the test set. The figures show DQA (denoted CrossPenalty + PostSort), its conformalized version based on (28) (denoted CrossPenalty + PostSort + Conformalization), and the Average, Median, QRA, and FQRA aggregators.

We can see in Figure 10 that DQA achieves fairly close to the desired 0.8 coverage when this is averaged over all data sets, but it fails to achieve this *per data set*. Crucially, its conformalized version achieves at least 0.8 coverage on *every individual data set*. Generally, the Average and Median aggregators tend to overcover, but there are still data sets where they undercover. QRA and FQRA have fairly accurate average coverage over all data sets, but also fail to cover on several individual data sets.

As for length, we can see in Figure 11 that conformalizing DQA often inflates the intervals, but never by more than 75% (a relative factor of 1.75), and typically only in the 0-30% range. The Average and Median aggregators, particularly the former, tend to have longer intervals by comparison. QRA and FQRA tend to output slightly longer intervals than DQA, and slightly shorter intervals than conformalized DQA.

It is worth emphasizing that conformalized DQA is the only method among those discussed here that is completely robust to the calibration properties of the constituent quantile regression models (as is true of conformal methods in general). The coverage exhibited by the other aggregators will generally be a function of that of the base models, to varying degrees (e.g., the Median aggregator is more robust than the Average aggregator).

7 Discussion

We studied methods for aggregating any number of conditional quantile models with multiple quantile levels, introducing weighted ensemble schemes with varying degrees of flexibility, and allowing for weight functions that vary over the input feature space. Being based on neural network architectures, our quantile aggregators are easy and efficient to implement in deep learning toolkits. To impose proper noncrossing of estimated quantiles, we examined penalties and isotonization operators that can be used either post hoc or during training. To guarantee proper coverage of central prediction intervals, we gave an efficient nested application of the conformal prediction method CQR-CV+ within our aggregation framework.

Finally, we carried out a large empirical comparison of all of our proposals and others from the literature across 34 data sets from two popular benchmark repositories, the most comprehensive evaluation quantile

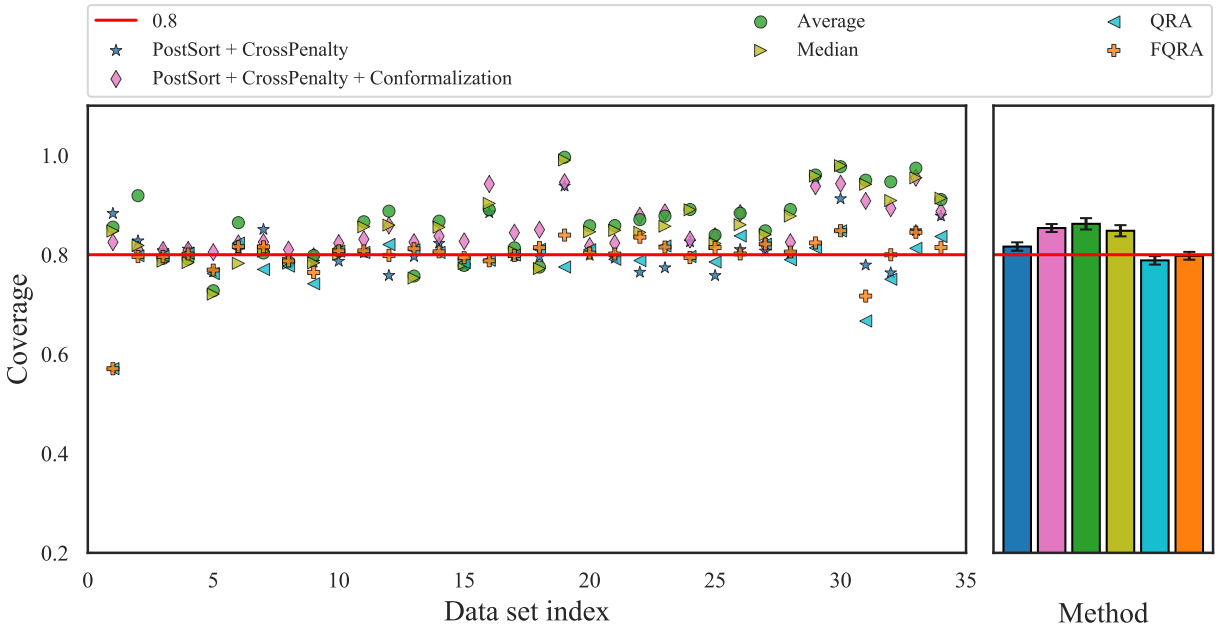

Figure 10: *Comparing coverage of DQA, conformalized DQA, and other aggregators at the nominal 0.8 level. The display is as in Figure 7 (individual results per data set on left panel, averaged results on right panel). The key point is that conformalized DQA manages to achieve valid coverage on each individual data set.*

