SIAM REVIEW
Vol. 66, No. 1, pp. 91–122

Easy Uncertainty Quantification (EasyUQ): Generating Predictive Distributions from Single-Valued Model Output[*]

Eva-Maria Walz[†]
Alexander Henzi[‡]
Johanna Ziegel[§]
Tilmann Gneiting[¶]

Abstract. How can we quantify uncertainty if our favorite computational tool—be it a numerical, statistical, or machine learning approach, or just any computer model—provides single-valued output only? In this article, we introduce the Easy Uncertainty Quantification (EasyUQ) technique, which transforms real-valued model output into calibrated statistical distributions, based solely on training data of model output–outcome pairs, without any need to access model input. In its basic form, EasyUQ is a special case of the recently introduced isotonic distributional regression (IDR) technique that leverages the pool-adjacent-violators algorithm for nonparametric isotonic regression. EasyUQ yields discrete predictive distributions that are calibrated and optimal in finite samples, subject to stochastic monotonicity. The workflow is fully automated, without any need for tuning. The Smooth EasyUQ approach supplements IDR with kernel smoothing, to yield continuous predictive distributions that preserve key properties of the basic form, including both stochastic monotonicity with respect to the original model output and asymptotic consistency. For the selection of kernel parameters, we introduce multiple one-fit grid search, a computationally much less demanding approximation to leave-one-out cross-validation. We use simulation examples and forecast data from weather prediction to illustrate the techniques. In a study of benchmark problems from machine learning, we show how EasyUQ and Smooth EasyUQ can be integrated into the workflow of neural network learning and hyperparameter tuning, and we find EasyUQ to be competitive with conformal prediction as well as more elaborate input-based approaches.

Key words. isotonic regression, kernel smoothing, computational model output, neural network training, pool-adjacent-violators (PAV) algorithm, probabilistic forecast, proper scoring rule

MSC codes. 68T01, 62G08, 86A10

DOI. 10.1137/22M1541915

[*] Received by the editors December 20, 2022; accepted for publication (in revised form) August 31, 2023; published electronically February 8, 2024.
https://doi.org/10.1137/22M1541915

Funding: The work of the first and fourth authors was supported by the German Science Foundation (DFG) project C2 of the Transregional Collaborative Research Center SFB/TRR 165 "Waves to Weather." The work of the second and third authors was supported by the Swiss National Science Foundation. The work of the fourth author was supported by the Klaus Tschira Foundation.

[†] Institute for Stochastics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, and Computational Statistics (CST) group, Heidelberg Institute for Theoretical Studies, Heidelberg, Germany (eva-maria.walz@kit.edu).

[‡] Seminar for Statistics, ETH Zürich, Zürich, Switzerland (alexander.henzi@stat.math.ethz.ch).

[§] Institute for Mathematical Statistics and Actuarial Science, University of Bern, Bern, Switzerland (johanna.ziegel@stat.unibe.ch).

[¶] Computational Statistics (CST) group, Heidelberg Institute for Theoretical Studies, Heidelberg, Germany, and Institute for Stochastics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany (tilmann.gneiting@h-its.org).

Contents

1. Introduction.

In an editorial that remains topical and relevant [71], SIAM President Nick Trefethen noted the following a decade ago:

> An answer that used to be a single number may now be a statistical distribution.

Indeed, with the increasing reliance of real-world decisions on the output of computer models—which might be numerical or statistical, parametric or nonparametric, simple or complex—and the advent of uncertainty quantification as a scientific field of its own, there is a growing consensus in the computational sciences community that decisions ought to be informed by full predictive distributions rather than single-valued model output. For recent perspectives on these issues and uncertainty quantification in general, we refer the reader to topical monographs [24, 68, 70] and review articles [1, 5, 26, 60].

How can we quantify uncertainty if the computational model at hand provides single-valued output only? With Nick Trefethen's comment in mind, we address the following problem: Given single-valued, univariate model output, how can we generate a prediction interval or, more generally, a probabilistic forecast in the form of a full statistical distribution? In this work, we introduce the Easy Uncertainty Quantification (EasyUQ) technique that serves this task, based solely on a training archive of model output–outcome pairs. The single-valued, univariate model output can be of any type—e.g., it might stem from a physics-based numerical model, might arise from a purely statistical or machine learning model, or might be based on human expertise. In a nutshell, EasyUQ applies the recently introduced isotonic distributional regression (IDR) [35] approach to generate discrete, calibrated predictive distributions, conditional on the model output at hand. The name stems from the threefold reasons that EasyUQ operates on the final model output only without any need for access to the original model input; that the method honors a natural assumption of isotonicity, namely, that higher values of the model output entail predictive distributions that are larger in stochastic order; and that the basic version of EasyUQ does not involve any tuning parameters and thus does not require user intervention. The more elaborate Smooth EasyUQ approach introduced in this paper subjects the EasyUQ distribution to kernel smoothing, to yield predictive probability densities that preserve key properties of the basic approach. Prediction intervals are readily extracted; e.g., the equal-tailed 90% interval forecast is framed by the quantiles at level 0.05 and 0.95 of the predictive distribution.

As the EasyUQ approach requires training data, it addresses general "weather-like" tasks [5, p. 441], which are characterized by frequent repetition of the task—e.g., hourly, daily, monthly, at numerous spatial locations, or for a range of customers or patients—in concert with short-to-moderate lead times of the forecasts, thus enabling the development of a sizable archive of forecast–outcome pairs. EasyUQ makes the best possible use of single-valued model output in the sense of empirical score minimization on the training data, subject to the natural constraint of isotonicity. Specifically, the larger the model output, the larger the predictive distribution, in the technical sense of the familiar stochastic order [65], i.e., the respective cumulative distribution functions (CDFs) do not intersect and their graphs move to the right as the model output increases. Subject to the isotonicity constraint, the EasyUQ distributions are optimal with respect to a large class of loss functions that includes the popular continuous ranked probability score (CRPS) [28, 47], all proper scoring rules for binary events, and all proper scoring rules for quantile forecasts, among others [35, Thm. 2]. For prediction, the EasyUQ and Smooth EasyUQ distributions are interpolated to the value of the model output at hand, while respecting isotonicity.

Figure 1 illustrates the EasyUQ approach on WeatherBench [57], a benchmark dataset for weather prediction that serves as a running example in this paper. Panel (a) shows single-valued forecasts of upper air temperature from the HRES numerical weather prediction model run by the European Centre for Medium-Range Weather Forecasts (ECMWF) [49] along with the associated observed temperatures in February 2017. The training data for EasyUQ, which converts the single-valued HRES model output into conditional predictive distributions, comprise the forecast–outcome pairs from 2010 through 2016, as illustrated in the scatter plot in panel (c). Panel (d) shows the EasyUQ predictive distributions for February 2017, which derive from the single-valued HRES forecasts in panel (a) and can be compared to the computationally much more expensive ECMWF ensemble forecasts in panel (b). To facilitate the comparison, panel (c) includes inset diagrams with the ECMWF ensemble and EasyUQ predictive CDFs for two particular days. Panels (e) and (f) show EasyUQ predictive CDFs and Smooth EasyUQ predictive densities when the HRES model output equals 263, 268, and 273 degrees Kelvin, respectively. The isotonicity property of the EasyUQ distributions is reflected by the nonintersecting CDFs. The boxes in panels (b) and (d) range from the 25th to the 75th percentile of the distribution and generate 50%

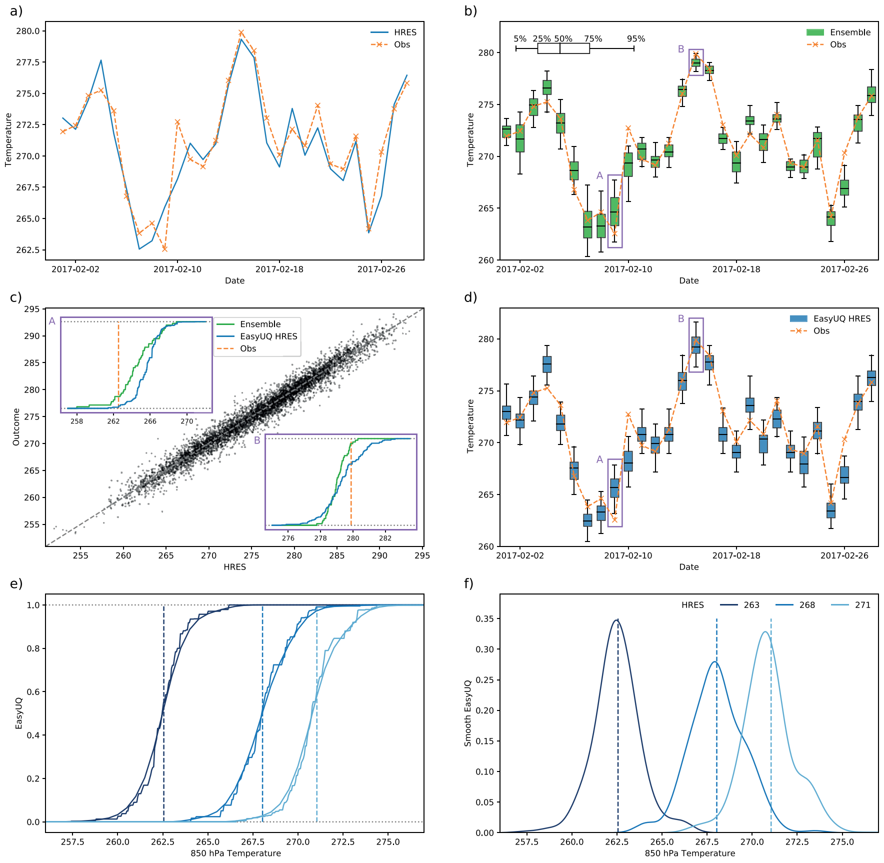

Fig. 1 *EasyUQ illustrated on WeatherBench data. Time series of three days ahead* (a) *single-valued HRES model forecasts,* (b) *state-of-the-art ECMWF ensemble forecasts, and* (d) *basic EasyUQ predictive distributions based on the single-valued HRES forecast along with associated outcomes of upper air temperature in February* 2017 *at a grid point over Poland, in degrees Kelvin. The boxplots show the quantiles at levels* 0.05, 0.25, 0.50, 0.75, *and* 0.95 *of the predictive distributions.* (c) *Scatterplot of HRES model output and associated outcomes in* 2010 *through* 2016, *which serve as training data. The inset diagrams show the ECMWF and EasyUQ predictive CDFs for* (A) 9 *February* 2017 *and* (B) 15 *February* 2017, *respectively.* (e) *Basic and Smooth EasyUQ predictive CDFs and* (f) *Smooth EasyUQ predictive densities at selected values of the single-valued HRES forecast. For further details, see section* 2.3.

prediction intervals, whereas the whiskers range from the 5th to the 95th percentile and form 90% intervals.

The remainder of the paper is organized as follows. Section 2 provides comprehensive descriptions of IDR and the basic EasyUQ method and gives details, background information, and a comparison to conformal prediction [72, 75] for both the WeatherBench temperature forecast challenge and a precipitation forecast example. In section 3, we introduce the Smooth EasyUQ technique, show that it retains the isotonocity property of the basic method, and discuss statistical large-sample consistency. For the selection of kernel parameters, we introduce multiple one-fit grid search, a computationally much less demanding approximate version of cross-validation. In section 4, we demonstrate that EasyUQ can be integrated into the workflow of neural network learning and hyperparameter tuning, and we use benchmark problems to compare its predictive performance to state-of-the-art techniques from machine learning and conformal prediction. The paper closes with remarks in section 5, where we return to the discussion of input-based vs. output-based uncertainty quantification.

While the basic version of EasyUQ arises as a special case of the extant IDR technique [35], we take the particular perspective of the conversion of single-valued model output into predictive distributions. Original contributions in this paper include the development of the Smooth EasyUQ method (sections 3.1 and 3.2), a detailed comparison to conformal prediction in case studies (sections 2.3, 2.4, and 3.3) and from computational and methodological perspectives (sections 3.4 and 5), and the integration and benchmarking of EasyUQ and Smooth EasyUQ for neural networks (section 4). In Appendix A, we prove the consistency of smoothed CDFs in general settings, which supports the usage of Smooth EasyUQ but is a result of broader and independent interest.

2. Basic EasyUQ.

We begin the section with a prelude on the evaluation of predictions in the form of full statistical distributions. Then we describe the IDR and EasyUQ techniques, and we illustrate EasyUQ on the WeatherBench data from [57] and on precipitation forecasts [35]. Generally, EasyUQ depends on the availability of training data of the form

$$(x_i, y_i), \quad i = 1, \ldots, n, \tag{2.1}$$

where $x_i \in \mathbb{R}$ is the single-valued model output and $y_i \in \mathbb{R}$ is the respective real-world outcome for $i = 1, \ldots, n$. For subsequent discussion, we note the contrast to more elaborate, input-based ways of uncertainty quantification that require access to the features or covariates from which the model output x_i is generated. In the WeatherBench example from Figure 1, we have training data comprising twice daily HRES forecasts and the associated observed temperatures in 2010 through 2016 as illustrated in panel (c), where $n = 5{,}114$, but we do not have access to the excessively high-dimensional input to the HRES model. In practice, one needs to find a predictive distribution given the value x of the model output at hand, which may or may not be among the training values $x_1 \leq \cdots \leq x_n$, and some form of interpolation is needed, while retaining isotonicity. In panel (e) of Figure 1 we illustrate predictive CDFs when x equals equals 263, 268, and 273 degrees Kelvin, respectively.

Extensions of this setting to situations where single-valued output from multiple computational models is available can be handled within the IDR framework, as we discuss below. If model output and real-world outcome are vector-valued—e.g., when temperature is predicted at multiple sites simultaneously—EasyUQ can be applied to each component independently, and the EasyUQ distributions for the components can be merged by exploiting dependence structures in the training data, based on empirical copula techniques such as the Schaake shuffle [61].

2.1. Evaluating Predictive Distributions.

A widely accepted principle in the generation of predictive distributions is that sharpness ought to be maximized subject to calibration [25]. Maximizing sharpness requires forecasters to provide informative, concentrated predictive distributions, and calibration posits that probabilities derived from these distributions conform with actual observed frequencies. This is in line with and generalizes the classical goal of prediction intervals being as narrow as possible while attaining nominal coverage.

