Machine Learning (2026) 115:120
https://doi.org/10.1007/s10994-025-06946-3

Aggregating Distribution Forecasts from Deep Ensembles

Benedikt Schulz[1] · **Lutz Köhler**[2] · **Sebastian Lerch**[2,3]

Received: 29 October 2024 / Revised: 11 November 2025 / Accepted: 3 December 2025

Abstract
The importance of accurately quantifying forecast uncertainty has motivated much recent research on probabilistic forecasting. In particular, a variety of deep learning approaches has been proposed, with forecast distributions obtained as output of neural networks. These neural network-based methods are often used in the form of an ensemble, e.g., based on multiple model runs from different random initializations or more sophisticated ensembling strategies such as dropout, resulting in a collection of forecast distributions that need to be aggregated into a final probabilistic prediction. With the aim of consolidating findings from the machine learning literature on ensemble methods and the statistical literature on forecast combination, we address the question of how to aggregate distribution forecasts based on such 'deep ensembles'. We analyze the effect on the aggregated forecast distribution for the standard practice of averaging model output, and whether more suitable approaches are readily available. Using theoretical arguments and a comprehensive analysis on twelve benchmark data sets, we systematically compare probability- and quantile-based aggregation methods for three neural network-based approaches with different forecast distribution types as output. Our results show that combining forecast distributions from deep ensembles can substantially improve the predictive performance. We propose a general quantile aggregation framework for deep ensembles that allows for corrections of systematic deficiencies and performs well in a variety of settings, often superior compared to a linear combination of the forecast densities. Finally, we investigate the effects of the ensemble size and derive recommendations of aggregating distribution forecasts from deep ensembles in practice.

Keywords Deep ensembles · Model combination · Neural networks · Probabilistic forecasting · Distribution forecasts

Editor: Willem Waegeman.

Extended author information available on the last page of the article

Published online: 09 May 2026

1 Introduction

Probabilistic forecasts in the form of predictive probability distributions over future quantities or events aim to quantify the uncertainty in the predictions and are essential to optimal decision making in applications (Gneiting & Katzfuss, 2014; Kendall & Gal, 2017). Motivated by their superior performance on a wide variety of machine learning tasks, much recent research interest has focused on the use of deep neural networks (NNs) for probabilistic forecasting. Different approaches for obtaining a forecast distribution as the output of a NN have been proposed over the past years, including parametric methods where the NN outputs parameters of a parametric probability distribution (Lakshminarayanan et al., 2017; D'Isanto & Polsterer, 2018; Rasp & Lerch, 2018), semi-parametric approximations of the quantile function of the forecast distribution (Bremnes, 2020) and nonparametric methods where the forecast density is modeled as a histogram (Gasthaus et al., 2019; Li et al., 2021). To account for the randomness of the training process based on stochastic gradient descent methods, NNs are often run several times from different random initializations. Lakshminarayanan et al. (2017) refer to this simple to implement and readily parallelizable approach as deep ensembles (DEs). We will adopt the term deep ensemble to refer to ensembles of NN predictions in general, independent of the ensemble generating mechanism.[1] Other than random initialization, more sophisticated approaches for the generation of DEs have been proposed with dropout (Gal & Ghahramani, 2016) being a prominent example. DEs for probabilistic forecasting thus yield an ensemble of predictive probability distributions. To provide a final probabilistic forecast, the ensemble of predictive distributions needs to be aggregated to obtain a single forecast distribution.

The task of combining predictive distributions has been studied extensively in the statistical literature, see Gneiting and Ranjan (2013), Petropoulos et al. (2022, Section 2.6) and Wang et al. (2023) for overviews. Combining probabilistic forecasts from different sources has been successfully used in a wide variety of applications including economics (Aastveit et al., 2019), epidemiology (Cramer et al., 2022; Taylor & Taylor, 2021), finance (Berkowitz, 2001), signal processing (Koliander et al., 2022) and weather forecasting (Baran & Lerch, 2016, 2018), and constitutes one of the typical components of winning submissions to forecasting competitions (Bojer & Meldgaard, 2021; Januschowski et al., 2022). On the other hand, forecast combination also forms the theoretical framework of some of the most prominent techniques in machine learning such as boosting (Freund & Schapire, 1996), bagging (Breiman, 1996) or random forests (Breiman, 2001), which are based on the idea of building ensembles of learners and combining the associated predictions. Generally, the individual component models (or ensemble members) can be based on entirely distinct modeling approaches, or on a common modeling framework where the model training is subject to different input data sets of other sources of stochasticity. The latter is the case for the DEs we consider in this study. For general reviews on ensemble methods in machine learning, we refer to Dietterich (2000), Zhou et al. (2002) and Ren et al. (2016).

Averaging ensemble members via the arithmetic mean is simple and standard practice for ensemble aggregation in general, and DEs in particular. While this is a powerful and widely accepted method for aggregating single-valued point forecasts, the question how probabilistic forecasts should be combined is more involved, and has been a focus of research

[1] Note that our terminology thus differs from Lakshminarayanan et al. (2017), who introduced the term DE exclusively for ensembles of NNs generated based on random initialization and random data ordering.

interest in the literature on statistical forecasting (Gneiting & Ranjan, 2013; Petropoulos et al., 2022). The effect of averaging model output depends on the chosen output distribution and one should be aware of its consequences to choose the most suitable for the application of interest. We will focus on readily applicable aggregation methods for the combination of probabilistic forecasts from DEs. A widely used approach is the linear aggregation of the forecast distributions, which is often referred to as linear (opinion) pool (LP). An alternative is given by aggregating the forecast distributions on the scale of quantiles by linearly combining the corresponding quantile functions, an approach that is commonly referred to as Vincentization (VI; see, for example, Genest 1992) dating back to Vincent (1912). In particular, the LP and VI have been compared to each other coming from a theoretical perspective in Lichtendahl et al. (2013) and Busetti (2017), and from a practical perspective in an application of DEs for electricity price forecasting in Marcjasz et al. (2023).

In recent years, there has been a surge in interest in DEs in general, see Ganaie et al. (2022) and Mohammed and Kora (2023) for overviews. In particular, DEs of distributional forecasts have been investigated in various studies, e.g., Lakshminarayanan et al. (2017) propose alternatives to Bayesian NNs that are identical to LP in case of classification but correspond to VI in case of regression, and Rahaman and Thiery (2020) investigate the uncertainty quantification properties of probabilistic DEs. Further, Fakoor et al. (2023) develop a VI framework for aggregating quantile regression models, e.g., based on DEs, using methods from deep learning. In contrast to their work, we focus only on DEs and full distributional forecasts, which can be defined arbitrarily.

This study is motivated by and based on our work in Schulz and Lerch (2022), where we use DEs to statistically postprocess probabilistic forecasts for the speed of wind gusts and propose a common framework of NN-based probabilistic forecasting methods with different types of forecast distributions. In the following, we apply a two-step procedure by first generating an ensemble of probabilistic forecasts and then aggregating them into a single final forecast, which matches the typical workflow of forecast combination from a forecasting perspective. Alternatively, it is also possible to incorporate the aggregation procedure directly into the model estimation (Fakoor et al., 2023).

1.1 Contribution

The main aim of our work is to consolidate findings from the statistical and machine learning literature on forecast combination and ensembling for probabilistic forecasting. Our study is the first to systematically investigate and compare the two central aggregation schemes for probabilistic forecasts, namely, probability (LP) and quantile aggregation (VI), applied to DEs. In addition to the LP and the standard approach to VI, we propose a novel general VI approach that is able to correct for systematic errors such as biases and miscalibration in the aggregated forecasts. Using theoretical arguments and a comprehensive evaluation on machine learning benchmark data sets, we analyze the aggregation methods with different ways to characterize the corresponding forecast distributions and different ensembling strategies. Our findings include advice on the choice of the most suitable aggregation method based on theoretical arguments, tailored to chosen type of distribution forecasts, and the application for different NN methods, ensembling strategies and data sets of varying complexity.

1.2 Outline

The remainder of the paper is organized as follows. Section 2 introduces the central concepts of probabilistic forecasting and the forecast aggregation methods. Three NN-based methods for probabilistic forecasting are presented in Sect. 3 along with a discussion of how the different aggregation methods can be used to combine the corresponding predictive distributions of an ensemble of such forecasts. Section 3 ends with a short introduction of the strategies used for the generation of DEs. In Sect. 4, we apply the aggregation methods in a comprehensive case study. Following the introduction of the study design and evaluation methods, we provide an in-depth analysis for two selected pairs of data set and ensembling strategy first, then we compare the performance for all data sets and ensembling strategies. Section 5 concludes with a discussion. Code with implementations of all methods is available online (https://github.com/benediktschulz/ADDE).