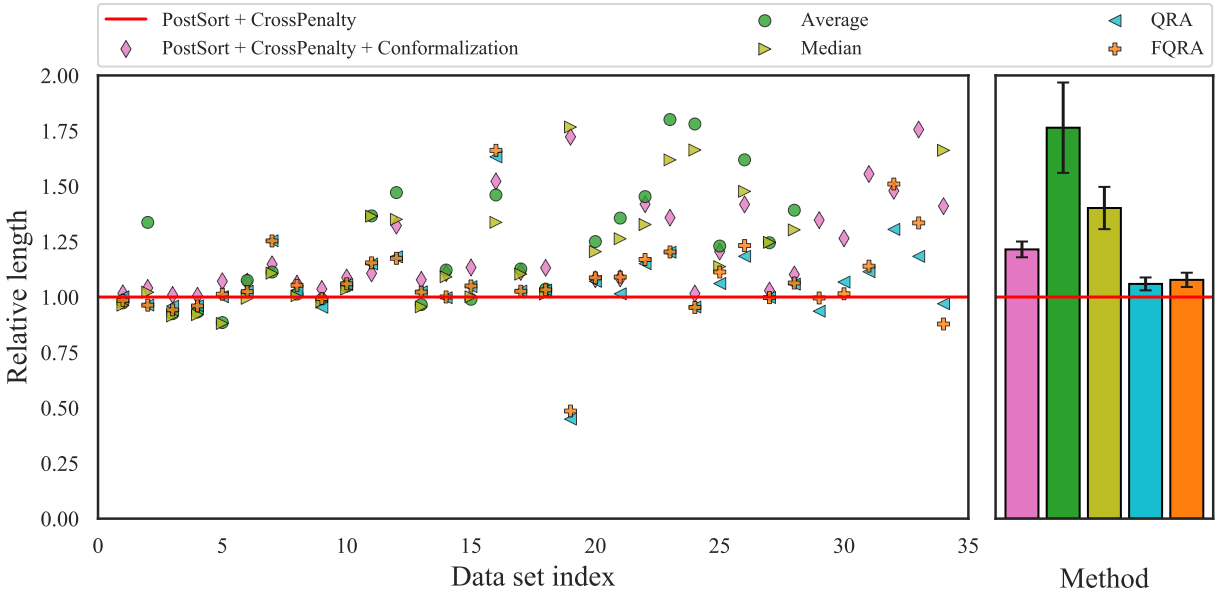

Figure 11: *Comparing prediction interval lengths from DQA, conformalized DQA, and others at the nominal 0.8 level. The display is again as in Figure 7 (individual results on the left, averaged results on the right). Conformalized DQA inflates interval lengths compared to DQA (in order to achieve valid coverage), but not by a huge amount, making it still clearly favorable to the Average and Median aggregators, and comparable to QRA.*

regression/aggregation methods that we know of. The conclusion is roughly that the most flexible model in our aggregation family, which we call DQA (deep quantile aggregation), combined with the simplest penalty and simplest post processor (just post sorting the quantile predictions), and conformalized using CQR-CV+, is often among the most accurate and well-calibrated aggregation methods available, and should be a useful method in the modern data scientist's toolbox.

References

A. Ali, J. Kolter, and R. J. Tibshirani. The multiple quantile graphical model. In *Advances in Neural Information Processing Systems*, 2016.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, 2015.

R. F. Barber, E. J. Candès, A. Ramdas, and R. J. Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference*, 10(2):455–482, 2021a.

R. F. Barber, E. J. Candès, A. Ramdas, and R. J. Tibshirani. Predictive inference with the jackknife+. *Annals of Statistics*, 49(1):486–507, 2021b.

R. E. Barlow, D. J. Bartholomew, J. M. Bremner, and H. D. Brunk. *Statistical Inference Under Order Restrictions: The Theory and Application of Isotonic Regression*. Wiley, 1972.

A. Belloni and V. Chernozhukov. ℓ_1-penalized quantile regression in high-dimensional sparse models. *Annals of Statistics*, 39(1):82–130, 2011.

K. Benidis, S. S. Rangapuram, V. Flunkert, B. Wang, D. Maddix, C. Turkmen, J. Gasthaus, M. Bohlke-Schneider, D. Salinas, L. Stella, L. Callot, and T. Januschowski. Neural forecasting: Introduction and literature overview. arXiv: 2004.10240, 2020.

H. D. Bondell, B. J. Reich, and H. Wang. Noncrossing quantile regression curve estimation. *Biometrika*, 97 (4):825–838, 2010.

J. Bracher, E. L. Ray, T. Gneiting, and N. G. Reich. Evaluating epidemic forecasts in an interval format. *PLOS Computational Biology*, 17(2):1–15, 2021.

J. Browell, C. Gilbert, R. Tawn, and L. May. Quantile combination for the eem20 wind power forecasting competition. In *Proceedings of the International Conference on the European Energy Market*, 2020.

V. Chernozhukov, I. Fernández-Val, and A. Galichon. Quantile and probability curves without crossing. *Econometrica*, 78(3):1093–1125, 2010.

D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In *International Conference on Learning Representations*, 2016.

R. Cumings-Menon and M. Shin. Probability forecast combination via entropy regularized Wasserstein distance. *Entropy*, 22(9), 2020.