A key tool for evaluating and comparing predictive distributions under this principle is that of proper scoring rules [28, 47], which are functions $\mathrm{S}(P, y)$ mapping a predictive distribution P and the outcome y to a numerical score such that

$$\mathbb{E}_{Y \sim P}[\mathrm{S}(P, Y)] \leq \mathbb{E}_{Y \sim P}[\mathrm{S}(Q, Y)]$$

for all distributions P, Q in a given class $\mathcal{P}$. Here $\mathbb{E}_{Y \sim P}[\cdot]$ denotes the expected value of the quantity in parentheses when Y follows the distribution P. From a decision-theoretic point of view, proper scoring rules encourage truthful forecasting, since forecasters minimize their expected score if they issue predictive distributions that correspond to their true beliefs.

Arguably the most widely used proper scoring rules are the continuous ranked probability score (CRPS),

$$\text{(2.2)} \qquad \mathrm{CRPS}(F, y) = \int_{-\infty}^{\infty} (F(z) - \mathbb{1}\{z \geq y\})^2 \, \mathrm{d}z,$$

which can be applied to cumulative distribution functions (CDFs) F on the real line for which the corresponding distribution has finite first moment, and the logarithmic score for a predictive CDF F with density f,

$$\text{(2.3)} \qquad \mathrm{LogS}(F, y) = -\log(f(y)).$$

The popularity of the CRPS is due to the facts that it allows arbitrary types of predictive distributions (discrete, continuous, mixed discrete-continuous), is reported in the same unit as the outcome, and reduces to the absolute error $\mathrm{AE}(x, y) = |x - y|$ if F assigns probability one to a point $x \in \mathbb{R}$. The LogS is (save for a change of sign) the ubiquitous loss function in maximum likelihood estimation. Closed form expressions for the CRPS and LogS are available for the most commonly used parametric distributions and have been implemented in software packages [38]. In practice, forecast methods are compared in terms of their average score over a collection (F_j, y_j) for $j = 1, \ldots, n$,

$$\bar{\mathrm{S}} = \frac{1}{n} \sum_{j=1}^{n} \mathrm{S}(F_j, y_j),$$

and the method achieving the lowest average score is considered superior.

2.2. Basic EasyUQ: Leveraging the Isotonic Distributional Regression (IDR) Technique. In this section, it will be instructive to think of the quantities involved as random variables, which we emphasize by using the upper case in the notation. If model output X serves to predict a future quantity Y, then one typically assumes that Y tends to attain higher values as X increases; in fact, the isotonicity assumption can be regarded as a natural requirement for X to be a useful forecast for Y. Isotonic distributional regression (IDR) is a recently introduced, nonparametric method for estimating the conditional distributions of a real-valued outcome Y given a covariate or feature vector X from a partially ordered space under general assumptions of isotonicity [35]. EasyUQ leverages the basic special case of IDR where X is the single-valued model output at hand. We review the construction and the most relevant properties of IDR for uncertainty quantification; for detailed formulations and proofs we refer the reader to the original paper [35].

Formally, EasyUQ assumes that the conditional distributions of the outcome Y given the model output X, which we identify with the CDFs $F_x(y) = \mathbb{P}(Y \le y \mid X = x)$, are increasing in stochastic order [65] in x, i.e., $F_x(y) \ge F_{x'}(y)$ for all $y \in \mathbb{R}$ if $x \le x'$, or equivalently $q_x(\alpha) \le q_{x'}(\alpha)$ for all $\alpha \in (0,1)$, where $q_x(\alpha) = F_x^{-1}(\alpha)$ is the conditional lower α-quantile. In plain words, the probability of the outcome Y exceeding any threshold y increases with the model output x. Isotonicity in this sense is a natural assumption that one expects to hold, to a reasonable degree of approximation, in many types of applications. An important exception arises for location-scale families. Specifically, the arguments in the proof of Proposition 1 in Gneiting and Vogel [30] imply that isotonicity is violated when the true predictive distributions come from a location-scale family with varying scale.[1] However, the practical impact of this result is limited, due to the fact that in typical practice the scale parameter varies only mildly [29] and violations remain minor. Crucially, estimators that enforce isotonicity tend to be superior to estimators that do not, even when the key assumption is violated, provided the deviation from isotonicity remains modest. For an illustration in a simulation setting, see the nonisotonic scenario (25) in Table 1 of Henzi, Ziegel, and Gneiting [35], where IDR retains acceptable performance relative to its competitors, despite the key assumption being violated. For a rigorous result, Theorem 7 of El Barmi and Mukerjee [20] demonstrates that, in the special case of discrete model output, EasyUQ has smaller large sample estimation error than nonisotonic alternatives even under mild violations of the isotonicity assumption.

EasyUQ assumes isotonicity with respect to the usual stochastic order. In situations where this assumption is severely violated, it may be worthwhile to consider isotonicity with respect to a weaker requirement for distributions to be ordered. An analogous method to IDR under increasing concave and convex stochastic ordering constraints has been introduced by [32]. An extension of EasyUQ in this direction is left for future work.

To estimate conditional CDFs under the given stochastic order constraints from training data of the form (2.1), we define

$$(\hat{F}_{x_1}(y), \dots, \hat{F}_{x_n}(y))' = \arg \min_{\theta \in \mathbb{R}^n \,:\, \theta_i \ge \theta_j \text{ if } x_i \le x_j} \sum_{i=1}^{n} (\theta_i - \mathbb{1}\{y_i \le y\})^2 \tag{2.4}$$

at $y \in \mathbb{R}$. If $x_1 < \cdots < x_n$, then by classical results about isotonic regression,

$$\hat{F}_{x_j}(y) = \min_{k=1,\dots,j} \max_{l=j,\dots,n} \frac{1}{l-k+1} \sum_{i=k}^{l} \mathbb{1}\{y_i \le y\}, \quad j = 1, \dots, n. \tag{2.5}$$

At any single threshold y, the computation can be performed efficiently in $\mathcal{O}(n \log(n))$ complexity with the well-known pool-adjacent-violators (PAV) algorithm. Since the loss function in (2.4) is constant for y in between the unique values $\tilde{y}_1 < \cdots < \tilde{y}_k$ of $y_1, \dots, y_n$, it suffices to compute (2.5) at the unique values, for which efficient recursive algorithms are available [34]. An estimate $\hat{F}_x$ for the conditional CDF at model output $x \in (x_i, x_{i+1})$ is obtained by pointwise linear interpolation in x. For $x \le x_1$ and $x \ge x_n$, we use $\hat{F}_{x_1}$ and $\hat{F}_{x_n}$, respectively. The EasyUQ conditional CDFs

[1] For example, if $F_1 = \mathcal{L}(Y|X = x_1) = \mathcal{N}(\mu_1, \sigma_1^2)$ and $F_2 = \mathcal{L}(Y|X = x_2) = \mathcal{N}(\mu_2, \sigma_2^2)$, where $x_1 \ne x_2$ and $\sigma_1 \ne \sigma_2$, then F_1 and F_2 are incomparable in stochastic order, whence isotonicity is violated. However, if σ_1 and σ_2 are close to each other, the CDFs of F_1 and F_2 cross in the far (left or right) tail only [30, proof of Proposition 1], so violations remain minor.

are step functions that correspond to discrete predictive distributions with mass at (a subset of) the unique values $\tilde{y}_1 < \cdots < \tilde{y}_k$ only.

The IDR approach has desirable properties that make it suitable for uncertainty quantification. By (2.4), the EasyUQ CDFs depend on the order of $x_1, \ldots, x_n$, but not on their values, and hence the solution is invariant under strictly monotone transformations of the model output, except for interpolation choices when $x \notin \{x_1, \ldots, x_n\}$. Furthermore, the EasyUQ distributions are in-sample calibrated [35, Thm. 2]. Importantly, a comparison of the loss function in (2.4) and the definition of the CRPS in (2.2) reveals that EasyUQ minimizes the CRPS over all conditional distributions satisfying the stochastic order constraints. Furthermore, the EasyUQ solution is universal, in the sense that it is simultaneously in-sample optimal with respect to comprehensive classes of proper scoring rules in terms of conditional CDFs or conditional quantiles, such as, e.g., weighted forms of the CRPS with the Lebesgue measure in (2.2) replaced by a general measure [35, Thm. 2]. Other approaches to estimating conditional CDFs, e.g., based on parametric models, nearest neighbors, or kernel regression, do not share the universality property, and estimates change depending on the loss function at hand.

In Figure 1 we illustrate EasyUQ predictive CDFs in the empirical WeatherBench example. Simulation examples, to which we turn now, have the advantage that the true conditional CDFs are available, so we can compare with them. Figure 2 illustrates the construction of the discrete EasyUQ predictive distributions step by step, based on a training archive of the form (2.1) with $n = 500$ simulated from a bivariate distribution, where the model output X is uniform on $(0, 10)$ and the outcome Y satisfies

$$Y \mid X \sim \text{Gamma}(\text{shape} = \sqrt{X}, \text{scale} = \min\{\max\{X, 2\}, 8\}). \tag{2.6}$$

EasyUQ converts the single-valued model output X into conditional predictive CDFs close to the right-skewed true ones. Indeed, IDR and, hence, EasyUQ are asymptotically consistent: As the training archive size n grows, the estimated EasyUQ CDFs converge to the true conditional CDFs [20, 35, 50]. Of particular relevance to EasyUQ is the following recent result [33, Thm. 5.1]: If $x_1, \ldots, x_n$ themselves are not fixed but are predictions from a statistical model that is estimated on the same training data, then IDR is a consistent estimator of the true conditional distributions, subject to mild regularity conditions.

The basic EasyUQ method extends readily to vector-valued model output. If $x_1, \ldots, x_n$ are vectors in a space with a partial order $\preceq$, then the same approach (2.4) applies with the usual inequality $\leq$ replaced by the partial order $\preceq$. This allows more flexibility in the sense that distributions F_x and $F_{x'}$ are allowed to be incomparable in stochastic order if x and x' are incomparable in the partial order. A prominent example concerns ensemble weather forecasts [27, 44, 52], where a numerical model is run several times under distinct conditions, and the partial order $\preceq$ that underlies IDR can be tailored to this setting [35].

To summarize, the basic EasyUQ method provides a data driven, theoretically principled, and fully automated approach to uncertainty quantification that is devoid of any need for implementation choices. Based on training data, EasyUQ converts single-valued model output into calibrated predictive distributions that reflect the uncertainty in the model output and training data, as opposed to tuning intense methods, where uncertainty quantification might reflect implementation decisions and user choices. The EasyUQ predictive solution is invariant under strictly monotone

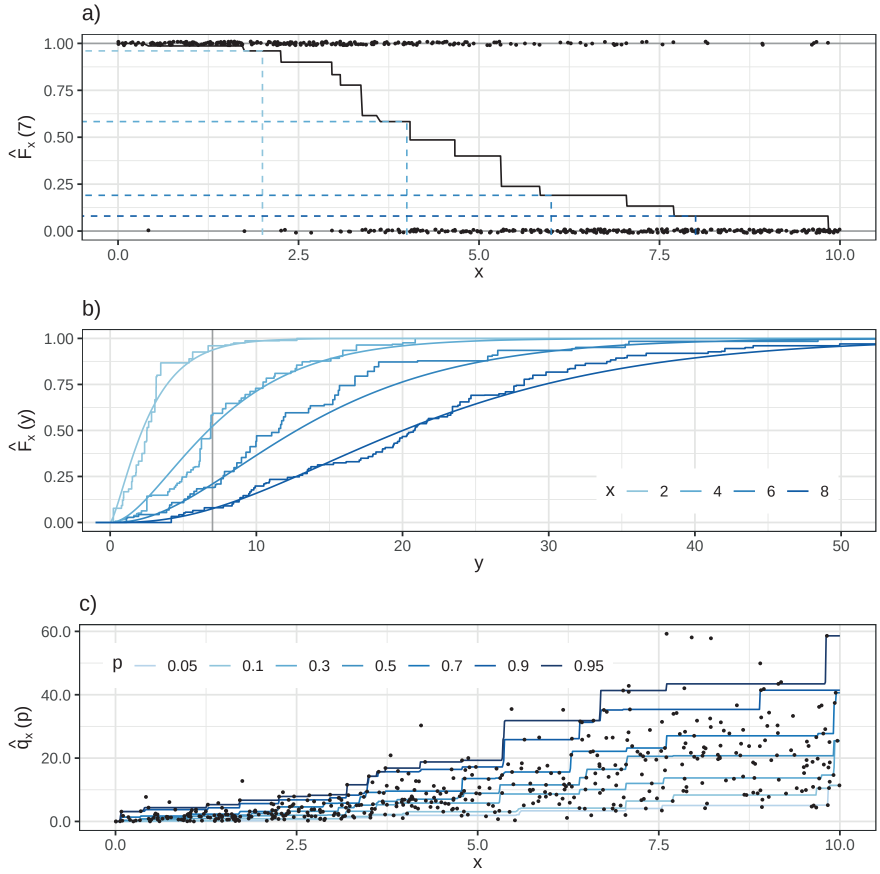

Fig. 2 *Computation of EasyUQ predictive distributions from a training archive of* $n = 500$ *model output–outcome pairs simulated according to* (2.6). (a) *The minimizer* $\hat{F}_x(y)$ *of* (2.5) *at* $y = 7$*, interpolated linearly in* x*. The jiggled dots show the indicators* $\mathbb{1}\{y_i \le y\}$. (b) *EasyUQ conditional CDFs* $\hat{F}_x$ *(step functions) and the respective true conditional CDFs (smooth curves) at selected values of* x*. The vertical line at* $y = 7$ *highlights the values marked in the top panel.* (c) *Training data* (x_i, y_i) *for* $i = 1, \ldots, n$*, and conditional quantile curves* $\hat{q}_x(p)$ *resulting from inversion of the EasyUQ CDFs* $\hat{F}_x$*. The lowest and highest quantile curves (levels* 0.05 *and* 0.95*) together delineate equal-tailed* 90% *prediction intervals.*

transformations of the model output, it is in-sample calibrated, it is in-sample optimal with respect to comprehensive classes of loss functions, and subject to mild conditions it is asymptotically consistent for both output from deterministic models and output from statistical or machine learning models, even when the model is learned on the same data.[2]

[2] By Theorem 2 of Henzi, Ziegel, and Gneiting [35], the fitted EasyUQ distributions are threshold calibrated, i.e., the predicted nonexceedance probabilities equal their empirical counterparts in the training data. Furthermore, the fitted distributions are empirical score minimizers under a large class of proper scoring rules. For discussion of the regularity conditions for asymptotic consistency, we refer the reader to Appendix A in this paper, section 5 in [33], and section 2.4 in [35].

2.3. Illustration on WeatherBench Challenge. In a notable development, WeatherBench [57] introduces a benchmark dataset for the comparison of purely data driven and numerical weather prediction (NWP) model based approaches to weather forecasting. Following up on the illustration in Figure 1, where we consider a grid point at (latitude, longitude) values of (53.4375, 16.875), we now provide background information and quantitative results at grid points worldwide.