2 Combining Probabilistic Forecasts

For the mathematical definition of the aggregation methods, we need to introduce the mathematical notation first. Given $n \geq 2$ individual probabilistic forecasts we aim to aggregate, we will denote their cumulative distribution functions (CDFs) by $F_1, \ldots, F_n$ and their quantile functions[2] by $Q_1, \ldots, Q_n$. The aggregation methods introduced in the following will typically assign weights w_1, ..., w_n to the individual forecast distributions. As we apply the aggregation methods to forecasts produced by the same data-generating mechanism based on a DE, we do not expect systematic differences between the individual forecasts. Therefore, we will focus on equal weighting of ensemble members in the remainder, while introducing the aggregation methods in general form for completeness.

Before formally introducing the LP and VI methods for aggregating distribution forecasts, detached from the application on DEs, we briefly comment on the main concepts of probabilistic forecasting, which are fundamental to the aggregation methods presented in this section and the forecast-generating NNs in Sect. 3.

2.1 Central Concepts of Probabilistic Forecasting

Probabilistic forecasts given in the form of predictive probability distributions for future quantities or events aim to quantify the uncertainty inherent to the prediction. In our evaluation of predictive performance, we will follow the principle of Gneiting et al. (2007) that a probabilistic forecast should aim to maximize sharpness subject to calibration. Calibration refers to the statistical consistency between the forecast distribution and the observation, whereas sharpness is a property of the forecast alone and refers to the degree of forecast uncertainty. A forecast is said to be sharper, the smaller the associated uncertainty. Typical deviations from calibration are overconfidence (or underdispersion), that is, a lack of spread in the forecast distribution, and underconfidence (or overdispersion), that is, too

[2] We refer to Wasserman (2004 Section 2.3) for an introduction to quantile functions.

much spread.[3] Quantitatively, calibration and sharpness can be assessed simultaneously using proper scoring rules, which assign a penalty to a forecast-observation pair and is minimized in expectation when we forecast the underlying true distribution (Gneiting & Raftery, 2007). In the remainder of the paper, we will focus on the widely used continuous ranked probability score (CRPS; Matheson and Winkler 1976). Proper scoring rules such as the CRPS are not only used for forecast evaluation but also provide valuable tools for estimating model parameters, which is referred to as optimum score estimation (Gneiting & Raftery, 2007). Section 4.2 explains how we use the concepts introduced above to assess the forecasts in the case study.

2.2 Linear Pool (LP)

The most widely used approach for forecast combination is the LP, which is the arithmetic mean of the individual forecasts (Stone, 1961). For probabilistic forecasts, the LP is calculated as the (in our case equally) weighted average of the predictive CDFs and results in a mixture distribution. Equivalently, the LP can be calculated by averaging the probability density functions (PDFs). We define the predictive CDF of the LP via

$$F_w(z) := \sum_{i=1}^{n} w_i F_i(z), \quad z \in \mathbb{R}, \tag{1}$$

where $w_i \geq 0$ for $i = 1, \ldots, n$ with $\sum_{i=1}^{n} w_i = 1$. Note that the weights need to sum up to 1 to ensure that F_w yields a valid CDF. Hence, our assumption of equal weights results in the choice of $w_i = \frac{1}{n}$ for $i = 1, \ldots, n$ in (1).

The LP has some appealing theoretical properties and has been the prevalent forecast aggregation method over the last decades. For example, Lichtendahl et al. (2013) and Abe et al. (2022) show that the score of the LP forecast is at least as good as the average score of the individual components in terms of different proper scoring rules. However, there are disadvantages to the use of the LP that is known to have suboptimal properties when aggregating probabilities, since a linear combination of probability forecasts results in less sharp and more underconfident forecasts (Ranjan & Gneiting, 2010). Gneiting and Ranjan (2013) extend this result to the general case of predictive distributions by showing that in case of distribution forecasts sharpness decreases and dispersion increases. In particular, a (non-trivial) combination of calibrated forecasts is not calibrated anymore. In the context of DEs, these downsides have also been observed in recent studies (Rahaman & Thiery, 2020; Wu & Gales, 2021).

Figure 1 illustrates the effect of forecast combination via the LP for two exemplary normal distributions. The aggregated forecast is a bimodal distribution, which is less confident, i.e., more spread out, than the individual members. At last, we want to comment on the special case of overconfident individual member forecasts, which are often observed in NN

[3] In contrast to the notion of confidence, which is rather intuitive and more commonly referred to in the machine learning literature, over- and underdispersion are rigorously defined in the statistical forecasting literature (Gneiting & Ranjan, 2013). While every over- or underconfident forecast is considered to be under- or overdispersed respectively, the opposite may not always be appropriate. Still, we will use the terminology synonymously and stick to the notion of confidence.

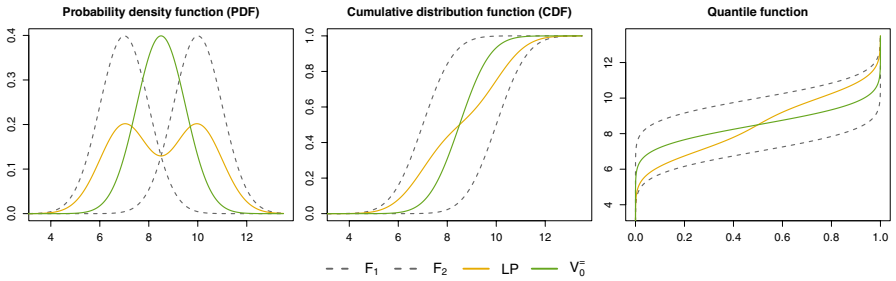

Fig. 1 Exemplary aggregation contrasting probability (LP) and quantile averaging ($\mathrm{V}_0^{=}$): PDF, CDF and quantile function of two normally distributed forecasts F_1 and F_2 ($\mu_1 = 7$, $\mu_2 = 10$, $\sigma_1 = \sigma_2 = 1$) together with those of the aggregated forecasts

applications, e.g., when models are overfitted due to sparse data or overparameterization. Empirical evidence suggests that the increase in spread by the LP typically improves predictive performance in such cases. However, in case of calibration or underconfidence, the forecasts become less well calibrated.

2.3 Vincentization (VI)

While the LP aggregates the forecasts on a probability scale, VI performs a quantile-based linear aggregation (Ratcliff, 1979; Genest, 1992). We extend the standard VI framework[4] by defining the VI quantile function via

$$Q_w^a(p) := a + \sum_{i=1}^{n} w_i Q_i(p), \quad p \in [0,1]\,, \tag{2}$$

where $a \in \mathbb{R}$ and $w_i \geq 0$ for $i = 1, \ldots, n$. In contrast to the LP, the weights do not need to sum to 1 and only their non-negativity is required to ensure the monotonicity of the resulting quantile function Q_w^a. Again, we will consider only the case of equal weights, which here translates to a free weight parameter $w_i = w_0 \geq 0$ for $i = 1, ..., n$. Further, a real-valued intercept a is added to the aggregated quantile functions to correct for systematic biases.

Given equal weights, we consider four different variants of VI. First, with weights that sum up to 1 and no intercept, that is, $a = 0$ and $w_0 = \frac{1}{n}$, which is referred to by $\mathrm{V}_0^{=}$. Similar to the LP, $\mathrm{V}_0^{=}$ does not require the estimation of any parameters. Further, we consider VI variants where we estimate the parameters a and w_0 both independently (while the other is fixed at the values of $\mathrm{V}_0^{=}$) and also simultaneously, resulting in the three variants $\mathrm{V}_a^{=}$ (where $w_0 = \frac{1}{n}$ and a is estimated), V_0^{w} (where $a = 0$ and w_0 is estimated) and V_a^{w} (where both a and w_0 are estimated). The parameters are estimated minimizing the CRPS following the optimum scoring principle. The standard procedure for training machine learning models where the available data is split into training, validation and test data sets offers a natural choice for estimating the combination parameters. Given NN models estimated based on

[4] To the best of our knowledge, VI is usually only applied with non-negative weights that sum up to 1 and without an intercept (e.g., Wolffram 2023). Exceptions include Wolffram (2021) and related, unpublished simulation experiments by Anja Mühlemann (University of Bern, 2020, personal communication).

the training set (where the validation set is used to determine hyperparameters), we estimate the coefficients of the VI approaches separately in a second step based on the validation set, which can be seen as a post-hoc calibration step (Guo et al., 2017). During this second step, the component models with quantile functions $Q_i, i = 1, \ldots, n$, are considered fixed and we only vary the combination parameters in (2). In the following, we will restrict our attention to fixed training and validation sets, but an extension of the approach described here to a cross-validation setting is straightforward. Table 1 provides an overview of the abbreviations and important characteristics of the different forecast aggregation methods we will consider below.