W. Dabney, M. Rowland, M. Bellemare, and R. Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

H. Dette and S. Volgushev. Non-crossing non-parametric estimates of quantile curves. *Journal of the Royal Statistical Society: Series B*, 70(3):609–627, 2008.

T. G. Dietterich. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, 2000.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. J. Smola. AutoGluon-Tabular: Robust and accurate AutoML for structured data. arXiv: 2003.06505, 2020.

R. Fakoor, J. Mueller, N. Erickson, P. Chaudhari, and A. J. Smola. Fast, accurate, and simple models for tabular data via augmented distillation. In *Advances in Neural Information Processing Systems*, 2020.

G. Fung, O. Mangasarian, and J. Shavlik. Knowledge-based support vector machine classifiers. In *Advances in Neural Information Processing Systems*, 2002.

P. Gaillard, Y. Goude, and R. Nedellec. Additive models and robust aggregation for GEFCom2014 probabilistic electric load and electricity price forecasting. *International Journal of forecasting*, 32(3):1038–1050, 2016.

C. Genest. Vincentization revisited. *Annals of Statistics*, 20(2):1137–1142, 1992.

P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 6:3–42, 2006.

T. Gneiting and M. Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1: 125–151, 2014.

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

T. Gneiting and R. Ranjan. Combining predictive distributions. *Electronic Journal of Statistics*, 7:1747–1782, 2013.

A. Hald. *A History of Mathematical Statistics from 1750 to 1930*. Wiley, 1998.

G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities*. Camrbridge, 1934.

K. Hatalis, A. J. Lamadrid, K. Scheinberg, and S. Kishore. Smooth pinball neural network for probabilistic forecasting of wind power. arXiv: 1710.01720, 2017.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

S. Jain, G. Liu, J. Mueller, and D. Gifford. Maximizing overall diversity for improved uncertainty estimates in deep ensembles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

G. Kapetanios, J. Mitchell, S. Price, and N. Fawcett. Generalised density forecast combinations. *Journal of Econometrics*, 188(1):150–165, 2015.

G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, 2017.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

D. Kivaranovic, K. D. Johnson, and H. Leeb. Adaptive, distribution-free prediction intervals for deep networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2020.

R. Koenker. *Quantile Regression*. Cambridge University Press, 2005.

R. Koenker. Additive models for quantile regression: Model selection and confidence bandaids. *Brazilian Journal of Probability and Statistics*, 25(3):239–262, 2011.

R. Koenker and G. Bassett. Regression quantiles. *Econometrica*, 46(1):33–50, 1978.

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

R. J. Kuczmarski. *CDC growth charts: United States*. US Department of Health and Human Services, Centers for Disease Control and Prevention, 2000.

V. Kuleshov, N. Fenner, and S. Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, 2018.

F. Laio and S. Tamea. Verification tools for probabilistic forecasts of continuous hydrological variables. *Hydrology and Earth System Sciences*, 11:1267–1277, 2007.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

Q. V. Le, A. J. Smola, and T. Gärtner. Simpler knowledge-based support vector machines. In *Proceedings of the International Conference on Machine Learning*, 2006.

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

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

K. C. Lichtendahl, Y. Grushka-Cockayne, and R. L. Winkler. Is it better to average probabilities or quantiles? *Management Science*, 59(7):1594–1611, 2013.

L. R. Lima and F. Meng. Out-of-sample return predictability: A quantile combination approach. *Journal of Applied Econometrics*, 32(4):877–895, 2017.

K. Maciejowska, J. Nowotarski, and R. Weron. Probabilistic forecasting of electricity spot prices using factor quantile regression averaging. *International Journal of Forecasting*, 32(3):957–965, 2016.

J. E. Matheson and R. L. Winkler. Scoring rules for continuous probability distributions. *Management Science*, 22(10):1087–1096, 1976.

N. Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7(35):983–999, 2006.

J. Mukhoti, P. Stenetorp, and Y. Gal. On the importance of strong baselines in Bayesian deep learning. In *Advances in Neural Information Processing Systems*, 2018.

J. Nowotarski and R. Weron. Computing electricity spot price prediction intervals using quantile regression and forecast averaging. *Computational Statistics*, 30(3):791–803, 2015.

J. Nowotarski and R. Weron. Recent advances in electricity price forecasting: A review of probabilistic forecasting. *Renewable and Sustainable Energy Reviews*, 81:1548–1568, 2018. ISSN 1364–0321.

A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski. Using bayesian model averaging to calibrate forecast ensembles. *Monthly Weather Review*, 133(5):1155–1174, 2005.

R. Ranjan and T. Gneiting. Combining probability forecasts. *Journal of the Royal Statistical Society: Series B*, 72(1):71–91, 2010.

S. Rasp and S. Lerch. Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review*, 146(11):3885–3900, 2018.

R. Ratcliff. Group reaction time distributions and an analysis of distribution statistics. *Psychological Bulletin*, 86(3):446–461, 1979.