Our experiments are based on the setup in WeatherBench and consider forecasts of upper air temperature at a vertical level of 850 hPa pressure. The forecasts are issued twice daily at 00 and 12 Coordinated Universal Time (UTC) at lead times of three and five days ahead. The single-valued HRES forecast is from the high-resolution model operated by the European Centre for Medium-Range Weather Forecasts (ECMWF), which represents the physics and chemistry of the atmosphere and is generally considered to be the leading global NWP model. To reduce the amount of data, WeatherBench regrids the HRES model output and the respective outcomes, which originally are on a 0.25 degree latitude–longitude grid (72×144), to coarser resolution (32×64) via bilinear interpolation. The CNN forecast is also single-valued; it is purely data driven and based on a convolutional neural network (CNN), with trained weights being available in WeatherBench. The single-valued Climatology forecast is the best performing baseline model from WeatherBench; it is obtained as the arithmetic mean of the observed upper air temperature in the training data, stratified by 52 calender weeks.

Conformal prediction (CP) [72, 75] is an increasingly popular, general technique for the construction of predictive distributions from single-valued model output. For a comparison with EasyUQ, we employ CP in the form of the Studentized least squares prediction machine (LSPM) [72, Algorithm 7.2] with the single-valued model output as sole covariate. We consider CP to be a key competitor as it is an output-based method that shares desirable properties of EasyUQ. Specifically, the LSPM supplements a least squares based point prediction of the outcome with a conformal predictive system for uncertainty quantification. Based on training data (x_i, y_i), where $i = 1, \ldots, n-1$, Algorithm 7.2 returns a fuzzy predictive distribution [72, eq. (7.7)] that is defined in terms of quantities $C_1, \ldots, C_{n-1}$. Comparative evaluation requires a crisp predictive distribution for which we use the empirical distribution of $C_1, \ldots, C_{n-1}$, which adheres to the bounds imposed by the fuzzy distribution.[3] For moderate to large training sets and x the value of the model output at hand, C_i typically is very close to $\hat{y} + y_i - \hat{y}_i$, where $\hat{y}$ and $\hat{y}_i$ are least squares point predictions based on x and x_i, respectively [72, sect. 7.3.4].

Finally, we consider the state-of-the-art approach to uncertainty quantification in weather prediction, namely, ensemble forecasts [27, 44, 52], which are input-based methods. Specifically, we use the world leading ECMWF Integrated Forecast System (IFS; see https://www.ecmwf.int/en/forecasts), which comprises 51 NWP runs, namely, a control run and 50 perturbed members [49]. The control run is based on the best estimate of the initial state of the atmosphere, and the perturbed members start from slightly different states that represent uncertainty. Even a single NWP

[3] Here and in section 3.4, we adopt the convention in Vovk, Gammerman, and Shafer [72, sect. 7.2] and assume that the size of the training set is $n-1$, rather than n, to allow for direct references to material therein. The respective crisp CDF is given by $F(y) = i/n$ for $y \in (C_{(i)}, C_{(i+1)})$ and $i = 0, 1, \ldots, n-1$, and $F(y) = i''/n$ for $y = C_{(i)}$ and $i = 1, \ldots, n-1$, where $C_{(0)} = -\infty$, $C_{(1)} \leq \cdots \leq C_{(n-1)}$ are the order statistics of $C_1, \ldots, C_{n-1}$, $C_{(n)} = \infty$, and $i'' = \max\{j : C_{(j)} = C_{(i)}\}$. For related discussion and alternative choices of a crisp CDF that is compatible with the fuzzy CDF, see section 2 of Boström, Johansson, and Löfström [7] and section 5 of Vovk et al. [74].

Table 1 *Predictive performance in terms of mean CRPS for WeatherBench forecasts of upper air temperature at lead times of three and five days, in degrees Kelvin. The evaluation period comprises calendar years* 2017 *and* 2018*. CP and EasyUQ generate predictive CDFs that are fitted at each grid point individually, based on training data from* 2010 *through* 2016*. Forecasts are issued twice daily, and scores are averaged over* 32×64 *grid points, for a total of* 2,990,080 *forecast cases.*

Forecast		CRPS	
Type	Method	Three Days	Five Days
Single-valued	Climatology	2.904	2.904
	CNN	2.365	2.782
	HRES	0.998	1.543
Distributional	CP on Climatology	2.055	2.055
	CP on CNN	1.673	1.955
	CP on HRES	0.731	1.123
Distributional	EasyUQ on Climatology	2.038	2.038
	EasyUQ on CNN	1.671	1.949
	EasyUQ on HRES	0.736	1.122
Distributional	ECMWF Ensemble	0.696	0.998

model run, such as the HRES run, is computationally very expensive, and computing power is the limiting factor to improving model resolution. Despite having coarser resolution, an ensemble typically requires 10 to 15 times more computing power than a single run [3]. In contrast, the implementation of the output-based CP and EasyUQ methods is fast, with hardly any resources needed beyond a single NWP model run.

To compare CP and EasyUQ predictive CDFs to the respective single-valued forecasts, we use the CRPS from (2.2) and recall that for single-valued forecasts the mean CRPS reduces to the mean absolute error (MAE). As evaluation period, we take calendar years 2017 and 2018; for estimating the CP and EasyUQ predictive distributions, we use training data from calendar years 2010 through 2016 and proceed grid point by grid point. The corresponding results are provided in Table 1. Not surprisingly, the ECMWF ensemble forecast has the lowest mean CRPS. However, CP and EasyUQ based on the HRES model output result in promising CRPS values, even though the methods require considerably less computing time and resources.

The CP and EasyUQ predictive distributions show nearly identical predictive performance. To understand this behavior, we note that in the case of temperature, Gaussian predictive distributions with fixed variance are typically very adequate (see, e.g., [29, Table 3]). In this light, key requirements of CP in the form of the LSPM (namely, fixed spread and fixed shape of the predictive distributions) and EasyUQ (namely, isotonicity) are reasonably met. While EasyUQ generates predictive distributions that vary in spread and shape, the variations remain modest (Figure 1(c)–(f)), and the CP distributions, which essentially are translates of each other, are competitive.

The subsequent case study turns to a weather variable that is not covered by the WeatherBench challenge, but which serves to illuminate and highlight differences between the CP and EasyUQ techniques.

2.4. Illustration on Precipitation Forecasts. Precipitation accumulation is generally considered the "most difficult weather variable to forecast" [19]. Indeed, the uncertainty quantification for deterministic forecasts of precipitation is more challenging than for temperature, since precipitation accumulation follows a mixture distribution with a point mass at zero—for no precipitation—and a continuous part on the positive real numbers. Applying CP without corrections is bound to transfer mass to negative values of precipitation accumulation. Taking advantage of knowledge about the outcome distribution, a natural remedy is to censor at zero and use the CDF

$$G(y) = \begin{cases} 0, & y < 0, \\ F(y), & y \geq 0, \end{cases}$$

in lieu of F.[4] In contrast, the EasyUQ predictive distributions reflect the nonnegativity of the outcomes in the training data, without any need for adaptation.

We now investigate the performance of CP and EasyUQ within the experimental setup from Henzi, Ziegel, and Gneiting [35], taking forecasts and observations of 24-hour accumulated precipitation from 6 January 2007 through 1 January 2017 at Frankfurt Airport, Germany. Just as in the WeatherBench example, we consider a weekly climatology, the HRES forecast, and the 51 member NWP ensemble from ECMWF. The weekly climatology is computed over the period 2007 to 2014, which is the same period that is used for CP and EasyUQ training. The evaluation period comprises calendar years 2015 and 2016. Table 2 shows the mean CRPS over the evaluation period for the various types of forecasts at lead times from one to five days. Evidently, the climatological forecasts, along with their scores, do not depend on the lead time. In contrast to the WeatherBench temperature example, EasyUQ outperforms CP for both Climatology and the HRES model output, and at all lead times. While censoring improves the distributional forecasts from CP, the performance gap to EasyUQ remains pronounced. EasyUQ on the HRES model output even outperforms the raw ECMWF ensemble at lead times of one and two days.[5]

Figure 3 provides a graphical comparison of CP on HRES, Censored CP on HRES, EasyUQ on HRES, and ECMWF ensemble forecasts at small ($x = 0.38$), moderate ($x = 3.40$), and large ($x = 11.93$) values of the HRES model output x. We see that the CP predictive distributions are essentially translates of each other, with mass potentially being transferred to negative values of precipitation accumulation, and censoring shifting any such mass to zero. In contrast, the ECMWF ensemble and EasyUQ distributions do not have mass at negative values, and they vary in shape

[4] In our experiments, we train without consideration of censoring, and we censor at zero ex post. For a nonnegative outcome, such a procedure guarantees improvement in the technical sense that $\mathrm{CRPS}(G, y) \leq \mathrm{CRPS}(F, y)$ for all $y \geq 0$. Alternatively, one might take censoring into account during training. However, methods of this latter type are more complex to implement, and improvements in CRPS cannot be guaranteed out-of-sample.

[5] This is largely due to the fact that gridded ensemble predictions are compared against station observations. To counter these effects, the ensemble forecast itself can be subjected to statistical postprocessing, i.e., the application of statistical methods to correct for biases and dispersion errors [29, 55]. Parametric methods based on distributional regression [48, 62] model the distribution of precipitation accumulation with censored logistic or censored generalized extreme value distributions. An alternative approach is taken in Bayesian model averaging [67], which posits separate parametric forms for the probability of zero precipitation and the density at positive amounts. Evidently, discrete-continuous mixture distributions considerably complicate model building and estimation, and great efforts are made to find suitable parametric families for specific weather variables. For a detailed performance comparison on the data on hand, see Figure 5 of Henzi, Ziegel, and Gneiting [35], whose study also includes versions of IDR with multivariate covariates derived from the full ECMWF ensemble and suitable partial orders on them, an option alluded to at the end of section 2.2. These yield improvements compared to both the raw ensemble forecast and EasyUQ on HRES, at the price of higher conceptual complexity, higher computational costs, and the need for access to the full ensemble, rather than single-valued HRES model output.

Table 2 *Predictive performance in terms of mean CRPS for forecasts of daily precipitation accumulation at Frankfurt Airport at lead times from one to five days, in millimeters. CP and EasyUQ generate predictive CDFs based on training data from* 2007 *through* 2014. *The evaluation period comprises calendar years* 2015 *and* 2016.

Forecast		CRPS				
Type	Method	1 Day	2 Days	3 Days	4 Days	5 Days
Single-valued	Climatology	2.187	2.187	2.187	2.187	2.187
	HRES	1.125	1.294	1.412	1.478	1.686
Distributional	CP on Climatology	1.382	1.382	1.382	1.382	1.382
	CP on HRES	0.886	0.966	1.063	1.081	1.129
	Censored CP on Climatology	1.324	1.324	1.324	1.324	1.324
	Censored CP on HRES	0.850	0.925	1.031	1.050	1.100
Distributional	EasyUQ on Climatology	1.242	1.242	1.242	1.242	1.242
	EasyUQ on HRES	0.732	0.803	0.876	0.945	1.001
Distributional	ECMWF Ensemble	0.752	0.847	0.856	0.918	0.981

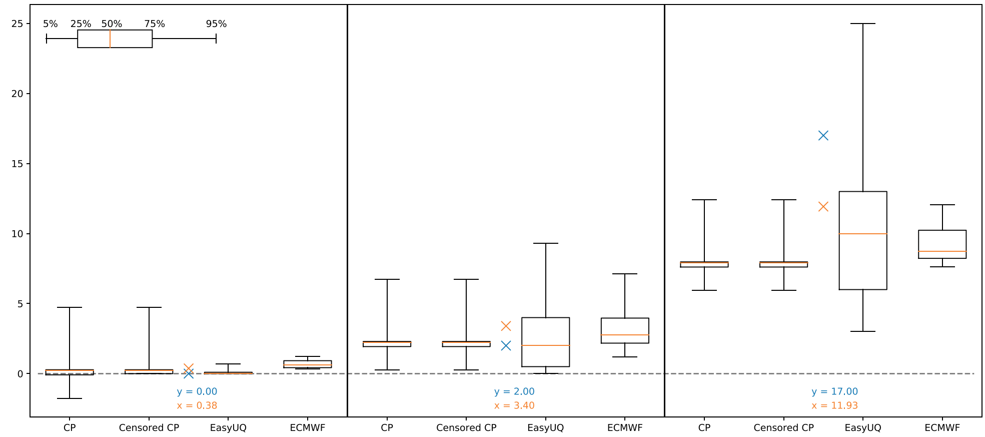

Fig. 3 *One-day ahead forecasts of daily precipitation accumulation at Frankfurt Airport valid* 23 *January* 2015 *(left, HRES model output x equal to* 0.38*, as indicated by the blue cross),* 14 *January* 2015 *(middle, $x = 3.40$), and* 21 *February* 2016 *(right, $x = 11.93$), in millimeters. The predictive distributions for CP on HRES, Censored CP on HRES, EasyUQ on HRES, and ECMWF ensemble techniques are shown. The observed precipitation accumulation was at $y = 0$, $y = 2$, and $y = 17$ millimeters, respectively.*

and scale. However, while the ECMWF ensemble tends to show forecast distributions that are too narrow, as is frequently observed in practice [27] and illustrated by the right-hand example, the EasyUQ distributions, which are based on the single-valued HRES forecast only, show what appears to be adequate spread. Remarkably, and unlike any other method that we are aware of, EasyUQ achieves this desirable performance in its very basic form, without any need for implementation decisions, parameter tuning, or other forms of adaptation and intervention.

3. Smooth EasyUQ. EasyUQ provides discrete predictive distributions with positive probability mass at the outcomes from the training archive. For genuinely discrete outcomes, the variable of interest attains a small number of unique values only, which is a desirable property. For genuinely continuous variables, it is preferable to use continuous predictive distributions. We now describe the Smooth EasyUQ technique, which turns the discrete basic EasyUQ CDFs into continuous Smooth EasyUQ CDFs with Lebesgue densities, while preserving isotonicity. To achieve this, Smooth EasyUQ applies kernel smoothing, which requires implementation choices, unlike basic EasyUQ which does not require any tuning. However, we provide default options.