While the effects of the LP on calibration and dispersion have been proven mathematically, no such strong statements for the VI exist. Lichtendahl et al. (2013), who compare the theoretical properties of the LP and $\mathrm{V}_0^=$, note that the aggregated predictive distributions both yield the same mean but the $\mathrm{V}_0^=$ forecasts are sharper, that is, the $\mathrm{V}_0^=$ predictive distribution has a variance smaller or equal to that of the LP. An illustration of these characteristics can bee seen in Fig. 1. Visually, we can conclude that both have the same mean due to symmetry arguments and that the bimodal PDF of the LP is wider than that of $\mathrm{V}_0^=$, i.e., $\mathrm{V}_0^=$ is sharper than the LP. Related work in the statistical literature includes comparisons to the LP which demonstrate that VI tends to perform better than the LP (Lichtendahl et al., 2013; Busetti, 2017).

Figure 2 illustrates the effects of VI and the influence of the individual VI parameters in the exemplary case of the two normal distributions from Fig. 1. First, we note that the intercept *a* only has an effect on the location of the resulting aggregated distribution, i.e., the predictive density of $\mathrm{V}_a^=$ is shifted along the *x*-axis. In contrast, the weight w_0 has an effect on both the location and the spread. If it is larger than $1/n$ (as in Fig. 2), the spread increases compared to the average spread of the individual forecasts. Other than that, it decreases for values smaller than $1/n$. However, if the shared weight is not equal to 1, it also shifts the location of the distribution, as we can see for V_0^w in Fig. 2. At last, we can simultaneously control both the location and the spread of the aggregated distribution by choosing a weight and an intercept using V_a^w.

Table 1 Overview of the aggregation methods for probabilistic forecasts, with F_i and Q_i denoting the predictive CDFs and quantile functions of the individual components models

Abbr	Scale	Formula	Parameters	Estimation
LP	Probability	$F_w = \frac{1}{n}\sum_{i=1}^n F_i$	–	–
$\mathrm{V}_0^=$	Quantile	$Q_w = \frac{1}{n}\sum_{i=1}^n Q_i$	–	–
$\mathrm{V}_a^=$	Quantile	$Q_w = \frac{1}{n}\sum_{i=1}^n Q_i + a$	$a \in \mathbb{R}$	CRPS
V_0^w	Quantile	$Q_w = w_0 \sum_{i=1}^n Q_i$	$w_0 \geq 0$	CRPS
V_a^w	Quantile	$Q_w = w_0 \sum_{i=1}^n Q_i + a$	$w_0 \geq 0, a \in \mathbb{R}$	CRPS

The column 'Parameters' indicates which parameters are estimated based on data, following the procedure described in Sect. 2.3

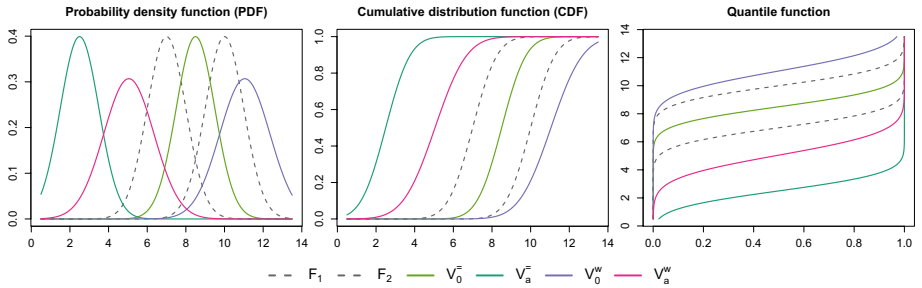

Fig. 2 Exemplary aggregation illustrating the effect of the parameters in the extended quantile aggregation (VI) framework: PDF, CDF and quantile function of the two normally distributed forecasts from Fig. 1 together with those of the forecasts aggregated via the VI methods from Sect. 2.3, with intercept $a = -6$ and weight $w_0 = 0.65$

3 Distribution Forecasts from Deep Ensembles

In this section, we will specify how we generate distribution forecasts from DEs. First, we will introduce the three NN approaches used to generate distributional forecasts and how to apply the aggregation methods presented in Sect. 2. Then, we will present the different ensembling strategies we consider for the generation of DEs.

3.1 Neural Network Methods for Probabilistic Forecasting

In the context of probabilistic wind gust prediction, Schulz and Lerch (2022) propose a framework for NN-based probabilistic forecasting that encompasses different approaches to obtain distribution forecasts as the output of a NN. The general framework is illustrated in Fig. 3 and forms the basis of our work here. In this section, we briefly introduce three NN variants and refer to Schulz and Lerch (2022) for details.[5]

While these three variants differ in their characterization of the forecast distribution and the loss function employed in the NN, their use in practice shares a common methodological feature that constitutes the main motivation for our work here. As discussed in the introduction, extant practice in NN-based forecasting often relies on DEs. This raises the question of how the distribution forecast from the three NN variants can be combined using the aggregation methods described in Sect. 2, and how this is connected with the standard practice of averaging model output.

3.1.1 Distributional Regression Network (DRN)

In the distributional regression network (DRN) approach, the forecasts are issued in the form of a parametric distribution. Under the parametric assumption, the predictive distribution is characterized by the distribution parameter (vector). Different variants of the DRN approach have been proposed over the past years and can be traced back to at least Bishop (1994) and Nix and Weigend (1994). Lakshminarayanan et al. (2017) and Rasp and Lerch

[5] Note that other types of distributional forecasts such as normalizing flows (Kobyzev et al., 2021) exist. However, we do not consider them here in the interest of brevity and since the approaches discussed in Sect. 3.1 share appealing properties with regards to the aggregation methods.

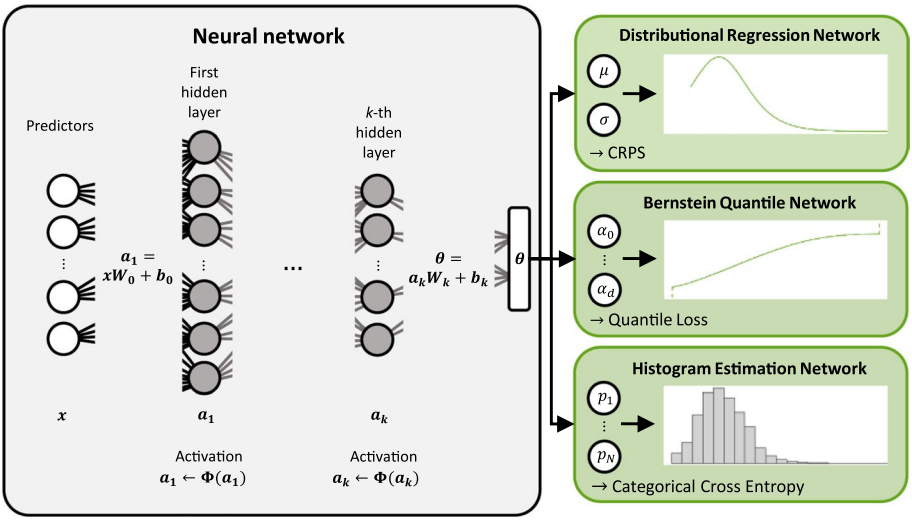

Fig. 3 Graphical illustration of the general framework for NN-based probabilistic forecasting

(2018) use a normal distribution with parameter vector (μ, σ), Schulz and Lerch (2022) use a zero-truncated logistic distribution with parameter vector (μ, σ), where for both distributions $\mu \in \mathbb{R}$ is the location and $\sigma > 0$ the scale parameter, and Bishop (1994) and D'Isanto and Polsterer (2018) use a mixture of normal distributions. To estimate the parameters of the NN, proper scoring rules such as the CRPS (Rasp & Lerch, 2018; D'Isanto & Polsterer, 2018; Schulz & Lerch, 2022) or the negative log-likelihood (Lakshminarayanan et al., 2017) serve as custom loss functions. Extensions of the DRN approach to other parametric families are generally straightforward provided that analytical closed-form expressions of the selected loss function are available (for example, Ghazvinian et al. 2021; Chapman et al. 2022).