T. Robertson, F. T. Wright, and R. L. Dykstra. *Order Restricted Statistical Inference*. Wiley, 1998.

Y. Romano, E. Patterson, and E. J. Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, 2019.

S. Smyl and N. G. Hua. Machine learning methods for GEFCom2017 probabilistic load forecasting. *International Journal of Forecasting*, 35(4):1424–1431, 2019.

H. Song, T. Diethe, M. Kull, and P. Flach. Distribution calibration for regression. In *International Conference on Machine Learning*, 2019.

M. Stone. The opinion pool. *Annals of Mathematical Statistics*, 32(4):1339–1342, 1961.

N. Tagasovska and D. Lopez-Paz. Single-model uncertainties for deep learning. In *Advances in Neural Information Processing Systems*, 2019.

I. Takeuchi, Q. V. Le, T. D. Sears, and A. J. Smola. Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7:1231–1264, 2006.

E. A. C. Thomas and B. H. Ross. On appropriate procedures for combining probability within the same family. *Journal of Mathematical Psychology*, 21:136–152, 1980.

R. J. Tibshirani. quantgen: Tools for generalized quantile modeling, 2020. URL https://github.com/ryantibs/quantgen.

A. Timmermann. Forecast combinations. *Handbook of Economic Forecasting*, 1:135–196, 2006.

B. Uniejewski and R. Weron. Regularized quantile regression averaging for probabilistic electricity price forecasting. *Energy Economics*, 95(C):S0140988321000268, 2021.

M. J. Van der Laan, E. C. Polley, and A. E. Hubbard. Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6(1), 2007.

J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo. OpenML: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013.

S. B. Vincent. The function of the vibrissae in the behavior of the white rat. *Behavior Monographs*, 1(5), 1912.

V. Vovk. Conditional validity of inductive conformal predictors. *Asian Conference on Machine Learning*, 2012.

V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

V. Vovk, I. Nouretdinov, V. Manokhin, and A. Gammerman. Cross-conformal predictive distributions. In *Conformal and Probabilistic Prediction and Applications*, 2018.

R. L. Winkler. A decision-theoretic approach to interval estimation. *Journal of the American Statistical Association*, 67(337):187–191, 1972.

D. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

Z. Xie and H. Wen. Composite quantile regression long short-term memory network. In *International Conference on Artificial Neural Networks*, 2019.

S. Zhang, Y. Wang, Y. Zhang, D. Wang, and N. Zhang. Load probability density forecasting by transforming and combining quantile forecasts. *Applied Energy*, 277:115600, 2020.

A Proofs

A.1 Proof of equivalence (6)

At the outset, we assume two conditions on F: it has a derivative f, and it yields an expectation. Starting with the right-hand side in (6), we can substitute $y = F^{-1}(\tau)$ to rewrite the integral as

$$2\int_0^1 (1\{Y \le F^{-1}(\tau)\} - \tau)(F^{-1}(\tau) - Y)\,\mathsf{d}\tau = 2\int_{-\infty}^{\infty} (1\{Y \le y\} - F(y))(y - Y)f(y)\,\mathsf{d}y.$$

Let $u'(y) = 2(1\{Y \le y\} - F(y))f(y)$ and $v(y) = y - Y$. The idea is now to use integration by parts, but there are two subtleties. First, one has to be careful about framing the antiderivative u of u', since $y \mapsto 1\{Y \le y\}$ is not classically differentiable. Note that we can actually redefine u' to be

$$u'(y) = 2(1\{Y \le y\} - F(y))(f(y) - \delta_Y(y)),$$

where δ_Y is the Dirac delta function centered at Y, because the "extra" piece integrates to zero:

$$\int_{-\infty}^{\infty} 2(1\{Y \le y\} - F(y))(y - Y)\delta_Y(y)\,\mathsf{d}y = 2(1\{Y \le y\} - F(y))(y - Y)\Big|_{y=Y} = 0.$$

With this new definition of u', its antiderivative is rigorously

$$u(y) = -(1\{Y \le y\} - F(y))^2,$$

because, in the distributional sense, the derivative of the heavyside function $y \mapsto 1\{Y \le y\}$ is indeed the delta function δ_Y. Thus we have

$$\begin{aligned}\int_{-\infty}^{\infty} u'(y)v(y)\,\mathsf{d}y &= u(y)v(y)\Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} u(y)v'(y)\,\mathsf{d}y \\ &= -(1\{Y \le y\} - F(y))^2(y - Y)\Big|_{-\infty}^{\infty} + \int_{-\infty}^{\infty} (1\{Y \le y\} - F(y))^2\,\mathsf{d}y.\end{aligned}$$

The second subtlety is to show that the first term above is indeed zero. This is really a question of how fast the tails of F decay. As F yields an expectation, note that we must have $1 - F(y) \lesssim y^{-p}$ for $p > 1$ (since $1/y$ is not integrable on any one-sided interval $[a, \infty)$ for $a > 0$). Hence

$$(1 - F(y))^2 y \lesssim y^{-2p+1} \to 0 \quad \text{as } y \to \infty,$$

and the other limit, as $y \to -\infty$, is similar.