3.1. Smooth EasyUQ: Kernel Smoothing under Isotonicity Preservation. Our goal is to transform the discrete basic EasyUQ CDFs $\hat{F}_x$ from (2.5) into smooth predictive CDFs $\check{F}_x$ that admit Lebesgue densities $\check{f}_x$ without abandoning the order relations honored by the basic technique. To this end, we define the Smooth EasyUQ CDF as

$$\check{F}_x(y) = \int_{-\infty}^{\infty} \hat{F}_x(t)\, K_h(y-t)\,\mathrm{d}t, \tag{3.1}$$

where $K_h(u) = (1/h)\,\kappa(u/h)$ for a smooth probability density function or kernel κ, such as a standardized Gaussian or Student-t density, with bandwidth $h > 0$. While the convolution approach in (3.1) is perfectly general for the smoothing of CDFs, we henceforth focus the presentation on EasyUQ. The choice of the kernel and the bandwidth are critical, and we tend to their selection in the next section, where we introduce multiple one-fit grid search as a computationally efficient alternative to cross-validation.

For now, recall that $\hat{F}_x(y)$ from (2.5) is a step function with possible jumps at the unique values $\tilde{y}_1 < \cdots < \tilde{y}_k$ of the outcomes $y_1, \dots, y_n$ in the training set. Hence, we can write (3.1) as

$$\check{F}_x(y) = \sum_{j=1}^{k} \hat{F}_x(\tilde{y}_j) \int_{\tilde{y}_j}^{\tilde{y}_{j+1}} K_h(y-t)\,\mathrm{d}t,$$

where $\tilde{y}_{k+1} = \infty$. To compute the density $\check{f}_x = \check{F}_x'$, we set $\tilde{y}_0 = -\infty$, note that $\hat{F}_x$ assigns mass $w_j(x) = \hat{F}_x(\tilde{y}_j) - \hat{F}_x(\tilde{y}_{j-1})$ to $\tilde{y}_j$, and find that

$$\check{f}_x(y) = \sum_{j=1}^{k} \hat{F}_x(\tilde{y}_j)\,[K_h(y-\tilde{y}_j) - K_h(y-\tilde{y}_{j+1})] = \sum_{j=1}^{k} w_j(x)\,K_h(y-\tilde{y}_j). \tag{3.2}$$

In other words, the Smooth EasyUQ density $\check{f}_x$ from (3.2) arises as a kernel smoothing of the discrete probability measure that corresponds to $\hat{F}_x$ and assigns weight $w_j(x)$ to $\tilde{y}_j$. Consequently, $\check{f}_x$ is a probability density function, $\check{F}_x$ is a proper CDF, and, notably, Smooth EasyUQ preserves the stochastic ordering of the basic EasyUQ estimates. In Figure 4 we illustrate the interpretation of the Smooth EasyUQ density as a kernel smoothing of the EasyUQ point masses $w_j(x)$ on the WeatherBench example.

Subject to mild regularity conditions, the asymptotic consistency of EasyUQ carries over to Smooth EasyUQ. To demonstrate this, we prove a general consistency theorem for estimates of conditional CDFs in Appendix A. Here, we sketch how the result applies in the special case of Smooth EasyUQ. Specifically, let $\check{F}_{x;n}$ denote the Smooth EasyUQ estimator from (3.1), where the basic estimate $\hat{F}_x$ is trained on a sample of size n from a population with true conditional CDFs F_x that are Hölder continuous with constants $\alpha \in (0, 1]$ and $L > 0$. Suppose that the function K_h in (3.1)

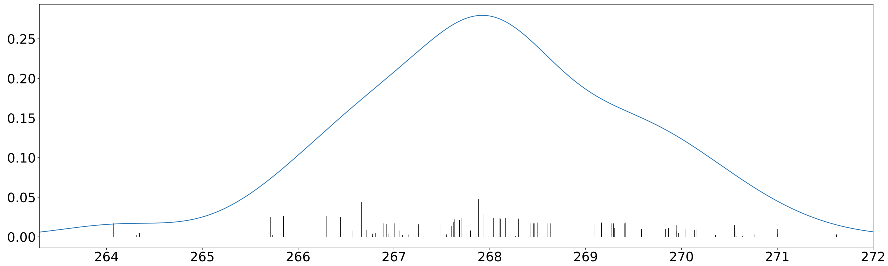

Fig. 4 *Smooth EasyUQ predictive density* (3.2) *in the WeatherBench example from Figure* 1(f) *at HRES model output x equal to* 268 *degrees Kelvin. The vertical bars show the weights $w_1(x), \ldots, w_k(x)$ that the discrete EasyUQ distribution $\hat{F}_x$ assigns to the unique values $\tilde{y}_1 < \cdots < \tilde{y}_k$ of the outcomes in the training set, where $k = 76$.*

uses a kernel κ with a finite absolute moment $c_{\kappa,\alpha}$ of order α, and allow the bandwidth $h_n > 0$ to vary with the sample size. Crucially, we assume that the sup-error of the basic estimate $\hat{F}_x$ at sample size n is upper bounded by ε_n in the asymptotic sense specified by (A.2); a choice of $\varepsilon_n \sim (\log(n)/n)^{1/3}$ applies to EasyUQ if on an interval (a, b) the covariate values $x_1, \ldots, x_n$ are sufficiently dense and the conditional CDFs are Lipschitz continuous in x. Then Theorem A.3 implies that, for some sequence of $\delta_n > 0$ converging to zero,

$$\lim_{n\to\infty} \mathbb{P}\left(\sup_{y\in\mathbb{R},\, x\in(a+\delta_n, b-\delta_n)} \left|\check{F}_{x;n}(y) - F_x(y)\right| \geq \varepsilon_n + L c_{\kappa,\alpha} h_n^\alpha \right) = 0. \tag{3.3}$$

If $h_n^\alpha = \mathcal{O}(\varepsilon_n)$, the Smooth EasyUQ CDFs converge with the same rate as the basic EasyUQ CDFs. For details, proofs, and discussion, see Appendix A. In a nutshell, smoothness conditions on the true conditional CDFs are essential and unproblematic, as one should not be replacing the basic version of EasyUQ by Smooth EasyUQ in practice, unless the subject matter indicates an absolutely continuous distribution of the outcome. For instance, in the precipitation forecasting example from section 2.4, basic EasyUQ outperforms Smooth EasyUQ and a censored version of it at all lead times; cf. Tables 2 and 4.

3.2. Choice of Kernel and Bandwidth: Multiple One-Fit Grid Search. In order to compute the Smooth EasyUQ density $\check{f}_x$ from (3.2), one needs to choose a kernel κ and a bandwidth $h > 0$ to yield a mixture of translates of the density $K_h(u) = (1/h)\,\kappa(u/h)$. While there is a rich literature on bandwidth selection for kernel density estimation and kernel regression (see, e.g., [39, 66]), caution is needed when applying established approaches to Smooth EasyUQ, due to the fact that smoothing is applied to estimated conditional CDFs rather than raw data.

Furthermore, while the extant literature focuses on bandwidth selection for a fixed kernel, approaches of this type are restrictive for our purposes. The Smooth EasyUQ density from (3.2) inherits the tail behavior of the kernel κ, and so the properties of the kernel are of critical importance to the quality of the uncertainty quantification in the tails of the conditional distributions. To allow for distinct tail behavior, we use the Student-t family and set $K_{\nu,h}(u) = (1/h)\,\kappa_\nu(u/h)$, where

$$\kappa_\nu(y) = \frac{\Gamma((\nu+1)/2)}{(\pi\nu)^{1/2}\,\Gamma(\nu/2)} \left(1 + \frac{y^2}{\nu}\right)^{-(\nu+1)/2} \tag{3.4}$$

is a standardized Student-t probability density function with $\nu > 0$ degrees of freedom. It is well known that the Student-t distribution has a finite first moment if $\nu > 1$ and a finite variance if $\nu > 2$. In the limit as $\nu \to \infty$, we find that $\kappa_\nu(y) \to \kappa_\infty(y)$ uniformly in y, where $\kappa_\infty(y) = (2\pi)^{-1/2} \exp(-y^2/2)$ is the standard Gaussian density function, so the ubiquitous Gaussian kernel emerges as a limit case in (3.4).

Turning to the choice of the tail parameter $\nu \in (0, \infty]$ and the bandwidth $h > 0$, we begin by discussing the latter. A popular approach for bandwidth selection, in both kernel regression and kernel density estimation, is leave-one-out cross-validation. Here the target criterion in terms of the bandwidth is

$$\text{CV}(h) = \frac{1}{n} \sum_{i=1}^{n} \text{S}(\check{F}_{x_i,-i,h}, y_i), \tag{3.5}$$

where S is a proper scoring rule and $\check{F}_{x_i,-i,h}$ is the Smooth EasyUQ CDF with covariate x_i and bandwidth h, estimated with all data from (2.1) except for the ith instance. The optimization of the target criterion (3.5) uses either the CRPS as loss function S, as is (implicitly) suggested for the estimation of conditional CDFs and quantile functions (see, e.g., [8, p. 801] and [45, p. 58]) and yielding a target that is asymptotically equivalent to the integrated mean squared error [35, sect. S4], or the LogS, as is proposed for ensemble smoothing [10]. We take the latter as the default choice, since the LogS is much more sensitive to the choice of the bandwidth h than the more robust CRPS.

However, there are a number of caveats. Empirical data are typically discrete to some extent and might contain ties in the response variable, such as in the setting of Figure 4, where there are only $m = 76$ unique values among the outcomes $y_1, \ldots, y_n$, even though $\check{f}_x$ is estimated from a training archive of size $n = 5{,}114$. In such cases, the optimal cross-validation bandwidth under the LogS may degenerate to $h = 0$, a problem that is also known in density estimation [66, pp. 51–55], in the estimation of Student-t regression models [21] and, in related form, in performance evaluation for forecast contests [40, 54]. Another issue is that leave-one-out cross-validation is computationally expensive, as for each value of h it requires the computation of n distinct IDR solutions. While a potential remedy is to remove a higher percentage of observations in each cross-validation step, we use a considerably faster approach, which we term one-fit grid search, that addresses both issues simultaneously.

One-fit grid search avoids repeated fits of IDR and computes EasyUQ only once, namely, on the full sample from (2.1). Specifically, given any fixed kernel κ, one-fit grid search finds the optimal bandwidth h in terms of the target criterion

$$\text{OF}(h) = \frac{1}{n} \sum_{i=1}^{n} \text{S}(\bar{F}_{x_i,-i,h}, y_i), \tag{3.6}$$

where $\bar{F}_{x_i,-i,h}$ removes the unique value $\tilde{y}_j = y_i$ from the support of $\check{F}_{x_i}$ in (3.1) by setting $w_j(x)$ in (3.2) to zero and rescaling the remaining weights. We choose the LogS as the default option for the loss function S in the one-fit criterion (3.6), and we use Brent's algorithm [9] for optimization. Effectively, one-fit grid search is a fast approximation to cross-validation, and when n is small, leave-one-out cross-validation and the original criterion in (3.5) can be used instead, of course. To choose a Student-t kernel, we repeat the procedure, i.e., we consider values of $\nu \in \{2, 3, 4, 5, 10, 20, \infty\}$ in (3.4), with $\nu = \infty$ yielding the Gaussian limit, apply one-fit grid search for each of these values to find the respective optimal bandwidth h, and select the combination

Table 3 *Predictive performance in terms of mean LogS and mean CRPS for WeatherBench density forecasts of upper air temperature at lead times of three and five days, in degrees Kelvin. The evaluation period comprises calendar years* 2017 *and* 2018*. The Single Gaussian, Smooth CP, and Smooth EasyUQ methods are trained at each grid point individually, based on data from* 2010 *through* 2016*. Forecasts are issued twice daily, and scores are averaged over* 32×64 *grid points, for a total of* 2,990,080 *forecast cases.*

Density Forecast	LogS		CRPS	
Days Ahead	Three	Five	Three	Five
Single Gaussian on Climatology	2.578	2.578	2.060	2.060
Single Gaussian on CNN	2.413	2.553	1.696	1.983
Single Gaussian on HRES	1.694	2.073	0.748	1.153
Smooth CP on Climatology	2.562	2.562	2.059	2.059
Smooth CP on CNN	2.384	2.519	1.672	1.952
Smooth CP on HRES	1.627	2.007	0.732	1.123
Smooth EasyUQ on Climatology	2.540	2.540	2.043	2.043
Smooth EasyUQ on CNN	2.375	2.509	1.667	1.945
Smooth EasyUQ on HRES	1.640	2.006	0.736	1.122
Smoothed ECMWF Ensemble	1.503	1.824	0.685	0.990

of ν and h for which the target criterion (3.6) is smallest overall. While being highly effective in our experience, multiple one-fit grid search is a crude approach, and we encourage further development.

3.3. Illustration on Temperature and Precipitation Forecast Examples.

For an initial illustration, we return to the WeatherBench challenge and the Smooth EasyUQ densities in Figures 1(f) and 4, where $n = 5{,}114$ and $m = 76$, and multiple one-fit grid search with respect to the LogS yields parameter values $\nu = \infty$ and $h = 0.60$ in the kernel density (3.4). Considering the $32 \times 64 = 2{,}048$ grid points in WeatherBench and predictions three days ahead, the value of ν selected the most frequently for Smooth EasyUQ on the HRES model output, namely, 619 times, is $\nu = 10$, with a median choice of $h = 0.49$. For Smooth EasyUQ on Climatology and CNN, $\nu = \infty$ was most frequently selected, namely, 1,391 and 1,361 times with median choices of $h = 0.85$ and $h = 1.04$, respectively.

A very simple and frequently used reference method for converting single-valued model output into a predictive density is the Single Gaussian technique [17]. It issues a Gaussian distribution with mean equal to the single-valued model output and a constant variance that is optimal with respect to the mean LogS on a training set, which here we take to be the same as for EasyUQ. Evidently, both Smooth EasyUQ and the Single Gaussian technique could be trained in terms of the CRPS as well. We also compare to the Smooth CP technique, which converts the discrete CP distributions to densities as described in the next section.

In Table 3, we evaluate Smooth EasyUQ, Smooth CP, and Single Gaussian density temperature forecasts in the WeatherBench setting. For evaluation, we use both the CRPS and the LogS. Throughout, Smooth EasyUQ and Smooth CP outperform the Single Gaussian method, though they do not match the performance of the smoothed ECMWF ensemble forecast, which we construct as follows. Let $\tilde{z}_1 < \cdots < \tilde{z}_k$ be the unique values of the ensemble members $z_1, \ldots, z_l$ of an ensemble forecast of size l. The smoothed ensemble CDF is then of the form (3.1) with mass $w_j = \frac{1}{l} \sum_{i=1}^{l} \mathbb{1}(z_i =$

Table 4 *Predictive performance in terms of mean CRPS for density forecasts of daily precipitation accumulation at Frankfurt Airport at lead times from one to five days, in millimeters. CP and EasyUQ generate predictive CDFs based on training data from* 2007 *through* 2014. *The evaluation period comprises calendar years* 2015 *and* 2016.