For VI, distributions from location-scale families form a special case that is equivalent to the standard practice of averaging model output. Given a CDF $F_{(0)}$, a distribution is said to be an element of a location-scale family if its CDF F satisfies

$$F(z; \mu, \sigma) = F_{(0)}\left(\frac{z-\mu}{\sigma}\right), \quad z \in \mathbb{R},$$

where $\mu \in \mathbb{R}$ denotes the location and $\sigma > 0$ the scale parameter. Popular examples include the normal and logistic distributions. For location-scale families, VI is shape-preserving, which means that if the individual forecasts are elements of the same location-scale family, the aggregated forecast is as well (Thomas & Ross, 1980). Further, the parameters of the aggregated forecast μ^{VI} and σ^{VI} are given by the weighted averages of the individual parameters μ_i and σ_i, $i = 1, ..., n$, together with the intercept a in case of the location parameter, that is,

$$\mu^{\text{VI}} = a + \sum_{i=1}^{n} w_i \mu_i \overset{(\text{V}\bar{0})}{=} \frac{1}{n}\sum_{i=1}^{n} \mu_i = \overline{\mu}, \quad \text{and} \quad \sigma^{\text{VI}} = \sum_{i=1}^{n} w_i \sigma_i \overset{(\text{V}\bar{0})}{=} \frac{1}{n}\sum_{i=1}^{n} \sigma_i = \overline{\sigma}, \quad (3)$$

where the second equalities each hold under our assumption of $V_0^{=}$. Note that the exemplary aggregation in Fig. 2 falls under this case. To summarize, the standard practice of averaging model output from DRN approaches is equivalent to $V_0^{=}$ for location-scale families, such as the normal and logistic distribution. Extension towards the other VI approaches can be achieved by direct transformation of the distributional parameters. Unlike VI, the LP results in a wide-spread, multi-modal distribution, and is thus not shape-preserving for location-scale families, as exemplified in Fig. 1. Both Lakshminarayanan et al. (2017) and Rasp and Lerch (2018) generate DEs based on random initialization. While Rasp and Lerch (2018) simply combine the forecasts by averaging the distribution parameters, Lakshminarayanan et al. (2017) use a more involved approach in their regression experiments that corresponds to a quantile aggregation procedure, but does not directly fit into our VI framework introduced above. Since the normal distribution is a location-scale family, parameter averaging is equivalent to $V_0^{=}$. For the application in Sect. 4, we will employ a normal distribution, i.e., we obtain the VI forecasts via (3), i.e., the standard practice of averaging model output. To evaluate the LP forecasts, we draw a random sample of size 1,000 from the mixture distribution by first randomly choosing an ensemble member and then generating a random draw from the corresponding distribution.

3.1.2 Bernstein Quantile Network (BQN)

Bremnes (2020) proposes a semi-parametric extension of the DRN approach we refer to as Bernstein quantile network (BQN). The probabilistic forecast is given in form of the quantile function Q, which is modeled as a linear combination of Bernstein polynomials, that is,

$$Q(p) := \sum_{j=0}^{d} \alpha_j B_{jd}(p), \quad p \in [0,1],$$

where $\alpha_0 < \cdots < \alpha_d$ and B_{jd} is the j-th basis Bernstein polynomial of degree $d \in \mathbb{N}$, $j = 0, \ldots, d$. The basis coefficients $\alpha_0, \ldots, \alpha_d$, which define the predictive distribution, are obtained as output of the NN. The parameters of the NN are estimated by minimizing the quantile loss evaluated at pre-defined quantile levels. Note that the support of the forecast distribution is equal to $[\alpha_0, \alpha_d]$ and therefore bounded.

To aggregate ensembles of BQN forecasts, Bremnes (2020) and Schulz and Lerch (2022) average the individual basis coefficient values across ensemble members. Resembling the shape-preservation for location-scale families in case of DRN, this is equivalent to $V_0^{=}$, which is obvious from the quantile function of the general case of VI for BQN forecasts,

$$Q_w(p) = a + \sum_{i=1}^{n} w_i \left(\sum_{j=0}^{d} \alpha_{ij} B_{jd}(p) \right) = \sum_{j=0}^{d} \left(a + \sum_{i=1}^{n} w_i \alpha_{ij} \right) B_{jd}(p), \quad p \in [0,1],$$

where α_{ij} is the coefficient of the j-th basis polynomial of the i-th ensemble member, $i = 1, \ldots, n$, $j = 0, \ldots, d$. Note that we can move the intercept a into the summation, as the sum of the Bernstein basis polynomials equals 1. Further, we see that we only need to add the intercept to the averaged coefficients to obtain the vincentisized BQN forecast.

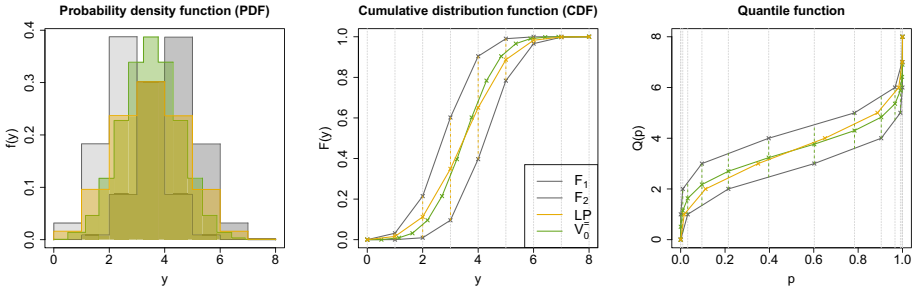

Fig. 4 Exemplary aggregation of HEN forecasts via probability (LP) and quantile averaging ($\text{V}\overline{0}$): PDF, CDF and quantile function of two HEN forecasts F_1 and F_2 together with those of the aggregated forecasts. The dashed vertical lines indicate the binning with respect to F_1, F_2 and F_w for the CDF plot and with respect to Q_w in the quantile function plot

Analogous to DRN for location-scale families, the standard practice is equivalent to quantile averaging.

Since a closed form of the CDF or density of a BQN forecast is not readily available, the LP cannot be expressed in a similar fashion. Analogous to DRN, the evaluation of the LP forecasts will therefore be based on a random sample of size 1,000 drawn from the aggregated distribution. Here, the inversion method allows to sample from the individual BQN forecasts. Further, the VI forecasts are evaluated based on a sample of 99 equidistant quantiles.[6]

3.1.3 Histogram Estimation Network (HEN)

The last method considered here is the histogram estimation network (HEN) which divides the support of the target variable in $N \in \mathbb{N}$ bins and assigns each bin the probability for the observation falling in that bin. Variants of this approach have been proposed in a variety of applications (for example, Gasthaus et al. 2019; Li et al. 2021). Mathematically, the HEN forecast is given by a piecewise uniform distribution. Let $b_0 < \cdots < b_N$ denote the edges of the bins $I_\ell = [b_{\ell-1}, b_\ell)$ with probabilities p_ℓ, $\ell = 1, \ldots, N$, where it holds that $\sum_{\ell=1}^{N} p_\ell = 1$. The CDF of a HEN forecast is then given by the piecewise linear function

$$F(z) = \sum_{\ell=1}^{N} \left(p_\ell \cdot \frac{\tilde{z} - b_{\ell-1}}{b_\ell - b_{\ell-1}} \cdot \mathbb{1}\{b_{\ell-1} \leq z\} \right) \quad \text{with} \quad \tilde{z} := \max\left(b_{\ell-1}, \min\left(b_\ell, z\right)\right), \quad z \in \mathbb{R}.$$

Note that a piecewise linear CDF corresponds to a piecewise linear quantile function and a piecewise constant PDF that resembles a histogram. Figure 4 illustrates the shape of these functions for exemplary HEN forecasts.

We here follow Schulz and Lerch (2022) by considering fixed bins and estimate only the corresponding probabilities as output of the NN. In contrast, we here standardize the target

[6] The numbers of samples and quantiles were chosen based on simulation experiments and theoretical considerations. Compared to random samples from the forecast distributions, a smaller number of equidistant quantiles is required to achieve approximations of the same accuracy, see Krüger et al. (2021) and references therein for a discussion of sample-based estimation of the CRPS.

variable and define the bin edges based on quantiles of the standard normal distribution. For prediction (and evaluation), the bin edges can easily be transformed back to the original scale of the target variable. As for DRN, the NN can be trained via CRPS minimization or maximum likelihood. Here, we use the latter, which corresponds to minimizing the categorical cross-entropy, a standard approach for classification tasks in machine learning.

Regarding the aggregation of HEN forecasts in case of fixed bins, the LP is equivalent to averaging the bin probabilities since

$$\begin{aligned} F_w(z) &= \sum_{i=1}^{n} w_i \left[\sum_{\ell=1}^{N} \left(p_{i\ell} \frac{\tilde{z} - b_{\ell-1}}{b_\ell - b_{\ell-1}} \mathbb{1}\{b_{\ell-1} \le z\} \right) \right] \\ &= \sum_{\ell=1}^{N} \left[\left(\sum_{i=1}^{n} w_i p_{i\ell} \right) \frac{\tilde{z} - b_{\ell-1}}{b_\ell - b_{\ell-1}} \mathbb{1}\{b_{\ell-1} \le z\} \right], \end{aligned}$$

where $z \in \mathbb{R}$ and $p_{i\ell}$ is the probability of the ℓ-th bin for the i-th ensemble member, $i = 1, \ldots, n$, $\ell = 1, \ldots, N$. An exemplary application of the LP for an approach akin to HEN forecasts in a stacked NN can be found in Clare et al. (2021). By contrast to the LP, the VI approach exhibits a particular advantage for HEN forecasts in that it results in a finer binning than the individual HEN models. To illustrate this effect, we note that the quantile function is a piecewise linear function with edges depending on the accumulated bin probabilities, that is, $p^*_\ell := \sum_{m=1}^{\ell} p_m$, $\ell = 1, \ldots, N$. In mathematical terms, the quantile function is given for $p \in [0, 1]$ by

$$Q(p) = b_0 + \sum_{\ell=1}^{N} (b_\ell - b_{\ell-1}) \left(\frac{\tilde{p} - p^*_{\ell-1}}{p^*_\ell - p^*_{\ell-1}} \cdot \mathbb{1}\{p^*_{\ell-1} \le p\} \right),$$

where $\tilde{p} := \max(p^*_{\ell-1}, \min(p^*_\ell, p))$. Therefore, the resulting VI quantile function is a piecewise linear function with one edge for each accumulated probability of the individual forecasts. As the forecast probabilities differ for each member of the DE, the associated quantile functions are subject to a different binning. Since the set of edges of the aggregated VI forecast is given by the union of all individual edges, this leads to a smoothed final forecast distribution with a finer binning than the individual model runs that differs for every forecast case, and eliminates the potential downside of too coarse fixed bin edges. Figure 4 illustrates the effects of the LP and $\mathrm{V}_0^=$ for two exemplary HEN forecasts, where the binning of the $\mathrm{V}_0^=$ forecast distribution is finer than that of the individual forecasts and of the LP.