A.2 Proof of equivalence (8)

Starting with the right-hand side in (8), consider the two summands corresponding to $\tau \in \{\alpha/2, 1 - \alpha/2\}$:

$$2\psi_{\alpha/2}(Y - \ell_\alpha) + 2\psi_{\alpha/2}(Y - u_\alpha), \tag{30}$$

where we have used $\ell_\alpha = F^{-1}(\alpha/2)$ and $u_\alpha = F^{-1}(1 - \alpha/2)$. If $Y > u_\alpha$, then (30) is

$$\alpha(Y - \ell_\alpha) + 2(1 - \alpha/2)(Y - u_\alpha) = \alpha(u_\alpha - \ell_\alpha) + 2(Y - u_\alpha).$$

Meanwhile, if $Y < \ell_\alpha$, then (30) is

$$2(1 - \alpha/2)(\ell_\alpha - Y) + \alpha(u_\alpha - Y) = \alpha(u_\alpha - \ell_\alpha) + 2(\ell_\alpha - Y).$$

Lastly, if $Y \in [\ell_\alpha, u_\alpha]$, then (30) is

$$\alpha(Y - \ell_\alpha) + \alpha(u_\alpha - Y) = \alpha(u_\alpha - \ell_\alpha).$$

In each case, it matches the summand in the left-hand side of (8) corresponding to exclusion probability α, which completes this proof.

A.3 Proof of Proposition 2

In this proof we denote the pinball loss, for a vector of quantiles $q \in \mathbb{R}^m$ across quantiles levels $\tau_1, \ldots, \tau_m$, by

$$L(q, y) = \sum_{i=1}^{m} \psi_{\tau_i}(y - q_i).$$

Proof of part (i). Suppose that there are only two quantile levels $\tau_1 < \tau_2$, and abbreviate $q_i = \hat{g}(x; \tau_i)$ for $i = 1, 2$. We will prove in this special case that sorting can only improve the pinball loss. Importantly, this suffices to prove the result in the general case as well, since sorting can always be achieved by a sequence of pairwise swaps (via bubble sort).

Assume $q_1 > q_2$, otherwise there is nothing to prove. Denote the pinball loss by

$$L(q, y) = \psi_{\tau_1}(y - q_1) + \psi_{\tau_2}(y - q_2).$$

We now break the argument down into three cases. In each case, when we speak of the loss difference, we mean the pre-sort minus the post-sort pinball loss (first line minus second line in each display that follows).

Case 1: $y > q_1$. Denoting $a = y - q_1$ and $b = y - q_2$ (and noting $b \geq a$), we have

$$\begin{aligned} L((q_1, q_2), y) &= \tau_1 a + \tau_2 b, \\ L((q_2, q_1), y) &= \tau_1 b + \tau_2 a, \end{aligned}$$

so the loss difference is $(\tau_2 - \tau_1)(b - a) \geq 0$. I

Case 2: $y \in [q_2, q_1]$. Denoting $a = q_1 - y$ and $b = y - q_2$, we have

$$\begin{aligned} L((q_1, q_2), y) &= (1 - \tau_1) a + \tau_2 b, \\ L((q_2, q_1), y) &= \tau_1 b + (1 - \tau_2) a, \end{aligned}$$

so the loss difference is $(\tau_2 - \tau_1)(a + b) \geq 0$.

Case 3: $y < q_2$. Denoting $a = q_1 - y$ and $b = q_2 - y$ (and noting $a \geq b$), we have

$$\begin{aligned} L((q_1, q_2), y) &= (1 - \tau_1) a + (1 - \tau_2) b, \\ L((q_2, q_1), y) &= (1 - \tau_1) b + (1 - \tau_2) a, \end{aligned}$$

so the loss difference is $(\tau_2 - \tau_1)(a - b) \geq 0$. This completes the proof.

Proof of part (ii). As in the proof of part (i), we will prove the result in a special case, and argue that it suffices to prove the result in general. Given $k + \ell$ quantile levels $\tau_1 < \cdots < \tau_{k+\ell}$, we abbreviate $q_i = \hat{g}(x; \tau_i)$ for $i = 1, \ldots, k + \ell$. We suppose, in what follows, that $q_1 = \cdots = q_k$ and $q_{k+1} = \cdots = q_{k+\ell}$. We then show that averaging can only improve the pinball loss, where by averaging, we mean replacing the original vector $(q_1, \ldots, q_{k+\ell})$ by $(\bar{q}, \ldots, \bar{q})$, with $\bar{q} = \frac{1}{k+\ell} \sum_{i=1}^{k+\ell} q_i$. This suffices to prove the desired result, because isotonic projection can be solved using PAVA, which carries out a sequence of steps (whenever there is a violation of the monotonicity condition) that are precisely of the form we study here.

Assume $q_k > q_{k+1}$, otherwise there is nothing to prove. We break the argument down into four cases. In each case, as in the proof of part (i), when we speak of the loss difference below, we mean the pre-sort loss minus the post-sort loss.