Density Forecast	1 Day	2 Days	3 Days	4 Days	5 Days
Single Gaussian on HRES	1.244	1.380	1.547	1.577	1.724
Censored Single Gaussian on HRES	1.013	1.145	1.266	1.276	1.401
Smooth CP on HRES	0.886	0.971	1.064	1.087	1.132
Censored Smooth CP on HRES	0.849	0.928	1.028	1.052	1.098
Smooth EasyUQ on HRES	0.760	0.828	0.901	0.968	1.033
Censored Smooth EasyUQ on HRES	0.745	0.817	0.893	0.960	1.016
Smoothed ECMWF Ensemble	0.762	0.855	0.863	0.924	0.986
Censored Smoothed ECMWF Ensemble	0.750	0.850	0.860	0.921	0.984

$\tilde{z}_j$) for $j = 1, \ldots, k$. Interestingly, this is the same as Bröcker–Smith smoothing of ensemble forecasts [10, relations (19)–(21)] with parameters $a = 1$ and $r_1 = r_2 = s_2 = 0$ being fixed. However, while Bröcker and Smith use a Gaussian kernel and optimize the bandwidth parameter only, we take a more flexible approach and consider values of $\nu \in \{2, 3, 4, 5, 10, 20, \infty\}$ for a Student-t kernel to find the optimal ν and bandwidth h in terms of the LogS. Across the 2,048 grid points, the most frequent choice is $\nu = 5$, namely, 743 times, with a median bandwidth value of $h = 0.50$.

While smoothing is warranted for temperature forecasts, it is problematic for forecasts of precipitation accumulation due to the nonnegativity of the outcome and the point mass at zero. Indeed, due to the kernel smoothing, the Smooth EasyUQ and smoothed ECMWF ensemble densities have mass on the negative halfaxis, unlike the discrete (basic) EasyUQ and (raw) ECMWF distributions, which are concentrated on the nonnegative halfaxis. Nonetheless, Table 4 compares the predictive performance of Single Gaussian, Smooth CP, Smooth EasyUQ, and smoothed ECMWF ensemble forecasts in the setting of section 2.4, in both the original and the censored variants. The results mirror the findings in Table 2 in that censoring yields improvement and EasyUQ outperforms CP, whereas CP outperforms the Single Gaussian technique.

3.4. Computational Considerations. We add a brief discussion of the computational complexity of output-based methods for uncertainty quantification. For this comparison, we utilize the setting of Algorithm 7.2 in Vovk, Gammerman, and Shafer [72], which requires predictive distributions for m new values of x based on a training set of size $n - 1$ with instances $(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$. We report upper estimates of the computational complexity for the Single Gaussian technique, CP, and EasyUQ, considering both training (i.e., initial operations on the training data only) and inference (i.e., operations to be repeated for each new value). For the simplistic Single Gaussian technique, training requires $\mathcal{O}(n)$ operations and inference is straightforward.

For EasyUQ, the main effort lies in training, where the complexity is upper bounded by $\mathcal{O}(n^2)$ operations [34]. Training the EasyUQ CDFs only on a fixed grid of ordinates guarantees a cost reduction to $\mathcal{O}(n \log n)$ operations, and Henzi, Ziegel, and Gneiting [35] describe approaches based on subset aggregation that reduce the computational burden for estimation. That said, the numerical experiments in our paper use the standard implementation throughout, without exception. For inference, each new value of x requires the determination of its position within the unique values across $x_1, \dots, x_{n-1}$, followed by interpolation of the trained EasyUQ CDFs at the predecessor and successor values, at up to $\mathcal{O}(mn)$ operations.

For CP in the form of the Studentized LSPM [72, Alg. 7.2] essentially no training is required, but inference incurs $\mathcal{O}(mn^2)$ operations. Residual-based approximations to CP, which are instances of split conformal predictive systems [72, sect. 7.3.4], [73], are much faster, shift the bulk of the cost to training at $\mathcal{O}(n)$ operations, and yield nearly identical predictive performance to CP in our experience, except when training sets are small.

For both CP and EasyUQ, we have implemented smoothing in ways that avoid cross-validation and honor the aforementioned bounds. Smooth EasyUQ uses one-fit grid search as developed in this paper. To generate the Smooth CP densities, we use kernel smoothing with a Gaussian kernel and bandwidth chosen according to Silverman's rule of thumb [66], applied to the quantities $C_1, \dots, C_{n-1}$ that arise for each new instance separately.

In the aforementioned experiments, we generally found the computational cost of EasyUQ to be nested in between the costs of CP and residual-based approximations to CP.[6] Compared to the enormous effort of running the HRES model or even the input-based ECMWF ensemble method, which require the operational use of supercomputers, run times and computational costs for the output-based Single Gaussian, CP, and EasyUQ techniques are negligible.

4. EasyUQ and Neural Networks.

Neural networks and deep learning techniques have enabled unprecedented progress in predictive science. However, as they "can struggle to produce accurate uncertainties estimates ... there is active research directed toward this end" [2, p. 67], which has intensified in recent years [1, 14, 18, 22, 37, 41, 42, 46, 75]. We now discuss how EasyUQ and Smooth EasyUQ can be used to yield accurate uncertainty statements from neural networks. Evidently, our methods apply in the ways described thus far, where single-valued model output is treated as given and fixed, with subsequent uncertainty quantification via EasyUQ or Smooth EasyUQ being a completely separate add-on, as illustrated using our temperature and precipitation examples. In the context of neural networks, this means that the network parameters are optimized to yield single-valued output, and only then is EasyUQ applied. We now describe a more elaborate approach where we integrate our methods within the typical workflow of neural network training and evaluation.

4.1. Integrating EasyUQ into the Workflow of Neural Network Learning and Hyperparameter Optimization.

Neural networks and associated methods for uncertainty quantification are developed and evaluated in well-designed workflows that involve multiple splits of the available data into training, validation, and test sets. For each split, the training set is used to learn basic neural network parameters, the

[6] To provide intuition into computation times, we report mean run times for the Single Gaussian technique, CP, and EasyUQ applied to the HRES forecast in the setting of Table 2, where the training set is of size 2,896 and the evaluation set of size 721. The mean run time averaged over the five lead times is 0.005 seconds for the Single Gaussian technique, 0.45 seconds for CP, and 0.085 seconds for EasyUQ. We note that the computing time for CP on a CPU is 33.64 seconds, but can be reduced to 0.45 seconds on a GPU. Evidently, the comparison faces the usual challenges, given that execution times depend on factors including but not limited to hardware architecture, disk speed, memory availability, and the programming language and compiler used. Specific to the situation at hand, we use code in Python, R, and C++, run some functions on a GPU and others on a CPU, and it is unlikely that every one of our implementations, which typically are based on packages, has been coded in the most efficient way.

Algorithm 4.1 Integration of Smooth EasyUQ into the workflow of neural network training and hyperparameter tuning. The procedure returns the mean score of the Smooth EasyUQ predictions across data splits.

```
for split in mysplit do
    separate data into training set, validation set, and test set
    for hyperpar in myhyperpar do
        learn neural network with hyperpar on training set
        use neural network output to fit basic EasyUQ on training set
        use moderated grid search to select EasyUQ parameters $\nu$ and $h$
        save selected $(\nu, h)$ and mean score on validation set
    end for
    select best hyperpar and associated $(\nu, h)$, based on smallest mean score
    relearn network with best hyperpar on combined training and validation sets
    use relearned neural network output to refit basic EasyUQ on combined
            training and validation sets
    use Smooth EasyUQ based on refitted EasyUQ with best $(\nu, h)$ for predictions
            on test set
    save scores on test set
end for
return mean score across splits
```

validation set is used to tune hyperparameters, and the test set is used for out-of-sample evaluation. Scores are then averaged over the tests sets across the splits, and methods with low mean score are preferred.

Algorithm 4.1 describes how Smooth EasyUQ can be implemented within this typical workflow of neural network learning and hyperparameter tuning. In a nutshell, we treat the kernel parameters for Smooth EasyUQ, namely, the Student-t parameter ν and the bandwidth h, as supplemental hyperparameters, and we optimize over both the neural network hyperparameters and the kernel parameters. As the evaluation occurs out-of-sample, the issues associated with the choice of the kernel parameters discussed in section 3.2 are mitigated, unless a dataset is genuinely discrete, in which case even out-of-sample estimates of the bandwidth h can degenerate to zero, thereby indicating that smoothing is ill-advised. To handle even such ill-advised cases, we use a procedure that we call moderated grid search [76]. Specifically, we first check whether using $\nu = 2$ or a Gaussian kernel results in a degeneration of the optimal bandwidth h to zero, and if so, we use the latter with bandwidth chosen according to Silverman's rule of thumb [66]. Otherwise, we consider values of $\nu \in \{2, 3, 4, 5, 10, 20, \infty\}$ in (3.4), with $\nu = \infty$ yielding the Gaussian limit. For each value of ν, we use Brent's method [9] to optimize the log score with respect to the bandwidth h on the validation set and choose the optimal combination of ν and h. Once network hyperparameters and kernel parameters have been determined, we relearn the neural network on the combined training and validation sets using the optimized hyperparameters and apply EasyUQ on the relearned single-valued neural network output. Finally, we apply Smooth EasyUQ based on the relearned EasyUQ solution and the selected kernel parameters to yield density forecasts on the test set.

While optimization could be performed with respect to the CRPS, the LogS, or any other suitable proper scoring rule, we follow the machine learning literature, where benchmarking is typically done in terms of the LogS. The CRPS serves as

Table 5 *Characteristics of datasets and predictive performance for competing methods of uncertainty quantification in regression problems, in terms of the mean logarithmic score (LogS) in a popular benchmark setting from machine learning* [18, 22, 36, 42]*. For each dataset, we show size, number of unique outcomes, and dimension of the input (covariate or feature) space. Italics indicate discrete datasets where the number of unique outcomes is small. For each method, we report the mean LogS from the reference stated, with further details provided in section* 4.2*. For each of the lower three blocks of comparable methods, the best (lowest) mean score is set in green. Two scores are numerically infinite; missing scores are marked NA.*

Method / Dataset	Boston	Concrete	Energy	Kin8nm	*Naval*	Power	Protein	*Wine*	Yacht	*Year*
Size	506	1,030	768	8,192	*11,934*	9,568	45,730	*1,599*	308	*515,345*
Unique Outcomes	229	845	586	8,191	*51*	4,836	15,903	*6*	258	*89*
Dimension Input Space	13	8	8	8	*16*	4	9	*11*	6	*90*
Distributional Forest [18]	2.67	3.38	1.53	−0.40	−4.84	2.68	2.59	1.05	2.94	NA
GAMLSS [18]	2.73	3.24	1.24	−0.26	−5.56	2.86	3.00	0.97	0.80	NA
GP Regression [18]	2.37	3.03	0.66	−1.11	−4.98	2.81	2.89	0.95	0.10	NA
NGBoost [18]	2.43	3.04	0.60	−0.49	−5.34	2.79	2.81	0.91	0.20	3.43
40 Deep Ensembles [42]	2.41	3.06	1.38	−1.20	−5.36	2.79	2.83	0.94	1.18	3.35
40 Laplace [76]	2.65	3.14	1.27	−1.00	NA	2.87	2.90	0.97	1.97	3.61
40 Single Gaussian [76]	2.78	3.20	1.14	−1.03	−5.37	2.83	2.93	0.98	2.11	3.61
40 Smooth CP [76]	2.89	3.14	1.20	−1.00	−5.52	2.85	2.88	0.97	1.88	NA
40 Smooth EasyUQ [76]	2.83	3.04	0.79	−1.05	−6.51	2.77	2.48	0.48	1.36	3.24
400 MC Dropout [22]	2.46	3.04	1.99	−0.95	−3.80	2.80	2.89	0.93	1.55	3.59
400 Laplace [76]	2.61	3.07	0.80	−1.11	NA	2.83	2.87	1.04	1.18	3.61
400 Single Gaussian [76]	3.41	3.32	0.85	−1.09	−6.32	2.81	2.87	1.38	2.04	3.61
400 Smooth CP [76]	2.87	3.05	0.83	−1.09	−6.65	2.78	2.84	1.01	1.03	NA
400 Smooth EasyUQ [76]	2.46	2.94	0.55	−1.13	−7.51	2.75	2.41	1.07	0.85	3.24
2L MC Dropout [22]	2.34	2.82	1.48	−1.10	−4.32	2.67	2.70	0.90	1.37	NA
2L Laplace [76]	2.57	2.98	0.56	−1.13	NA	2.76	2.81	1.22	1.24	3.60
2L Single Gaussian [76]	∞	3.78	0.74	−0.96	−7.19	2.76	2.77	10.51	∞	3.61
2L Smooth CP [76]	2.66	2.94	0.63	−1.18	−7.33	2.70	2.67	1.01	0.74	NA
2L Smooth EasyUQ [76]	2.49	2.71	0.36	−1.21	−8.20	2.67	2.30	0.95	0.50	3.23

an attractive alternative, much in line with recent developments in neural network training, where optimization is performed with respect to the CRPS [16, 58]. Its use becomes essential in simplified versions of Algorithm 4.1 that work with basic EasyUQ rather than Smooth EasyUQ.

4.2. Application in Benchmark Settings from Machine Learning. As noted, our intent is to compare Smooth EasyUQ in the integrated version of Algorithm 4.1 to extant, state-of-the-art methods for uncertainty quantification from the statistical and machine learning literatures. The comparison is made on ten datasets for regression tasks using the experimental setup proposed and developed by Hernández-Lobato and Adams [36], Gal and Ghahramani [22], Lakshminarayanan, Pritzel, and Blundell [42], and Duan et al. [18]. Characteristics of the ten datasets are summarized in Table 5, including the size of the datasets, the number of unique outcomes, and the dimension of the input space for the regression problem.

Each dataset is randomly split 20 times into training (72%), validation (18%), and test (10%) sets. However, for the larger datasets, Protein and Year, the train-test split is repeated only five times and a single time, respectively. After finding the optimal set of (hyper)parameters, methods are retrained on the combined training and validation sets (90%) and the resulting predictions are evaluated on the held-out test set (10%). We use the same splits as in the extant literature in the implementation from https://github.com/yaringal/DropoutUncertaintyExps, and the final score is obtained by computing the average score over the splits.

Following the literature, we consider four techniques for the direct generation of conditional predictive distributions that do not use neural networks, namely, a semiparametric variant of the distributional forest technique [18, 63], generalized additive models for location, scale, and shape (GAMLSS) [69], Gaussian process (GP) regression [56], and natural gradient boosting (NGBoost) [18]. We adopt the exact implementation choices of [18] for these techniques, which in some cases involve smoothing. Except for NGBoost, scores for the Year dataset are unavailable (NA), in part, because methods fail to be computationally feasible for a dataset of this size.