Altogether, the standard practice of averaging model output is equivalent to the LP for the HEN approach, unlike the other two cases. Alternatively, we can use VI to refine the initial binning and obtain more flexible forecast distributions.

3.2 Ensembling Strategies

Various methods have been proposed for the generation of DEs each addressing different aspects of uncertainty in the training process. For this study, we picked the most common DE approaches, the underlying ideas of which we briefly present in the following. Implementation details are deferred to Appendix A.

3.2.1 Naive Ensemble

Due to the random initialization of the NN weights and the stochastic gradient descent algorithm, the process of training a standard NN is subject to stochasticity and therefore multiple training runs will result in different weight estimates. Hence, a straightforward way to generate a DE is to simply train several models based on different random initializations, which we refer to as naive ensemble. It is not only simple to implement, but has also been shown to result in improved predictive performance (see, e.g., Lakshminarayanan et al. 2017; Fort et al. 2019).

3.2.2 Bagging

Bagging ('bootstrap aggregating') is one of the earliest ideas for generating ensembles (Breiman, 1996) and forms the basis of other ensembling-based methods such as random forests (Breiman, 2001). Bagging generates multiple models such that each is based on a different bootstrapped sample of the original training data. For NNs, the ensemble models thus do not only differ due to the random initialization and stochastic gradient descent, but also due to the bootstrapped training sets.

3.2.3 BatchEnsemble

Although parallelizable, one disadvantage of the naive ensemble and bagging is that the computational costs increase linearly with the ensemble size as no modifications are applied to make the ensemble generation more efficient. To address this, Wen et al. (2020) introduce BatchEnsemble, an efficient ensemble method with parallel mini-batch training and shared weights, which reduces the computational costs significantly and performs comparably to the naive ensemble in their original study. Note that a variety of related techniques that enable the efficient generation of deep ensembles has been proposed. For example, the approaches introduced in Huang et al. (2017); Durasov et al. (2021); Havasi et al. (2021); Turkoglu et al. (2022); Mühlematter et al. (2024) might serve as alternatives.

3.2.4 Dropout Variants (MC Dropout, Variational Dropout, Concrete Dropout)

A widely used technique for regularization, that can also be used for ensembling, is dropout (Srivastava et al., 2014), which operates by randomly dropping neurons with a given probability. We consider three variants of dropout that differ in the choice of the dropout rate: Monte Carlo (MC) dropout treats the (overall) dropout rate as additional hyperparameter that is chosen beforehand, variational dropout learns the dropout rate during training (Kingma et al., 2015), and concrete dropout improves over the former by adapting the rate-learning process allowing that the rate is learned directly per layer, and by using a heteroscedastic variance (Gal et al., 2017). Here, we apply dropout both during training and inference, where we generate ensembles by repeatedly predicting with one base model (Gal & Ghahramani, 2016), a mechanism that inherently differs from the previous strategies. In the following, we will differentiate between multi- and base-model approaches.

3.2.5 Bayesian Neural Networks

The final method for ensemble generation is based on Bayesian neural networks (BNNs; Lampinen and Vehtari 2001; Jospin et al. 2022), which account for the uncertainty in the learning task using Bayesian ideas. Instead of learning the weights of the NN as deterministic values, the distribution of the weights is modelled. By sampling from the learned distributions, we can generate ensembles of predictions.[7] As for the dropout models, we train one base model that is used for the generation of the DE, i.e., this strategy is also a base-model approach. Note that the various dropout variants can also be interpreted as Bayesian approaches, see, e.g., Gal and Ghahramani (2016).

4 Case Study

We compare the performance of the five aggregation methods for each of the three NN variants and seven ensembling strategies on various data sets, which comprise a data set on wind speed forecasting (Schulz & Lerch, 2024), two simulated data sets (Li et al., 2021) and nine open-source machine learning benchmark data sets (see, e.g., Gal and Ghahramani 2016; Lakshminarayanan et al. 2017).[8]

As a detailed analysis for all combinations of data sets, ensembling strategies and NN variants is too cumbersome, we first provide an in-depth analysis for two selected cases and then investigate results over all combinations on a higher level. The in-depth analysis is carried out to highlight reoccurring effects of the aggregation methods on the predictive performance in terms of scores, calibration and sharpness. While we observe similar effects within the application to certain data sets and ensembling strategies, the properties of DE and aggregation differ for each data set and ensembling strategy. Hence, we also investigate the aggregation methods over all combinations, where we draw conclusions on the effects of aggregation based on the underlying characteristics of the DE that differ over the cases.

4.1 Study Setup

To account for the uncertainty in data sampling, we generate different partitions of each data set by splitting them multiple times in training, validation and test sets. We follow Gal and Ghahramani (2016) and use 20 random partitions of the data sets except for the Protein and Wind data sets, where the number is restricted to 5 due to the size. For the 9 machine learning benchmark data sets, the training, validation and test set each make up 72%, 18% and 10%, respectively. For the 5 partitions of the Wind data set, we use one of the calendar years as test set for each of the 5 partitions and 20% of the remaining data for validation, as for the other data sets. The two synthetic data sets, which are referred to as Scenarios 1 and 2, are adopted from Li et al. (2021) and described in more detail in Appendix C. For Scenarios

[7] Note that in our case, those sampled predictions are not deterministic point forecasts, but full probability distributions. Therefore, the situation at hand differs from standard Bayesian forecasting setups, where one may approximate the posterior predictive distribution via Monte Carlo methods, see, e.g., Krüger et al. (2021).

[8] An earlier version of the manuscript only included the two simulation studies and the wind gust data. Following the inclusion of the benchmark data sets, we decided to keep the wind data and simulations to have a larger variety of data sets.

Table 2 Overview of the data set sizes from the case study in Sect. 4

Data set	Total	Training	Validation	Testing	Features
Wind	378,833	252,946	63,237	62,650	67
Scenarios 1–2	7,000	5,000	1,000	1,000	5
Protein	45,730	32,925	8,232	4,573	9
Naval	11,934	8,592	2,149	1,193	16
Power	9,568	6,888	1,723	957	4
Kin8nm	8,192	5,898	1,475	819	8
Wine	1,599	1,151	288	160	11
Concrete	1,030	741	186	103	8
Energy	768	552	139	77	8
Boston	506	364	91	51	13
Yacht	308	222	55	31	6

Table 3 Design parameters for the case study in Sect. 4

Factor	Size	Values
Forecast distribution	3	DRN, BQN, HEN
Aggregation method	5	LP, $V_0^=$, $V_a^=$, V_0^w, V_a^w
Ensembling strategy	7	Naive ensemble, bagging, BatchEnsemble, MC dropout variational dropout, concrete dropout, Bayesian NN
Data set	12	Wind, Scenarios 1–2, Protein, Naval, Power, Kin8nm Wine, Concrete, Energy, Boston, Yacht
Ensemble size	10	2, 4,..., 20
Partition	5/20	1, 2,..., 5 for Wind and Protein, 1, 2,..., 20 otherwise

In total, we obtain 88,200 forecast configurations for DEs and 220,500 for aggregation

1 and 2, we do not create partitions from the same data set but instead by repeated random generation. For each partition, we then calculate 20 ensemble members, which are used to build ensembles of sizes 2, 4, ..., 20. In the interest of computational requirements, we use steps of size 2 and restrict the maximum ensemble size to 20. Further, previous tests have shown that increasing the ensembles above a size of 20 results only in a marginal improvement in the performance. A summary of the data is provided in Table 2.

For each combination of data set, ensembling strategy and NN variant, we perform hyperparameter tuning and choose one combination of hyperparameters, which is then used for all partitions. Over a set of pre-defined choices, we chose the best-performing models on the validation sets of the first two random partitions. More details on the tuning procedure and the chosen hyperparameters are provided in Appendix B. Table 3 provides an overview of all the factors in Sect. 4.