Case 1: $y \geq q_k$. We have

$$\begin{aligned} L((q_1, \ldots, q_{k+\ell}), y) &= \sum_{i=1}^{k} \tau_i (y - q_k) + \sum_{i=k+1}^{k+\ell} \tau_i (y - q_{k+1}), \\ L((\bar{q}, \ldots, \bar{q}), y) &= \sum_{i=1}^{k+\ell} \tau_i (y - \bar{q}), \end{aligned}$$

so the loss difference is

$$\sum_{i=1}^{k}\tau_i(\bar{q}-q_k)+\sum_{i=k+1}^{k+\ell}\tau_i(\bar{q}-q_{k+1})=\frac{k\ell}{k+\ell}\left(\frac{1}{\ell}\sum_{i=k+1}^{k+\ell}\tau_i-\frac{1}{k}\sum_{i=1}^{k}\tau_i\right)(q_k-q_{k+1})\geq 0.$$

Case 2: $y\in[q_{k+1},q_k]$ and $y>\bar{q}$. We have

$$L((q_1,\ldots,q_{k+\ell}),y)=\sum_{i=1}^{k}(1-\tau_i)(q_k-y)+\sum_{i=k+1}^{k+\ell}\tau_i(y-q_{k+1}),$$
$$L((\bar{q},\ldots,\bar{q}),y)=\sum_{i=1}^{k+\ell}\tau_i(y-\bar{q}),$$

so the loss difference is

$$k(q_k-y)+\sum_{i=1}^{k}\tau_i(\bar{q}-q_k)+\sum_{i=k+1}^{k+\ell}\tau_i(\bar{q}-q_{k+1})=k(q_k-y)+\frac{k\ell}{k+\ell}\left(\frac{1}{\ell}\sum_{i=k+1}^{k+\ell}\tau_i-\frac{1}{k}\sum_{i=1}^{k}\tau_i\right)(q_k-q_{k+1})\geq 0.$$

Case 3: $y\in[q_{k+1},q_k]$ and $y\leq\bar{q}$. We have

$$L((q_1,\ldots,q_{k+\ell}),y)=\sum_{i=1}^{k}(1-\tau_i)(q_k-y)+\sum_{i=k+1}^{k+\ell}\tau_i(y-q_{k+1}),$$
$$L((\bar{q},\ldots,\bar{q}),y)=\sum_{i=1}^{k+\ell}(1-\tau_i)(\bar{q}-y),$$

so the loss difference is

$$\ell(y-q_{k+1})+\sum_{i=1}^{k}(1-\tau_i)(q_k-\bar{q})+\sum_{i=k}^{k+\ell}(1-\tau_i)(q_{k+1}-\bar{q})$$
$$=\ell(y-q_{k+1})+\frac{k\ell}{k+\ell}\left(\frac{1}{k}\sum_{i=1}^{k}(1-\tau_i)-\frac{1}{\ell}\sum_{i=k+1}^{k+\ell}(1-\tau_i)\right)(q_k-q_{k+1})\geq 0.$$

Case 4: $y<q_k$. Then

$$L((q_1,\ldots,q_{k+\ell}),y)=\sum_{i=1}^{k}(1-\tau_i)(q_k-y)+\sum_{i=k+1}^{k+\ell}(1-\tau_i)(q_{k+1}-y),$$
$$L((\bar{q},\ldots,\bar{q}),y)=\sum_{i=1}^{k+\ell}(1-\tau_i)(\bar{q}-y),$$

so the loss difference is

$$\sum_{i=1}^{k}(1-\tau_i)(q_k-\bar{q})+\sum_{i=k}^{k+\ell}(1-\tau_i)(q_{k+1}-\bar{q})=\frac{k\ell}{k+\ell}\left(\frac{1}{k}\sum_{i=1}^{k}(1-\tau_i)-\frac{1}{\ell}\sum_{i=k+1}^{k+\ell}(1-\tau_i)\right)(q_k-q_{k+1})\geq 0.$$

This completes the proof.

A.4 Proof of Proposition 5

Proof of part (i). This is immediate from the fact that $f(v)=w_1f_1(v)+w_2f_2(v)$ and $f_2(v)/f_1(v)\to 0$.

Proof of part (ii). This part is more subtle. Using (14), note that we may write

$$\frac{1}{\bar{f}(\bar{Q}(u))} = \frac{w_1}{f_1(Q_1(u))} + \frac{w_2}{f_2(Q_2(u))}.$$

For $v = \bar{Q}(u)$, consider

$$\frac{f_1(v)}{\bar{f}(v)} = \frac{w_1 f_1(\bar{Q}(u))}{f_1(Q_1(u))} + \frac{w_2 f_1(\bar{Q}(u))}{f_2(Q_2(u))}.$$

It will be convenient to work on the log scale. Introduce $p_1 = \log f_1$ and $p_2 = \log f_2$. Then it suffices to show that as $u \to 1$,

$$\begin{gathered} p_1(\bar{Q}(u)) - p_1(Q_1(u)) \to \infty, \quad \text{and} \\ p_1(\bar{Q}(u)) - p_2(Q_2(u)) \text{ remains bounded away from } -\infty, \end{gathered}$$

or

$$\begin{gathered} p_1(\bar{Q}(u)) - p_2(Q_2(u)) \to \infty, \quad \text{and} \\ p_1(\bar{Q}(u)) - p_1(Q_1(u)) \text{ remains bounded away from } -\infty. \end{gathered}$$

We now divide the argument into two cases. Without a loss of generality, we set $w_1 = w_2 = 1/2$.