The remaining methods considered in Table 5 are based on neural networks, and we adopt the network architectures proposed by Hernández-Lobato and Adams [36] and Gal and Ghahramani [22]. Specifically, we use the ReLU nonlinearity and either a single or two hidden layers, containing 50 hidden units for the smaller datasets, and 100 hidden units for the larger Protein and Year datasets. To tune the network hyperparameters, namely, the regularization parameter λ and the batch size, we use grid search. Thus, the nested hyperparameter selection in the Smooth EasyUQ Algorithm 4.1 finds a best combination of λ, the batch size, ν, and h by optimizing the mean LogS. Our intent is to compare EasyUQ and Smooth EasyUQ to state-of-the-art methods for uncertainty quantification from machine learning, namely, Monte Carlo (MC) Dropout [22] and Deep Ensembles [42], which perform uncertainty quantification directly within the workflow of neural network fitting. Furthermore, these methods are input-based, i.e., they require access to, and operate on, the original covariate or feature vector. As seen in the table, the dimensionality of the input space in the benchmark problems varies between 4 and 90.

In contrast, EasyUQ, CP, and the Single Gaussian technique operate on the basis of the final model output only and so can be applied without the original, potentially high-dimensional covariate or feature vector being available. For CP we adapt our previously described implementation with further refined splits into training (57.6%), calibration (14.4%), validation (18%), and test (10%) sets. Smooth CP uses the respective variant of Algorithm 4.1. An intermediary role between input-based and output-based methods is assumed by the recently developed Laplace approach [37, 59], which leverages scalable Laplace approximations based on weights of the trained network. For our numerical experiments we use the `laplace` software library for PyTorch [15].

A critical implementation decision in the intended comparisons is the number of training epochs in learning the neural network. While the original setup specifies 40 training epochs [36], MC Dropout uses 400 or, in the 2-layer configuration, 4,000 iterations [22]. Therefore, to enable proper comparison, we apply the competing methods in three distinct neural network configurations, namely, a single-layer network with 40 training epochs (prefix 40 in Tables 5 and 6), a single-layer network with 400 training epochs (prefix 400), and a 2-layer architecture with 4,000 training epochs (prefix 2L). In Tables 5 and 6, key comparisons between techniques for uncertainty quantification are then within the respective three groups of methods for which the neural network configurations used are identical.

4.3. Comparison of Predictive Performance.

We assess the predictive performance of EasyUQ, Smooth EasyUQ, and other methods for probabilistic forecasting and uncertainty quantification by comparing the mean LogS in Table 5. We use the LogS from (2.3) in negative orientation, so smaller values correspond to better performance. Evidently, the use of the LogS, which is customary in machine learning,

Table 6 *Predictive performance for competing methods of uncertainty quantification in regression problems in terms of the mean CRPS in a popular benchmark setting from machine learning* [18, 22, 36, 42]*. For each dataset, we show size, number of unique outcomes, and dimension of the input (covariate or feature) space. Italics indicate discrete datasets where the number of unique outcomes is small. For Kin8mn and Naval the mean CRPS has been multiplied by factors of* 10 *and* 1,000*, respectively. For each block of comparable methods, the best (lowest) mean score is set in green. For details, see section* 4.2.

Method / Dataset	Boston	Concrete	Energy	Kin8nm	*Naval*	Power	Protein	*Wine*	Yacht	*Year*
Size	506	1,030	768	8,192	*11,934*	9,568	45,730	*1,599*	308	*515,345*
Unique Outcomes	229	845	586	8,191	*51*	4,836	15,903	*6*	258	*89*
Dimension Input Space	13	8	8	8	*16*	4	9	*11*	6	*90*
40 Deep Ensembles	1.59	3.04	0.78	0.48	0.41	2.23	2.40	0.34	0.45	4.35
40 Laplace	1.71	3.02	0.45	0.49	NA	2.46	2.46	0.35	0.80	4.72
40 Single Gaussian	1.72	3.03	0.41	0.48	0.66	2.24	2.48	0.35	0.83	4.72
40 CP	1.73	3.04	0.45	0.49	0.58	2.24	2.47	0.36	0.90	NA
40 Smooth CP	1.74	3.05	0.45	0.49	0.58	2.24	2.47	0.36	0.91	NA
40 EasyUQ	1.69	2.94	0.34	0.48	0.54	2.21	2.22	0.31	0.66	4.35
40 Smooth EasyUQ	1.64	2.89	0.33	0.48	0.55	2.20	2.20	0.32	0.64	4.34
400 MC Dropout	1.56	2.79	0.37	0.48	1.22	2.21	2.40	0.35	0.57	4.73
400 Laplace	1.66	2.67	0.29	0.44	NA	2.17	2.36	0.37	0.41	4.73
400 Single Gaussian	1.61	2.72	0.29	0.44	0.27	2.17	2.36	0.38	0.41	4.73
400 CP	1.70	2.77	0.30	0.45	0.20	2.16	2.38	0.37	0.42	NA
400 Smooth CP	1.71	2.77	0.30	0.45	0.20	2.16	2.38	0.37	0.43	NA
400 EasyUQ	1.75	2.72	0.26	0.44	0.12	2.16	2.10	0.35	0.39	4.33
400 Smooth EasyUQ	1.60	2.61	0.25	0.44	0.13	2.15	2.09	0.37	0.35	4.33
2L MC Dropout	1.45	2.19	0.33	0.41	1.07	1.92	1.95	0.33	0.47	4.63
2L Laplace	1.64	2.29	0.22	0.44	NA	2.01	2.15	0.42	0.41	4.65
2L Single Gaussian	1.89	2.27	0.25	0.41	0.11	2.03	2.04	0.45	0.25	4.69
2L CP	1.70	2.47	0.24	0.42	0.11	2.02	2.02	0.38	0.36	NA
2L Smooth CP	1.71	2.48	0.24	0.42	0.11	2.03	2.02	0.38	0.36	NA
2L EasyUQ	2.07	2.40	0.24	0.42	0.03	1.98	1.83	0.42	0.30	4.30
2L Smooth EasyUQ	1.66	2.14	0.21	0.40	0.04	1.97	1.82	0.40	0.27	4.31

prevents comparisons to the basic versions of EasyUQ and CP, to which we turn in Table 6.

A first insight from Table 5 is that, in general, the methods in the second, third, and fourth blocks, which are based on neural networks, perform better relative to the direct methods not based on neural networks in the first block (from top to bottom). Thus, we focus attention on the comparison of distinct methods for uncertainty quantification in neural networks, namely, Deep Ensembles [42] or MC dropout [22], the Laplace approach [59], the Single Gaussian technique, Smooth CP, and Smooth EasyUQ. The 2-layer architecture generally improves results compared to using a single layer for the neural network. Smooth EasyUQ dominates the Single Gaussian and Smooth CP techniques and generally yields lower mean LogS than Deep Ensembles, MC Dropout, or the Laplace approach. In 24 of the $3 \times 10 = 30$ fivefold comparisons across the bottom three blocks, Smooth EasyUQ achieves or shares the top score. For eight of the ten datasets considered, the best performance across all 19 methods considered, including both neural network–based approaches and techniques not based on neural networks, is achieved or shared by Smooth EasyUQ under the 2-layer network architecture. While this is not an exhaustive evaluation and no single method dominates universally, we note that Smooth EasyUQ is highly competitive with state-of-the-art techniques for uncertainty quantification from machine learning.

To allow comparison with the basic form of EasyUQ, which generates discrete predictive distributions, we use Table 6 and the mean CRPS from (2.2) to assess predictive performance. Each of the three blocks in the table allows for a seven-way comparison among either Deep Ensembles or MC Dropout, the Laplace approach, the Single Gaussian technique, conformal prediction in its basic (CP) and smoothed (Smooth CP) forms, the basic version of EasyUQ, and Smooth EasyUQ. As noted, the Naval, Wine, and Year datasets are distinctly discrete, with 51, 6, and 89 unique outcomes, respectively. For data of this type, predictive distributions ought to be discrete. Accordingly, there are no benefits of using Smooth EasyUQ for these datasets compared to using basic EasyUQ, which adapts readily to discrete outcomes. In six of the $3 \times 3 = 9$ sevenfold comparisons for the discrete datasets, the basic version of EasyUQ achieves the lowest mean score. Across the remaining seven datasets and for all three network configurations, smoothing is beneficial and Smooth EasyUQ outperforms the basic version of EasyUQ. In 15 of the $3 \times 7 = 21$ sevenfold comparisons on these datasets, Smooth EasyUQ achieves or shares the top score. All but one of the binary comparisons between Smooth CP and Smooth EasyUQ, all but two of the comparisons between the Single Gaussian technique and Smooth EasyUQ, all but one of the comparisons between the Laplace method and Smooth EasyUQ, and all but eight of the comparisons between Deep Ensembles or MC Dropout and Smooth EasyUQ are in favor of the latter.

5. Discussion. In this paper we have proposed EasyUQ and Smooth EasyUQ as general methods for the conversion of single-valued computational model output into calibrated predictive distributions, based on a training set of model output–outcome pairs and a natural assumption of isotonicity. Contrary to recent comments in review articles that lament an “absence of theory” [1, p. 244] for data-driven approaches to uncertainty quantification, the basic version of EasyUQ enjoys strong theoretical support, in sharing the optimality and consistency properties of the general isotonic distributional regression (IDR) [35] method. The basic EasyUQ approach is fully automated, does not require any implementation choices, and the generated predictive distributions are discrete. The more elaborate Smooth EasyUQ approach developed in this paper generates predictive distributions with Lebesgue densities based on a kernel smoothing of the original IDR distributions, while preserving the key properties of the basic approach. Code for the implementation of IDR in Python [53] and replication material for this article are openly available [76].

The method is general, handling both discrete outcomes, with the basic technique being tailored to this setting, and continuous outcomes, for which Smooth EasyUQ is the method of choice. It applies whenever single-valued model output is to be converted into a predictive distribution, covering both the case of point forecasts, as in the WeatherBench example, and computational model output in all facets, such as in the machine learning example, where EasyUQ and Smooth EasyUQ convert single-valued neural network output into predictive distributions. Percentiles extracted from the predictive distributions can be used to generate prediction intervals.

The proposed term EasyUQ stems from various desirable properties. First, the basic version of EasyUQ does not involve tuning parameters or require user intervention. Second, EasyUQ operates on the natural, easily interpretable and communicable assumption that larger values of the computational model output yield predictive distributions that are stochastically larger. Third, EasyUQ is an output-based technique, i.e., it merely requires training data in the form of model output–outcome pairs (x_i, y_i) as in (2.1), without any need to access the potentially high-dimensional covariate or feature vector z_i, which serves as input to the computational model that generates x_i. This property is shared with the widely used single Gaussian technique and related methods such as the early geostatistical output perturbation (GOP) [23] approach and the quantile regression averaging (QRA) [51] method for the generation of prediction intervals.

The term conformal prediction [46, 75, 72] refers to a family of output-based methods that yield predictive distributions and prediction intervals that enjoy attractive out-of-sample coverage guarantees, but often mean that the shape and scale of the predictive distributions do not vary with the model output. In simple problems, where predictive distributions that are essentially translates of each other are appropriate, both CP and EasyUQ perform well and typically yield very similar predictive performance, as illustrated by the temperature example in section 2.3. The flexibility of EasyUQ, which allows for predictive distributions that vary in shape and/or scale, subject to the isotonicity condition, materializes in more challenging problems, where predictive distributions that are translates of each other fail. While EasyUQ adapts to such settings without any need for user intervention, CP might suffer considerable loss in predictive performance, even if adapted manually, as exemplified in the precipitation example in section 2.4.

While adaptive variants of CP are available, their predictive performance in both simulated and real-data settings has been mixed compared to standard variants [75]. Recently, Boström, Johansson, and Löfström [7] investigated Mondrian (i.e., covariate-conditional) CP as a flexible alternative, in which conformal predictive distributions are built on separate categories formed by binning covariates (in our case, the model output). This requires additional implementation decisions, namely, on the choice of the bins. Boström, Johansson, and Löfström [7] take five bins with equal numbers of training instances, which improves predictive performance in their experiments. From a methodological point of view, in situations where the isotonicity assumption of IDR is met, the binning approach of Mondrian CP can be understood as an approximation to EasyUQ. EasyUQ finds optimal binnings without manual intervention [35, Thm. 2], and training borrows strength from the entirety of the training data, whereas Mondrian CP diminishes the training sample by splitting it, which introduces a trade-off between training data size and adaptivity. A limitation of EasyUQ is that estimates under isotonicity constraints tend to be inconsistent at the boundary of the covariate domain [31], which raises the danger of disproportionately decreased spread of EasyUQ distributions at extreme values of the model output. In settings where this is of concern, a potential remedy is to resort to Mondrian CP at extreme values while reaping the benefits of EasyUQ at moderate values of the model output. We leave further methodological development in these directions to future work.

In contrast to CP and EasyUQ, input-based methods such as MC Dropout [22], Deep Ensembles [42], the techniques proposed by Camporeale and Carè [11] and Chung et al. [14], and the reference methods considered by Duan et al. [18] require access to the covariate or feature vector z_i. Input-based methods are much more flexible than output-based methods and thus have higher potential in principle, as evidenced by the success of ensemble methods in numerical weather prediction [3, 27]. However, they tend to be more computationally intense than output-based methods, and as the machine learning example in our paper shows, they may not outperform the latter. Generally, sophisticated input-based methods for uncertainty quantification might realize their potential when applied to substantively informed, highly complex computational models, as in the case of numerical weather prediction, where predictive uncertainty varies. Output-based approaches to uncertainty quantification typically are less complex and thus easier to implement and might nonetheless yield competitive predictive performance when applied to output from data-driven models, such as the neural network models in the benchmark setting from machine learning.

We end the paper with speculations about the usage of EasyUQ and Smooth EasyUQ in weather prediction. The current approach to forecasts at lead times of hours to weeks rests on ensembles of physics-based numerical models [3, 27], but it is being challenged by the advent of purely data-driven models based on ever more sophisticated neural networks [19, 64]. Published only recently, the WeatherBench comparison [57] showed a huge performance gap between forecasts from physics-based numerical models and neural network–based, purely data-driven forecasts, with the latter being clearly inferior, as exemplified in our Tables 1 and 3. Fast breaking developments suggest that the situation may have reversed since then, with purely data-driven approaches now outperforming physics-based forecasts of univariate weather quantities [4, 6, 13, 43]. There is a caveat, though, as under the new, data-driven paradigm, spatiotemporal and intervariable dependence structures might be misrepresented due to the lack of physical constraints in the model and a need for hierarchical temporal aggregation in the generation of weather scenarios [6, 19]. However, the resulting neural network based forecasts can be subjected to EasyUQ and Smooth EasyUQ, and samples from the resulting predictive distributions can be merged by empirical copula techniques such as ensemble copula coupling (ECC) [61] to adopt and transfer spatiotemporal and intervariable dependence structures in physics-based ensemble forecasts. Hybrid approaches of this type might combine and extract the best from both traditional physics-based and emerging data-driven approaches to weather prediction and may turn out to be superior to both.