4.2 Evaluation Methods

The main concepts for the evaluation of probabilistic forecasts have already been introduced in Sect. 2.1. Recall that the central paradigm of probabilistic forecasting is to maximize the sharpness subject to calibration. We quantify the forecast performance using the CRPS (Matheson & Winkler, 1976), a proper scoring rule that rewards calibration and sharpness. To compare competing forecasting methods with respect to a reference based on a

proper scoring rule, we calculate the associated skill score, which is defined as the relative improvement over the reference method – here, the continuous ranked probability skill score (CRPSS). Note that the skill score is computed by first averaging the mean scores and then calculating the relative improvement. In contrast to proper scoring rules, skill scores are positively oriented with 1 indicating optimal predictive performance, 0 no improvement over the reference and a negative skill a decrease in performance. In the case study, we will assess the improvement from aggregation by comparison with the corresponding average of the DE. Hence, the CRPSS will be calculated based on the CRPS of the aggregated forecast and the average CRPS of the individual NNs as reference.[9] In addition, we generate quantile-based prediction intervals (PIs) to assess the calibration of the forecast distributions via the empirical coverage, and the sharpness via the length of the PIs. If a forecast is well-calibrated, the empirical coverage should resemble the nominal coverage, and a forecast is the sharper, the smaller the length of the PI. The nominal level of the PIs is a tuning parameter for evaluation, which we choose to be 90% for the case study in Sect. 4. If not noted otherwise, the evaluation is based on the mean values computed for each combination of the factors in Table 3, in particular, we also calculate the mean value for each partition separately.

Further, we qualitatively assess calibration through histograms of the probability integral transform (PIT), which is defined as the value of the CDF at the observation.[10] A probabilistic forecast is (well-)calibrated, if the PIT is uniformly distributed, resulting in a flat histogram. A U-shaped PIT histogram indicates overconfidence, whereas a hump-shaped histogram indicates underconfidence. In mathematical terms, the dispersion of a predictive distribution can be defined as the variance of the PIT. For a calibrated forecast, the variance should equal that of an uniform distribution, i.e., $1/12 \approx 0.0833$. A variance smaller than 1/12 corresponds to underconfidence (or overdispersion), a variance larger than 1/12 to overconfidence (or underdispersion). For further background and details on the assessment of probabilistic forecasts, we refer to Schulz and Lerch (2022, Appendix A) and the references therein.

At last, we briefly address how we measure diversity within an ensemble of predictive distributions. We differentiate between three different measures, namely, in terms of the location, prediction uncertainty and performance. As measure of the location diversity, we use the standard deviation of the mean values of the individual member's predictive distributions. We proceed analogously for the prediction uncertainty by using the PI length instead of the mean, or the CRPS for the performance respectively. To be able to compare the ensemble diversity among different settings, we apply a (z-score-)standardization of the underlying quantity (separately for each configuration from Table 3) in a preliminary step before calculating the standard deviation over the ensemble members.

[9] Note that this does not correspond to the mean improvement over the individual forecasts. However, averaging the median skill scores of the individual ensemble member predictions over the repetitions of the simulations yields qualitatively analogous results (not shown).

[10] Technically, we here use the unified PIT, a generalization proposed in Vogel et al. (2018), due to the format of some of the aggregated forecast distributions.

4.3 In-Depth Analysis

For the in-depth analysis, we pick two cases that highlight potential effects of aggregation on DE forecasts of the three NN variants. The two cases have been chosen as they are typical for reoccurring situations in which certain patterns arise. The first example is based on DEs generated with Bayesian NNs and the Kin8nm data set, which has a size of 8,192. We will refer to those as Bayesian deep ensembles, however, recall that the individual ensemble members here still are predictive distributions. The second example is based on another ensemble strategy and a much smaller data set with only 506 samples, namely, Bagging and the Boston data set.

4.3.1 Kin8nm and Bayesian Deep Ensembles

First, we visually inspect the calibration of both the DE forecasts and the aggregated forecasts via the PIT histograms in Fig. 5. We find that none of the NN variants generate calibrated forecasts. Both the DRN and HEN forecasts are overconfident resulting in a U-shaped histogram, while the BQN forecasts result in a wave-shaped histogram. Comparing with the aggregated forecasts, we find that the histogram shapes change systematically. For the LP, we obtain forecasts that are underconfident but to a different extent for all three NN variants. This effect is also observed for all other cases, as it aligns with the theoretical property that the LP increases the dispersion, i.e., less confidence, with respect to the individual ensemble members. All VI forecasts of DRN and HEN result in flat histograms indicating calibrated forecasts. For BQN, the VI forecasts have the same wave-like shape as the individual forecasts and seem to be a bit underconfident. Hence, they were not able to correct for this specific kind of miscalibration. In general, miscalibration different from the typical under- and overconfidence are not corrected by aggregation. Between the VI variants, we do not observe systematic differences.

Following the visual inspection of calibration, we quantitatively analyze the predictive performance dependent on the size of the ensemble via Fig. 6. Unsurprisingly, all aggregation methods improve upon the individual forecasts in terms of the CRPS, for each NN variant to a different extent. The ranking is identical over the different NN variants with the VI approaches using parameter estimation performing best, followed by $\mathrm{V}_0^{=}$ and LP, a ranking typical for base-model approaches. Most improvement from increasing the ensemble size is obtained up to ensembles of size 10, a pattern that will reemerge in the overall analysis.

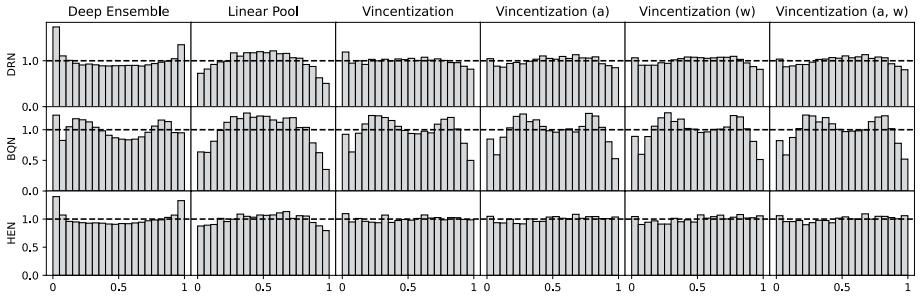

Fig. 5 Analysis of calibration, under- and overconfidence: PIT histograms of Bayesian DEs and the aggregation methods for the three NN variants and the Kin8nm data. The ensembles are of size 10

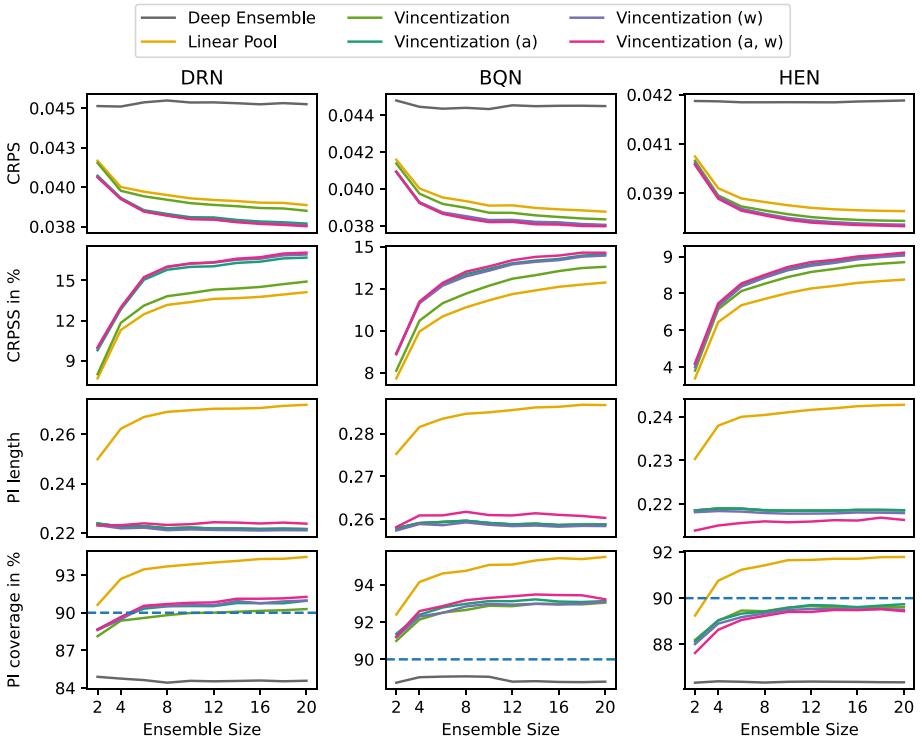

Fig. 6 Performance depending on ensemble size: Evaluation metrics of Bayesian DEs and the aggregation methods for the three NN variants and the Kin8nm data. Note the different scales on the vertical axis

Looking at the PI length, we find that while V_0^w and V_a^w have only a small influence on the PI length, the LP increases the PI length by a larger margin explaining the reduced confidence we observed in the PIT histograms. Note that $V_0^=$ and $V_a^=$ do not affect the PI length by definition and are therefore not included in the analysis of the PI lengths. Further, the PI length of the LP increases strongly for small ensemble sizes and remains almost constant for larger sizes. This increase in the PI length of the LP is resembled in the corresponding coverage, where the LP yields the PIs with the largest coverages, much larger than that of the individual forecasts. The VI forecasts of DRN and HEN are not only calibrated, but also their associated coverages are close to the nominal level. For BQN, all aggregation methods increase the coverages beyond the nominal level deviating even more. The increase in PI coverage by aggregation is also observed in most of other cases.