Case 1: $p_1(Q_1(u)) - p_2(Q_2(u))$ remains bounded away from $-\infty$. Denoting $v = \bar{Q}(u)$, $v_1 = Q_1(u)$, $v_2 = Q_2(u)$, we have, using log-concavity of p_1,

$$\begin{aligned} p_1(v) - p_2(v_2) &= p_1(v_1/2 + v_2/2) - p_2(v_2) \\ &\geq p_1(v_1)/2 + p_1(v_2)/2 - p_2(v_2) \\ &= \big(p_1(v_1) - p_2(v_2)\big)/2 + \big(p_1(v_2) - p_2(v_2)\big)/2. \end{aligned}$$

The first term above is bounded below by assumption; the second term diverges to ∞ (under our hypothesis that $f_2(v)/f_1(v) \to 0$).

Case 2: $p_1(Q_1(u)) - p_2(Q_2(u)) \to -\infty$. We have, just as in the first case,

$$\begin{aligned} p_1(v) - p_1(v_1) &= p_1(v_1/2 + v_2/2) - p_1(v_1) \\ &\geq p_1(v_1)/2 + p_1(v_2)/2 - p_2(v_2) + p_2(v_2) - p_1(v_1) \\ &= -\big(p_1(v_1) - p_2(v_2)\big)/2 + \big(p_1(v_2) - p_2(v_2)\big)/2. \end{aligned}$$

Now both terms above diverge to ∞, and this completes the proof.

B More experimental details and results

In the following, we provide further details about models: the hyperparameter space, optimization details, etc. In our experiments, for each method described below, we perform hyperparameter tuning over a randomly chosen subset of 20 values from the full Cartesian product of possibilities.

B.1 Base models

We consider $p = 6$ base models in total. The first 3 are neural network models, that each compute quantile estimates differently. Each use a standard feed-forward (multilayer perceptron) architecture.

- Conditional Gaussian network (CGN, Lakshminarayanan et al., 2017): this model assumes that the conditional distribution of $Y|X = x$ is Gaussian. It therefore outputs estimates of sufficient statistics, namely the conditional mean $\mu(x)$ and variance $\sigma^2(x)$. It is optimized by minimizing the negative log-likelihood, and subsequent quantile estimation is done based on the normal distribution.

- Simultaneous quantile regression (SQR, Tagasovska and Lopez-Paz, 2019): this model takes as input a target quantile level τ concatenated with a feature vector x, and outputs a corresponding quantile estimate. It is trained by minimizing a pinball loss for many different randomly sampled quantile levels τ. To estimate multiple quantile levels with SQR, we must thus feed in the same x numerous times, each paired with a different quantile level τ.

- Deep quantile regression (DQR): this model falls naturally out of our neural aggregation framework, as explained in Section 3.3. It is the only neural network based model considered in this paper that can easily utilize different isotonization approaches discussed in Section 4. For DQR as a base model in all experiments, we use the simplest method for ensuring noncrossing among the all options: the crossing penalty along with post sorting.

We optimize each of these neural network models using Adam (Kingma and Ba, 2015), and using ELU activation function (Clevert et al., 2016). We adaptively vary the mini-batch size depending on the data set size. They also share the same architecture/optimization hyperparameter search space: # of fully connected layers: {2, 3}, # of hidden units: {64, 128}, dropout ratio: {0.0, 0.05, 0.1}, learning rate: {1e-3, 3e-4}, weight decay: {1e-5, 1e-7}. In all settings, we use early stopping where the validation loss is evaluated every epoch and if it has not decreased for the last 500 updates, the optimization is stopped by returning the epoch with the lowest validation loss.

The next 3 base models are not neural networks.

- Quantile random forest (RandomForest, Meinshausen, 2006) and extremely randomized trees (ExtraTrees, Geurts et al., 2006): both models are based on random forests trained as regular (conditional mean) regressors. After training, quantile estimates are computed via empirical quantiles of the data in relevant leaf nodes of the trees.

- Quantile gradient boosting (LightGBM, Ke et al., 2017): this model is a gradient boosting framework that uses tree-based base learners. As a boosting framework, it can optimize the pinball loss, and thus be directly used for quantile regression. It handles multiple quantile levels by simply using a separate model for each level.

For the random forests models, we use the `scikit-garden` (`https://scikit-garden.github.io/`) implementation for both, and both have the same hyperparameter search space: minimum # of samples for splitting nodes: {8, 16, 64}, minimum # of sample for leaf nodes: {8, 16, 64} .