Appendix A. Consistency of Smoothed Conditional CDF Estimates. In this appendix, we prove the uniform asymptotic consistency of smoothed estimators of conditional CDFs under mild conditions. We operate in a very general setting in which the smoothed estimate

$$\check{F}_{x;n}(y) = \int_{-\infty}^{\infty} \hat{F}_{x;n}(t)\, K_{h_n}(y-t)\, \mathrm{d}t \tag{A.1}$$

arises from a basic estimate $\hat{F}_{x;n}$ that uses a sample of the form (2.1) of size n. We do not make further assumptions on the form or origin of the basic estimate, though in (3.3) we specialize to EasyUQ. As in the main text, we let $K_h(u) = (1/h)\,\kappa(u/h)$ for a smooth probability density κ, such as a Gaussian or a Student-t density, but we now allow for the possibility that the bandwidth $h_n > 0$ varies with the sample size.

In formulating the subsequent consistency result, we only require that the basic estimates $\hat{F}_{x;n}$ be asymptotically consistent and that the true conditional CDFs $F_x(y)$ be smooth in y, and we put mild assumptions on κ. IDR and its special case EasyUQ indeed are asymptotically consistent under reasonable assumptions. Specifically, let $(X_{ni}, Y_{ni}) \in \mathcal{X} \times \mathbb{R}$ for $i = 1, \ldots, n$ be a triangular array of covariates and real-valued observations, which are independent across i for any fixed $n = 1, 2, \ldots$ and have the same distribution as a pair (X, Y) with conditional CDFs $F_x(y) = \mathbb{P}(Y \le y \mid X = x)$. Let $\hat{F}_{x;n}$ be the IDR CDF computed from this sample with an arbitrary admissible interpolation method for $x \notin \{X_{n1}, \ldots, X_{nn}\}$. Here, $\mathcal{X}$ is some subset of $\mathbb{R}^d$ that is equipped with a partial order $\preceq$. The key assumption is that the conditional CDFs F_x are nondecreasing in stochastic order, i.e., $x \preceq x'$ implies that $F_x(y) \ge F_{x'}(y)$ for all $y \in \mathbb{R}$. For continuous covariates, one furthermore needs to assume that the covariate values become sufficiently dense in $\mathcal{X}$ as n increases, and that a uniform continuity assumption on the conditional CDFs holds; cf. the references discussed below. Then the following assumption on uniform consistency is satisfied.

Assumption A.1. *There exists a sequence* $(\varepsilon_n)_{n=1,2,\ldots}$ *such that, for all* x *in some set* $\mathcal{X}_n \subseteq \mathcal{X}$, *we have*

$$\lim_{n\to\infty} \mathbb{P}\left(\sup_{y\in\mathbb{R},\, x\in\mathcal{X}_n} |\hat{F}_{x;n}(y) - F_x(y)| \geq \varepsilon_n \right) = 0. \tag{A.2}$$

The sequence of sets $(\mathcal{X}_n)_{n=1,2,\ldots}$ in the above assumption usually consists of all points in $\mathcal{X}$ whose distance from the boundary of $\mathcal{X}$ is not less than $\delta_n > 0$, where δ_n is a sequence that converges to zero. Multivariate covariates are treated in Henzi, Ziegel, and Gneiting [35], who demonstrate Assumption A.1 with $\varepsilon_n = \varepsilon > 0$ for any constant $\varepsilon > 0$ and $\delta_n = \delta$ for any constant $\delta > 0$. The case $\mathcal{X} = (a, b) \subset \mathbb{R}$ with the usual total order corresponds to the typical setting for EasyUQ and is treated by Mösching and Dümbgen [50], who show that one can choose ε_n of order $(\log(n)/n)^{\alpha/(2\alpha+1)}$ if the conditional CDFs are Hölder continuous in x with index $\alpha \in (0, 1]$. In this case, the sets $\mathcal{X}_n$ are of the form $(a + \delta_n, b - \delta_n)$ with δ_n converging to zero at rate $(\log(n)/n)^{1/(2\alpha+1)}$. The case of ordinal covariates in a finite set was investigated by El Barmi and Mukerjee [20], and it can be shown that one can choose $\varepsilon_n = (\log(n)/n)^{1/2}$ and consistency holds for all values of x. While the authors do not explicitly state this convergence rate, it follows from the last displayed equation prior to their Theorem 1, according to which the maximal error (in sup-norm) of the IDR CDFs is less than or equal to the error of the empirical CDFs stratified by the covariate. The sup-norm error of the empirical CDFs can be bounded by $(\log(n)/n)^{1/2}$ (by the Dvoretzky–Kiefer–Wolfowitz inequality, $\log(n)$ could be replaced by any other sequence diverging to ∞), so the rate stated above applies.

A situation of particular applied relevance arises for distributional single index models (DIMs) [33], which can be interpreted as a special case of EasyUQ. Specifically, let $(Z_{ni}, Y_{ni}) \in \mathcal{Z} \times \mathbb{R}$ for $i = 1, \ldots, n$, where n is a positive integer, be a triangular array of covariates and observations, which are independent across i for any fixed n and have the same distribution as some pair (Z, Y). Suppose that there is a function $\theta\colon \mathcal{Z} \to \mathbb{R}$, called the index function, such that the conditional CDFs $F_x(y) = \mathbb{P}(Y \leq y \mid \theta(Z) = x)$ are stochastically ordered in x. For each sample size n, the index function is estimated by $\hat{\theta}_n$. Denote by $\hat{F}_{x;n}$ the IDR CDF computed from the pseudo-observations $(\hat{\theta}_n(Z_{n1}), Y_{n1}), \ldots, (\hat{\theta}_n(Z_{nn}), Y_{nn})$ with an arbitrary admissible interpolation method for $x \notin \{\hat{\theta}_n(Z_{n1}), \ldots, \hat{\theta}_n(Z_{nn})\}$. If the index function is estimated consistently at a sufficiently fast rate and the pseudocovariates $\theta(Z_i)$ become sufficiently dense in some interval $\mathcal{X} \subset \mathbb{R}$, then Assumption A.1 is satisfied with a rate ε_n of $(\log(n)/n)^{1/6}$ and $\mathcal{X}$ of the form $(a + \delta_n, b - \delta_n)$ for $\delta_n = (\log(n)/n)^{1/6}$.

The second assumption is a natural condition on the true conditional CDFs F_x, without which one would not want to smooth in the first place.

Assumption A.2. *There exist constants* $L > 0$ *and* $\alpha \in (0, 1]$ *such that for all* $x \in \mathcal{X}$ *and* $u, v \in \mathbb{R}$,

$$|F_x(u) - F_x(v)| \leq L\, |u - v|^{\alpha}.$$

In particular, Assumption A.2 is satisfied with $\alpha = 1$ if the conditional distributions admit Lebesgue densities that are uniformly bounded.

THEOREM A.3. *Suppose that Assumptions* A.1 *and* A.2 *hold, and assume that* $c_{\kappa,\alpha} = \int_{-\infty}^{\infty} |s|^{\alpha}\, \kappa(s)\, \mathrm{d}s$ *is finite. Then*

$$\lim_{n\to\infty} \mathbb{P}\left(\sup_{y\in\mathbb{R},\, x\in\mathcal{X}_n} |\check{F}_{x;n}(y) - F_x(y)| \geq \varepsilon_n + L c_{\kappa,\alpha} h_n^{\alpha} \right) = 0. \tag{A.3}$$

An immediate consequence is that if $h_n^{\alpha} = \mathcal{O}(\varepsilon_n)$, then the smoothed estimate $\check{F}_{x;n}$ admits the same convergence rate as the basic estimate $\hat{F}_{x;n}$.

Proof of Theorem A.3. The error of $\check{F}_{n;x}$ is upper bounded as

$$\begin{aligned} |\check{F}_{x;n}(y) - F_x(y)| &= \left| \int_{-\infty}^{\infty} [\hat{F}_{x;n}(t) - F_x(t) + F_x(t) - F_x(y)]\, K_{h_n}(y-t)\, \mathrm{d}t \right| \\ &\leq \tilde{\varepsilon}_n(x) + \int_{-\infty}^{\infty} |F_x(y - h_n s) - F_x(y)|\, \kappa(s)\, \mathrm{d}s, \end{aligned}$$

where $\tilde{\varepsilon}_n(x) = \sup_{z\in\mathbb{R}} |\hat{F}_{n;x}(z) - F_x(z)|$, as we see by making the change of variable $s = (y-t)/h_n$. By Assumption A.2 we obtain that

$$\int_{-\infty}^{\infty} |F_x(y - h_n s) - F_x(y)|\, \kappa(s)\, \mathrm{d}s \leq L c_{\kappa,\alpha} h_n^{\alpha}.$$

Hence

$$|\check{F}_{n;x}(y) - F_x(y)| \leq \tilde{\varepsilon}_n(x) + L c_{\kappa,\alpha} h_n^{\alpha},$$

and the claim now follows from Assumption A.1. □

We emphasize that Assumption A.1 is a high-level condition that is not specific to IDR, DIMs, or EasyUQ. Theorem A.3 implies that any sequence of conditional CDF estimates $(\hat{F}_{x;n})$ that is consistent in the sense of Assumption A.1 can be smoothed consistently via (A.1), either with a fixed kernel κ or with a kernel κ selected from a suitably limited class of candidate functions. This is a result of independent interest that goes well beyond the classical setting of smoothing an empirical distribution. To give an example, the approach can be applied to smoothing conditional CDF estimators under weaker stochastic dominance constraints [32], where Theorem A.3 directly yields consistency of the smoothed estimator. Another method where smoothing might be beneficial is the distributional random forest technique [12, 63], which, like IDR, generates discrete estimators of conditional CDFs. However, the conclusions from Theorem A.3 do not apply directly in this case, since only pointwise but not uniform consistency has been proven for these estimators [12, Cor. 5].

Acknowledgments. We thank Sam Allen, Steffen Betsch, Timo Dimitriadis, Alexander Jordan, Gregor Koehler, and Sebastian Lerch for insightful comments, and we are grateful to Peter Knippertz for many discussions, meteorological advice and insights, and encouragement. Special thanks to Gregor Koehler for suggesting and implementing code for GPU computation. Two anonymous referees provided a wealth of constructive feedback, for which we are most grateful.

REFERENCES

[1] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, *A review of uncertainty quantification in deep learning: Techniques, applications and challenges*, Inform. Fusion, 76 (2021), pp. 243–297. (Cited on pp. 92, 109, 114)

[2] E. Baker, P. Barbillon, A. Fadikar, R. B. Gramacy, R. Herbei, D. Higdon, J. Huang, L. R. Johnson, P. Ma, A. Mondal, B. Pires, J. Sacks, and V. Sokolov, *Analyzing stochastic computer models: A review with opportunities*, Statist. Sci., 37 (2022), pp. 64–89. (Cited on p. 109)

[3] P. Bauer, A. Thorpe, and G. Brunet, *The quiet revolution of numerical weather prediction*, Nature, 525 (2015), pp. 47–55. (Cited on pp. 101, 115, 116)

[4] Z. Ben Bouallègue, M. C. A. Clare, L. Magnusson, E. Gascón, M. Maier-Gerber, M. Janoušek, M. Rodwell, F. Pinault, J. S. Dramsch, S. T. K. Lang, B. Raoult, F. Rabier, M. Chevallier, I. Sandu, P. Dueben, M. Chantry, and F. Pappenberger, *The Rise of Data-Driven Weather Forecasting: A First Statistical Assessment of Machine Learning-Based Weather Forecasts in an Operational-Like Context*, preprint, https://arxiv.org/abs/2307.10128, 2023. (Cited on p. 116)

[5] J. O. Berger and L. A. Smith, *On the statistical formalism of uncertainty quantification*, Annu. Rev. Stat. Appl., 6 (2019), pp. 433–460. (Cited on pp. 92, 93)

[6] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, *Accurate medium-range global weather forecasting with 3D neural networks*, Nature, 619 (2023), pp. 533–538. (Cited on p. 116)

[7] H. Boström, U. Johansson, and T. Löfström, *Mondrian conformal predictive distributions*, in Tenth Symposium on Conformal and Probabilistic Prediction and Applications, Proc. Mach. Learn. Res. 102, ML Research Press, 2021, pp. 3–15. (Cited on pp. 100, 115)

[8] A. Bowman, P. Hall, and T. Prvan, *Bandwidth selection for the smoothing of distribution functions*, Biometrika, 85 (1998), pp. 799–808. (Cited on p. 106)

[9] R. P. Brent, *Algorithms for Minimization without Derivatives*, Prentice-Hall, 1973. (Cited on pp. 106, 110)

[10] J. Bröcker and L. A. Smith, *From ensemble forecasts to predictive distribution functions*, Tellus A, 60 (2008), pp. 663–678. (Cited on pp. 106, 108)

[11] E. Camporeale and A. Carè, *ACCRUE: Accurate and reliable uncertainty estimate in deterministic models*, Int. J. Uncertain. Quantif., 11 (2021), pp. 81–94. (Cited on p. 115)

[12] D. Ćevid, L. Michel, J. Näf, P. Bühlmann, and N. Meinshausen, *Distributional random forests: Heterogeneity adjustment and multivariate distributional regression*, J. Mach. Learn. Res., 23 (2022), pp. 14987–15065. (Cited on p. 118)

[13] K. Chen, T. Han, J. Gong, L. Bai, F. Ling, J.-J. Luo, X. Chen, L. Ma, T. Zhang, R. Su, Y. Ci, B. Li, X. Yang, and W. Ouyang, *FengWu: Pushing the Skillful Global Medium-Range Weather Forecast beyond* 10 *Days Lead*, preprint, https://arxiv.org/abs/2304.02948, 2023. (Cited on p. 116)

[14] Y. Chung, W. Neiswanger, I. Char, and J. Schneider, *Beyond pinball loss: Quantile methods for calibrated uncertainty quantification*, in 35th Conference on Neural Information Processing Systems (NeurIPS), Neural Information Processing Systems Foundation, 2021, pp. 1–14. (Cited on pp. 109, 115)