Up next, we analyze the effect of the ensemble size and the variability over the partitions based on Fig. 7. First, we note that in none of the cases aggregation degrades performance. Also, the boxes seem to stabilize and the variability becomes smaller with increasing ensemble size. Between the aggregation methods, we see that the variants with parameter estimation have a larger variability than LP and $V_0^=$. Although parameter estimation improves the predictive performance over all partitions, we see that in certain cases, especially small ensemble sizes, it results in the worst performance among the aggregation methods. Contrarily, the best results are also obtained by parameter estimation, i.e., we observe both posi-

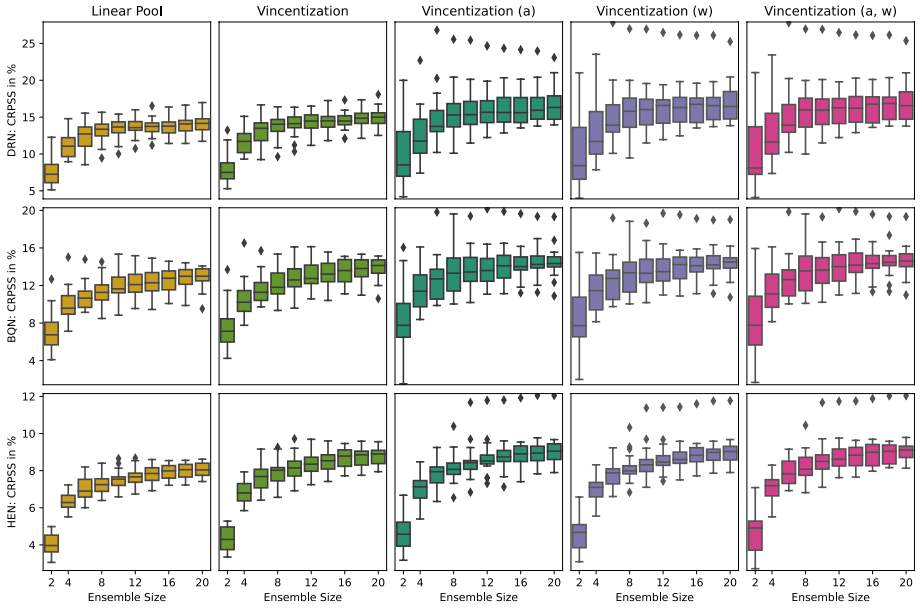

Fig. 7 Performance variability depending on ensemble size: Boxplots over the CRPSS values of the aggregation methods for each ensemble size for Bayesian DEs and the Kin8nm data

tive and negative outliers in terms of the performance. These result are also representative for the remaining cases.

At last, we quantify the computational costs required to aggregate the DE forecasts. Note that we focus on the computational costs from aggregation and not the generation of the DE forecasts themselves. For the analysis of the computational costs, we will differentiate between the time required for prediction only (with given coefficients) and time for estimating coefficients. Both are shown dependent on the ensemble size in Fig. 8. First, we find that there is a clear distinction between the LP and VI methods for prediction, but no notable difference within the VI methods themselves. The latter can be explained by the fact that the VI framework assumes equal weights, therefore, aggregation is always based on the sum of the individual forecasts followed by one multiplicative operation and, in case of an intercept, one additive operation. For the distinction between LP and VI, we find that averaging the model output is unsurprisingly the most efficient operation taking no longer than two milliseconds for a test set of size 819 (Fig. 18 in Appendix D displays the individual times via boxplots). For HEN, VI is computationally a bit more involved but still does not take longer than a second. The LP variants for DRN and BQN, which are based on mixture sampling, take the most time, with up to more than 100 s for BQN, where a customized sampling needed to be implemented. Although the ensemble size has an effect on the computational costs for prediction, it is negligible compared to differences between the methods. The time required for estimation is in contrast to prediction not dependent on the ensemble size. Again, the reason is that we only need to generate a sum of the DE predictions once before transforming with the coefficients. Between the methods, the two-parameter V_a^w-approach takes most time followed by $\mathrm{V}_a^=$ and V_0^w for all cases. Interestingly, the time required for optimum score estimation is consistent across the network variants. In absolute terms, esti-

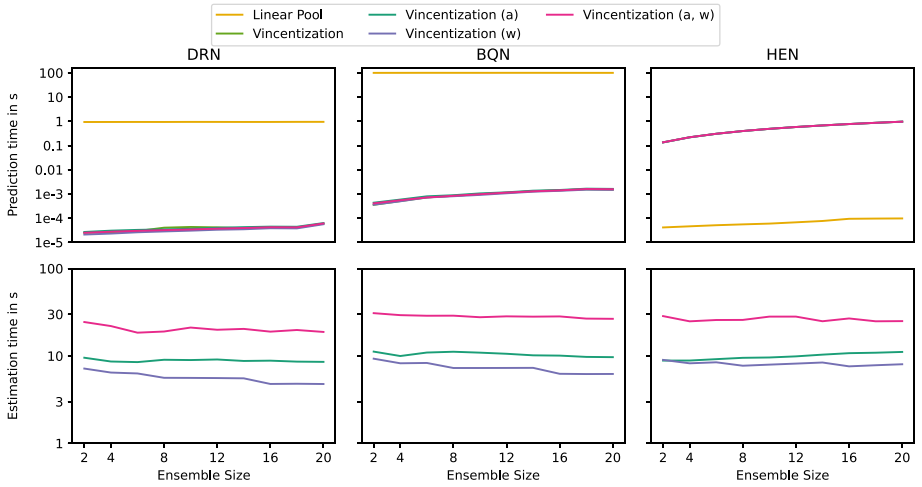

Fig. 8 Analysis of computational costs of aggregation for Kin8nm and Bayesian DEs: Average prediction and estimation times in seconds (for aggregation of the entire data set) dependent on the ensemble size. Note the logarithmic scale on the y-axis. The Kin8nm data set contains on average 819 test (prediction) and 1,475 validation samples (estimation)

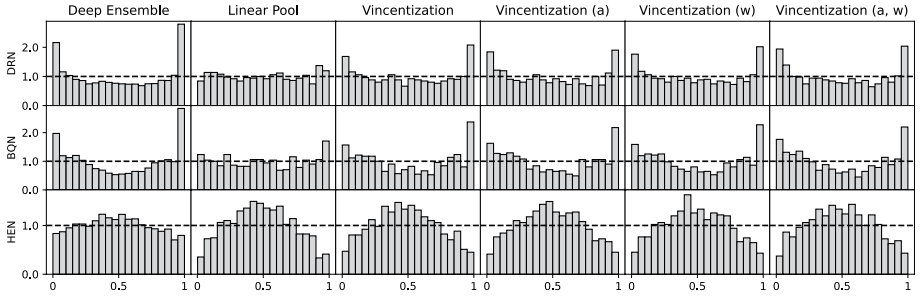

Fig. 9 Analysis of calibration, under- and overconfidence: PIT histograms of Bagging DEs and the aggregation methods for the three NN variants and the Boston data. The ensembles are of size 10

mation with V_a^w on a data set with on average 1,475 samples has a range from 10 to 60 s (see Fig. 18 in Appendix D) but mostly around 30 s. The one-parameter alternatives range from 5 to 15 s for $\mathrm{V}_a^=$ and 5 to 10 s for V_0^w.

4.3.2 Boston and Bagging Deep Ensembles

Now, we move on to the Boston data set and Bagging DEs. In contrast to the Kin8nm data set, the Boston data set is relatively small, including only a total of 506 samples. Another difference to the previous example is that the DEs are not generated using a base-model approach but instead a multi-model approach.

Again, we start by looking at the PIT histograms in Fig. 9. While DRN and BQN generate strongly overconfident forecasts, HEN results in underconfident forecasts. For Bagging, the methods generally tend to result in more overconfident forecasts. The LP reduces confi-

dence with respect to the DE, which results in calibrated forecasts for DRN and BQN. This case provides a good example of the strength of the LP for overconfident forecasts, which are often observed for NNs trained on small data sets, as in our case study. In contrast, the VI forecasts are not able to fully correct for the overconfidence but instead only slightly. For HEN, both the LP and VI forecasts are even less confident after aggregation. While performance metrics such as the CRPS improve by calibration, there is no guarantee that an aggregation method will improve the calibration of the forecasts, as now seen in both examples.