For the gradient boosting model, the hyperparameter space is: # of leaves: {10, 50, 100}, minimum child samples: {3, 9, 15}, minimum child weight: {1e-2, 1e-1, 1}, subsample ratio: {0.4, 0.6, 0.8}, subsample ratio of columns: {0.4, 0.6}, ℓ_1 regularization weight: {1e-1, 1, 5}, ℓ_2 regularization weight: {1e-1, 1, 5}.

B.2 Aggregation models

We consider 4 aggregation models from outside framework: Average, Median, quantile regression averaging (QRA, Nowotarski and Weron, 2015), and factor QRA (FQRA, Maciejowska et al., 2016). The first 3 do not require tuning. For FQRA, we tune over the number of factors, between 1-6 (the number of base models).

We also study a total of 6 aggregation models within our framework: three global aggregators and three local ones (each having coarse, medium, or fine weight parameterizations). For the global aggregators, the hyperparameter search space we use is: crossing penalty weight λ: {0.5, 1.0, 2.0, 5.0, 10.0}, crossing penalty margin scaling δ_0: {1e-1, 5e-2, 1e-2, 1e-3, 1e-4}. For the local aggregators, the hyperparameter search space additionally includes: # of layers: {2, 3}, # of hidden units: {64, 128}, dropout ratio: {0.0, 0.05, 0.1}. All are optimized using the Adam optimizer (Kingma and Ba, 2015), ELU activation function (Clevert et al., 2016), and adaptively-varied mini-batch size. We also employ the same early stopping strategy as described above.

B.3 Revisiting Figure 1

Figure 12 plots the same results in Figure 1 but it displays a difference in WIS values, rather than a ratio of WIS values (which is how we define relative WIS). That is, shown on the y-axis is the difference of the WIS

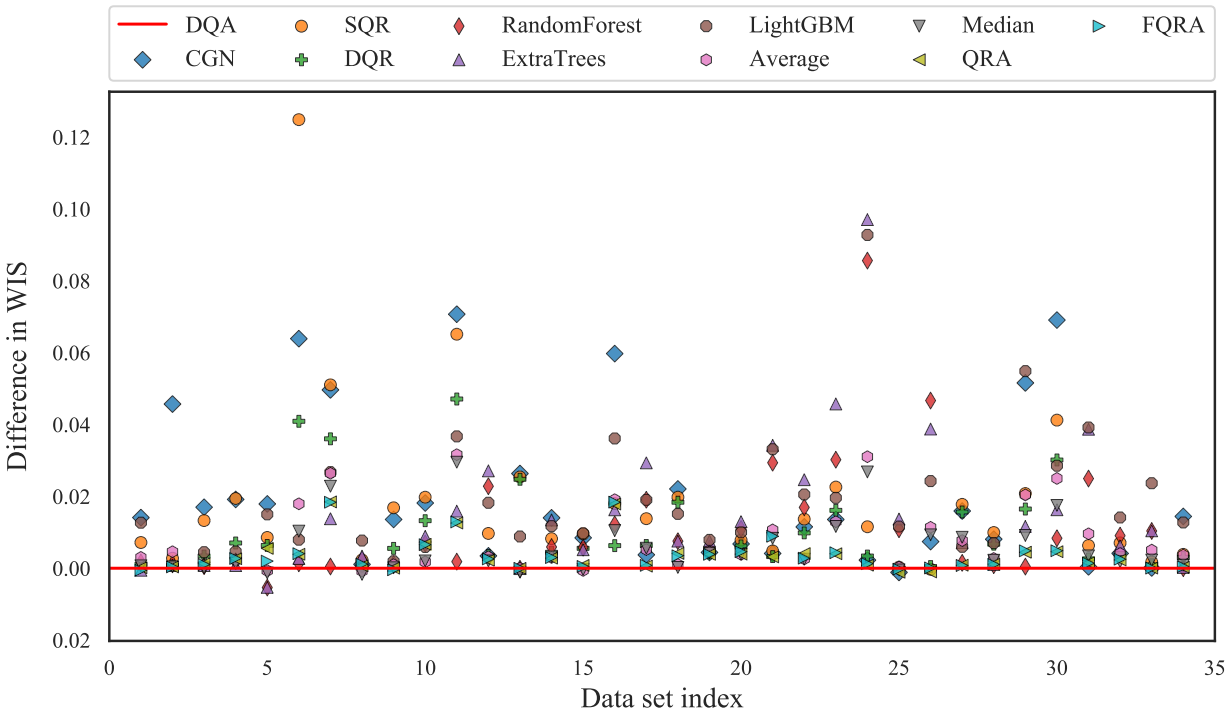

Figure 12: *As in Figure 1, but with WIS differences (of each method to DQA) on the y-axis rather than WIS ratios.*

of each quantile regression minus that of DQA, where now 0 indicates equal WIS performance to DQA, and a number greater than 0 indicates worse performance than DQA. In general, WIS will be on the scale of the response variable, but recall that we have standardized the response variable in each data set, thus the comparison in Figure 12 makes sense across data sets. The story in this figure is very much similar to that in Figure 1, except that we can more clearly see that for some data sets with low PVE values (e.g., at index 5), there are a handful of models that perform a little better than DQA.