[15] E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig, *Laplace redux—effortless Bayesian deep learning*, in 35th Conference on Neural Information Processing Systems, Neural Information Processing Systems Foundation, 2021, pp. 1–15. (Cited on p. 112)

[16] A. D'Isanto and K. L. Polsterer, *Photometric redshift estimation via deep learning: Generalized and pre-classification-less, image based, fully probabilistic redshifts*, Astronomy & Astrophys., 609 (2018), art. A111. (Cited on p. 111)

[17] K. Doubleday, V. V. S. Hernandez, and B.-M. Hodge, *Benchmark probabilistic solar forecasts: Characteristics and recommendations*, Solar Energy, 206 (2020), pp. 52–67. (Cited on p. 107)

[18] T. Duan, A. Anand, D. Y. Ding, K. K. Thai, S. Basu, A. Ng, and A. Schuler, *NGBoost: Natural gradient boosting for probabilistic prediction*, in 37th International Conference on Machine Learning, Proc. Mach. Learn. Res. 119, ML Research Press, 2020, pp. 2690–2700. (Cited on pp. 109, 111, 112, 113, 115)

[19] I. Ebert-Uphoff and K. Hilburn, *The outlook for AI weather prediction*, Nature, 619 (2023), pp. 473–474. (Cited on pp. 101, 116)

[20] H. El Barmi and H. Mukerjee, *Inferences under a stochastic constraint: The k-sample case*,

J. Amer. Statist. Assoc., 100 (2005), pp. 252–261. (Cited on pp. 97, 98, 117)

[21] C. Fernandez and M. F. J. Steel, *Multivariate Student-t regression models: Pitfalls and inference*, Biometrika, 86 (2009), pp. 153–167. (Cited on p. 106)

[22] Y. Gal and Z. Ghahramani, *Dropout as a Bayesian approximation: Representing model uncertainty in deep learning*, in 33rd International Conference on Machine Learning, Proc. Mach. Learn. Res. 48, ML Research Press, 2016, pp. 1050–1059. (Cited on pp. 109, 111, 112, 113, 115)

[23] Y. Gel, A. E. Raftery, and T. Gneiting, *Calibrated probabilistic mesoscale weather field forecasting: The geostatistical output perturbation method*, J. Amer. Statist. Assoc., 99 (2004), pp. 575–583. (Cited on p. 115)

[24] R. Ghanem, D. Higdon, and H. Owhadi, eds., *Handbook of Uncertainty Quantification*, Springer, 2017. (Cited on p. 92)

[25] T. Gneiting, F. Balabdaoui, and A. E. Raftery, *Probabilistic forecasts, calibration and sharpness*, J. R. Stat. Soc. Ser. B Stat. Methodol., 69 (2007), pp. 243–268. (Cited on p. 95)

[26] T. Gneiting and M. Katzfuss, *Probabilistic forecasting*, Annu. Rev. Stat. Appl., 1 (2014), pp. 125–151. (Cited on p. 92)

[27] T. Gneiting and A. E. Raftery, *Weather forecasting with ensemble methods*, Science, 310 (2005), pp. 248–249. (Cited on pp. 98, 100, 103, 115, 116)

[28] T. Gneiting and A. E. Raftery, *Strictly proper scoring rules, prediction, and estimation*, J. Amer. Stat. Assoc., 102 (2007), pp. 359–378. (Cited on pp. 93, 96)

[29] T. Gneiting, A. E. Raftery, A. H. Westveld, and T. Goldman, *Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation*, Monthly Weather Rev., 133 (2005), pp. 1098–1118. (Cited on pp. 97, 101, 102)

[30] T. Gneiting and P. Vogel, *Receiver operating characteristic (ROC) curves: Equivalences, beta model, and minimum distance estimation*, Mach. Learn., 111 (2022), pp. 2147–2159. (Cited on p. 97)

[31] A. Guntuboyina and B. Sen, *Nonparametric shape-restricted regression*, Statist. Sci., 33 (2018), pp. 563–594. (Cited on p. 115)

[32] A. Henzi, *Consistent estimation of distribution functions under increasing concave and convex stochastic ordering*, J. Business Econom. Statist., 41 (2023), pp. 1203–1214. (Cited on pp. 97, 118)

[33] A. Henzi, G.-R. Kleger, and J. F. Ziegel, *Distributional (single) index models*, J. Amer. Statist. Assoc., 118 (2023), pp. 489–503. (Cited on pp. 98, 99, 117)

[34] A. Henzi, A. Mösching, and L. Dümbgen, *Accelerating the pool-adjacent-violators algorithm for isotonic distributional regression*, Methodol. Comput. Appl. Probab., 24 (2022), pp. 2633–2645. (Cited on pp. 97, 108)

[35] A. Henzi, J. F. Ziegel, and T. Gneiting, *Isotonic distributional regression*, J. R. Stat. Soc. Ser. B Stat. Methodol., 83 (2021), pp. 963–993. (Cited on pp. 93, 95, 96, 97, 98, 99, 102, 106, 108, 114, 115, 117)

[36] J. M. Hernández-Lobato and R. P. Adams, *Probabilistic backpropagation for scalable learning of Bayesian neural networks*, in 32nd International Conference on Machine Learning, Proc. Mach. Learn. Res. 37, ML Research Press, 2015, pp. 1861–1869. (Cited on pp. 111, 112, 113)

[37] A. Immer, M. Bauer, V. Fortuin, G. Rätsch, and M. E. Khan, *Scalable marginal likelihood estimation for model selection in deep learning*, in 38th International Conference on Machine Learning, 2021. Proc. Mach. Learn. Res. 139, ML Research Press, 2021, pp. 4563–4573. (Cited on pp. 109, 112)

[38] A. Jordan, F. Krüger, and S. Lerch, *Evaluating probabilistic forecasts with scoringRules*, J. Statist. Software, 90 (2019), pp. 1–37. (Cited on p. 96)

[39] M. Köhler, A. Schindler, and S. Sperlich, *A review and comparison of bandwidth selection methods for kernel regression*, Internat. Statist. Rev., 82 (2014), pp. 243–274. (Cited on p. 105)

[40] J. Kohonen and J. Suomela, *Lessons learned in the challenge: Making predictions and scoring them*, in Machine Learning Challenges: Evaluating Predictive Uncertainty, Visual Object Classification, and Recognizing Textual Entailment, J. Quiñonero-Candela, I. Dagan, B. Magnini, and F. d'Alché-Buc, eds., Springer, 2006, pp. 95–116. (Cited on p. 106)

[41] V. Kuleshov, N. Fenner, and S. Ermon, *Accurate uncertainties for deep learning using calibrated regression*, in 35th International Conference on Machine Learning, Proc. Mach. Learn. Res. 80, ML Research Press, 2018, pp. 2796–2804. (Cited on p. 109)

[42] B. Lakshminarayanan, A. Pritzel, and C. Blundell, *Simple and scalable predictive uncertainty estimation using deep ensembles*, in 31st Conference on Neural Information Process-

ing Systems (NIPS), Neural Information Processing Systems Foundation, 2017. (Cited on pp. 109, 111, 112, 113, 115)

[43] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, A. Pritzel, S. Ravuri, T. Ewalds, F. Alet, Z. Eaton-Rosen, W. Hu, A. Merose, S. Hoyer, G. Holland, J. Stott, O. Vinyals, S. Mohamed, and P. Battaglia, *GraphCast: Learning Skillful Medium-Range Global Weather Forecasting*, preprint, https://arxiv.org/abs/2212.12794, 2022. (Cited on p. 116)

[44] M. Leutbecher and T. N. Palmer, *Ensemble forecasting*, J. Comput. Phys., 227 (2008), pp. 3515–3539. (Cited on pp. 98, 100)

[45] Q. Li, J. Lin, and J. S. Racine, *Optimal bandwidth selection for nonparametric conditional distribution and quantile functions*, J. Business Econom. Statist., 31 (2013), pp. 57–65. (Cited on p. 106)

[46] C. Marx, S. Zhou, W. Neiswanger, and S. Ermon, *Modular conformal calibration*, in 39th International Conference on Machine Learning, 2022. Proc. Mach. Learn. Res. 62, ML Research Press, 2022, pp. 15180–15195. (Cited on pp. 109, 115)

[47] J. E. Matheson and R. L. Winkler, *Scoring rules for continuous probability distributions*, Management Sci., 22 (1976), pp. 1087–1096. (Cited on pp. 93, 96)

[48] J. W. Messner, G. J. Mayr, D. S. Wilks, and A. Zeileis, *Extending extended logistic regression: Extended versus separate versus ordered versus censored*, Monthly Weather Rev., 142 (2014), pp. 3003–3014. (Cited on p. 102)

[49] F. Molteni, R. Buizza, T. N. Palmer, and T. Petroliagis, *The ECMWF ensemble prediction system: Methodology and validation*, Quart. J. Roy. Meteorol. Soc., 122 (1996), pp. 73–119. (Cited on pp. 93, 100)

[50] A. Mösching and L. Dümbgen, *Monotone least squares and isotonic quantiles*, Electron. J. Statist., 14 (2020), pp. 24–49. (Cited on pp. 98, 117)

[51] J. Nowotarski and R. Weron, *Computing electricity spot price prediction intervals using quantile regression and forecast averaging*, Comput. Statist., 30 (2015), pp. 791–803. (Cited on p. 115)

[52] T. N. Palmer, *Predicting uncertainty in forecasts of weather and climate*, Rep. Progr. Phys., 63 (2000), pp. 71–116. (Cited on pp. 98, 100)

[53] *Python Language Reference*, Python Software Foundation, 2023; available at https://python.org/. (Cited on p. 114)

[54] J. Quiñonero-Candela, C. E. Rasmussen, F. Sinz, O. Bousquet, and B. Schölkopf, *Evaluating predictive uncertainty challenge*, in Machine Learning Challenges: Evaluating Predictive Uncertainty, Visual Object Classification, and Recognizing Textual Entailment, J. Quiñonero-Candela, I. Dagan, B. Magnini, and F. d'Alché-Buc, eds., Springer, 2006, pp. 1–27. (Cited on p. 106)

[55] A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski, *Using Bayesian model averaging to calibrate forecast ensembles*, Monthly Weather Rev., 133 (2005), pp. 1155–1174. (Cited on p. 102)

[56] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2005. (Cited on p. 112)

[57] S. Rasp, P. D. Dueben, S. Scher, J. A. Weyn, S. Mouatadid, and N. Thuerey, *WeatherBench: A benchmark dataset for data-driven weather forecasting*, J. Adv. Model. Earth Syst., 12 (2020), art. e2020MS002203. (Cited on pp. 93, 95, 100, 116)

[58] S. Rasp and S. Lerch, *Neural networks for postprocessing ensemble weather forecasts*, Monthly Weather Rev., 146 (2017), pp. 3885–3900. (Cited on p. 111)

[59] H. Ritter, A. Botev, and D. Barber, *A scalable Laplace approximation for neural networks*, in International Conference on Learning Representations, ICLR, 2018, pp. 1–15. (Cited on pp. 112, 113)

[60] C. J. Roy and W. L. Oberkampf, *A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing*, Comput. Methods Appl. Mech. Engrg., 200 (2011), pp. 2131–2144. (Cited on p. 92)

[61] R. Schefzik, T. L. Thorarinsdottir, and T. Gneiting, *Uncertainty quantification in complex simulation models using ensemble copula coupling*, Statist. Sci., 28 (2013), pp. 616–640. (Cited on pp. 95, 116)

[62] M. Scheuerer, *Probabilistic quantitative precipitation forecasting using ensemble model output statistics*, Quart. J. Roy. Meteorol. Soc., 140 (2014), pp. 1086–1096. (Cited on p. 102)

[63] L. Schlosser, T. Hothorn, R. Stauffer, and A. Zeileis, *Distributional regression forests for probabilistic precipitation forecasting in complex terrain*, Ann. Appl. Statist., 13 (2019), pp. 1564–1589. (Cited on pp. 112, 118)

[64] M. G. Schultz, C. Betancourt, B. Gong, F. Kleinert, M. Langguth, L. H. Leufen,

A. Mozaffari, and S. Stadtler, *Can deep learning beat numerical weather prediction?*, Philos. Trans. Roy. Soc. A, 379 (2021), art. 20200097. (Cited on p. 116)

[65] M. Shaked and J. G. Shanthikumar, *Stochastic Orders*, Springer, 2007. (Cited on pp. 93, 97)

[66] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall, 1986. (Cited on pp. 105, 106, 109, 110)

[67] J. M. Sloughter, A. E. Raftery, T. Gneiting, and C. Fraley, *Probabilistic quantitative precipitation forecasting using Bayesian model averaging*, Monthly Weather Rev., 135 (2007), pp. 3209–3220. (Cited on p. 102)

[68] R. C. Smith, *Uncertainty Quantification: Theory, Implementation, and Applications*, SIAM, 2014, https://doi.org/10.1137/1.9781611973228. (Cited on p. 92)

[69] D. M. Stasinopoulos and R. A. Rigby, *Generalized additive models for location, scale and shape (GAMLSS) in* R, J. Statist. Software, 23 (2007), pp. 1–46. (Cited on p. 112)

[70] T. J. Sullivan, *Introduction to Uncertainty Quantification*, Springer, 2015. (Cited on p. 92)

[71] N. Trefethen, *Discrete or continuous?*, SIAM News, 45 (2012), p. 1, https://people.maths.ox.ac.uk/trefethen/may12.pdf. (Cited on p. 92)

[72] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*, 2nd ed., Springer, 2022. (Cited on pp. 94, 100, 108, 109, 115)

[73] V. Vovk, I. Nouretdinov, V. Manokhin, and A. Gammerman, *Cross-conformal predictive distributions*, in Conformal and Probabilistic Prediction and Applications, Proc. Mach. Learn. Res. 91, ML Research Press, 2018, pp. 37–51. (Cited on p. 109)

[74] V. Vovk, I. Petej, I. Nouretdinov, V. Manokhin, and A. Gammerman, *Computationally efficient versions of conformal predictive distributions*, Neurocomputing, 397 (2020), pp. 292–308. (Cited on p. 100)

[75] V. Vovk, I. Petej, P. Toccaceli, A. Gammerman, E. Ahlberg, and L. Carlsson, *Conformal calibration*, in Conformal and Probabilistic Prediction and Applications, Proc. Mach. Learn. Res. 128, ML Research Press, 2020, pp. 84–99. (Cited on pp. 94, 100, 109, 115)

[76] E.-M. Walz, *Replication material for "Easy Uncertainty Quantification (EasyUQ): Generating predictive distributions from single-valued model output,"* 2023; available at https://github.com/evwalz/easyuq. (Cited on pp. 110, 111, 114)