Turning to the evaluation measures in Fig. 10, we obtain results that differ from those of the Kin8nm data. The LP performs best for DRN and BQN, the two cases for which it generated calibrated forecasts. In case of the VI variants, V_a^w has the lowest CRPSS. Even though we observed the VI variants performing best for overconfident forecasts in the Kin8nm example, the LP performs especially well in these situations. Further, base-model approaches generally result in better performance using VI, the effect is not as strong for multi-model approaches such as Bagging. In contrast, the VI variants outperform the LP for HEN, where we do not observe over- but instead underconfidence. HEN favors aggregation by VI over the LP, also due to the fact that HEN more often results in underconfident forecasts. For the PI related measures, we obtain similar results as for the Kin8nm data, i.e., the LP increases the PI lengths drastically and all aggregation methods result in a larger PI coverage than that of the DE. Interestingly, this also holds for V_a^w, which decreases the PI length and results in sharper forecasts despite the observed underconfidence, which might

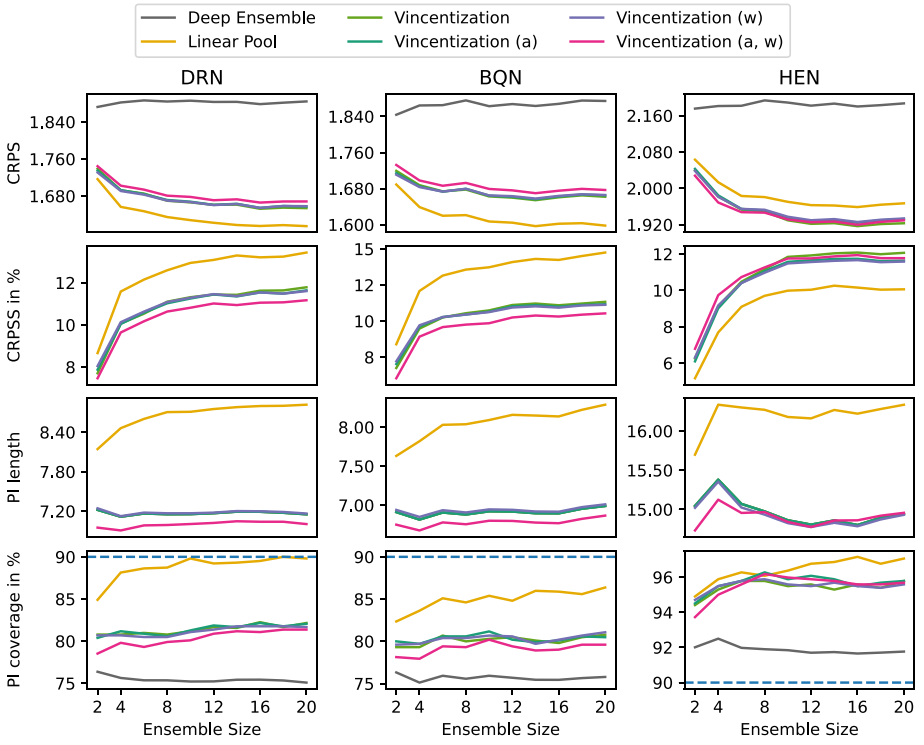

Fig. 10 Performance depending on ensemble size: Evaluation metrics of Bagging DEs and the aggregation methods for the three NN variants and the Boston data. Note the different scales on the vertical axis

be a result of overfitting, as the validation sets have an average size of 91 samples. Although the PI length becomes smaller, the PI coverage increases.

Since the results are similar to those observed for the Kin8nm data set, we do not show the effect of the ensemble size and the variability over the partitions analogously to Fig. 7. The analysis of computational costs for prediction and estimation with the aggregation methods comes to the same conclusions as for the Kin8nm data set, only the scale of the computation times changes with the data set size (see Fig. 19 in Appendix D). Overall, we conclude that the results of the in-depth analysis agree with the theoretical properties presented in Sect. 2. No aggregation method was superior throughout all cases, in some cases aggregation calibrated the forecasts, in some it did not and reduced confidence even further. Still, aggregation improved the predictive performance with respect to the DE throughout all cases.

4.4 Comprehensive Analysis of All Data Sets

Following the detailed analysis of selected examples, we apply the aggregation methods to forecasts of data sets. We start the evaluation with an overall analysis of the relative performance of the aggregation methods based on the CRPS. First, we compare only the two aggregation methods that do not require parameter estimation, namely, the LP and $\mathrm{V}_0^{=}$. Figure 11 shows the proportion of cases where one of the two methods is superior, meaning it has a lower CRPS, dependent on either the NN variant, the ensembling strategy or the data set. While the LP and $\mathrm{V}_0^{=}$ perform almost equally for DRN and BQN, $\mathrm{V}_0^{=}$ performs better than LP in three out of four cases for HEN. In case of the ensembling strategies, there

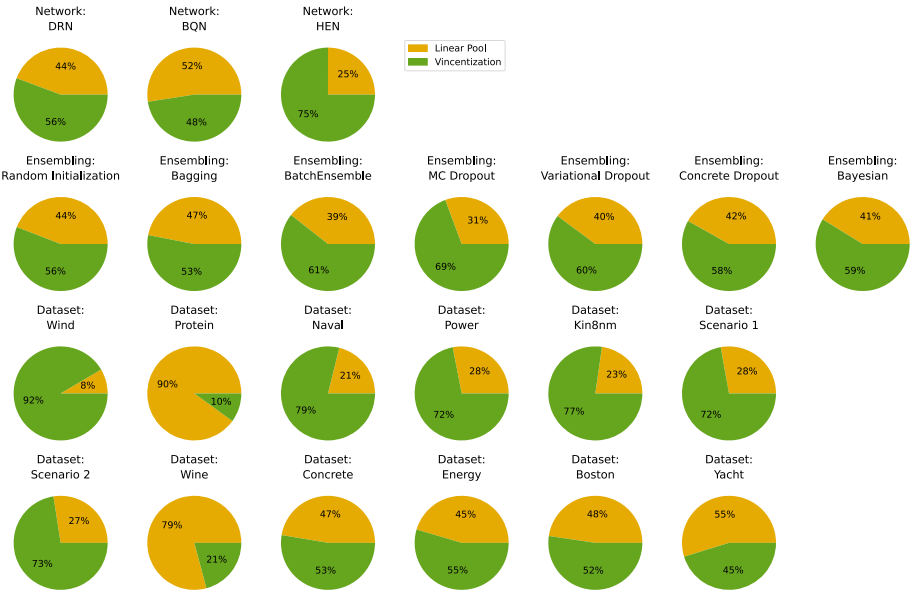

Fig. 11 Comparison of probability (LP) and quantile averaging ($\mathrm{V}_0^{=}$): Pie charts showing the proportion of cases in which either the LP is superior to $\mathrm{V}_0^{=}$ (yellow) or vice versa (green) in terms of the CRPS dependent on the NN variant, ensembling strategy or data set. The data sets are ordered according to their size starting with the largest

is a trend towards $V_0^=$ with larger proportions for ensembles generated with one base model such as MC dropout. Regarding the data sets, there are two sets for which the LP is the dominant aggregation method, namely, the Protein and Wine data sets. Besides these, $V_0^=$ is preferred among all data sets but the smallest. In terms of the size of the data sets, we find that the proportion of superior $V_0^=$ cases increases with the size.

If we include the other VI variants with parameter estimation in the comparison, we find that parameter estimation is able to improve upon $V_0^=$, and that the patterns in the differences between LP and $V_0^=$ persist and even become stronger. The conclusions drawn from the comparison of the LP and $V_0^=$ can be extended towards the other VI variants. In particular, Fig. 12 shows that the VI variants have the largest proportions of best performances for all cases but the Protein and Wine data set. The VI variants are especially dominant for HEN, the base-model strategies and the larger data sets up to Scenario 2, where the proportions of the VI methods increases with the data set size. Among the VI variants, V_a^w most often performs best followed by $V_a^=$ and V_0^w. This effect also becomes smaller as the sample size decreases. Further, parameter estimation is especially favorable in case of the base-model approaches such as the dropout variants and Bayesian NNs. Figure 20 in Appendix D shows not only the distribution of the best method but instead all ranks. Most interestingly, we find that the LP either performs best or worst most of the time. Hence, we find a clear distinction between LP and VI variants.

Before we investigate on the effect of the DE characteristics on the performance of the aggregation methods, we briefly analyze the PI lengths. The left panel in Fig. 13 shows the relative PI length difference of the aggregated forecasts with respect to associated DE. As expected, we find that the LP increases the PI length in almost all of the cases, while V_0^w and V_a^w are centered around zero. For HEN, V_a^w mostly decreases the PI length, which might be a reason that LP does not work as well as VI for this NN variant. As pointed out in the

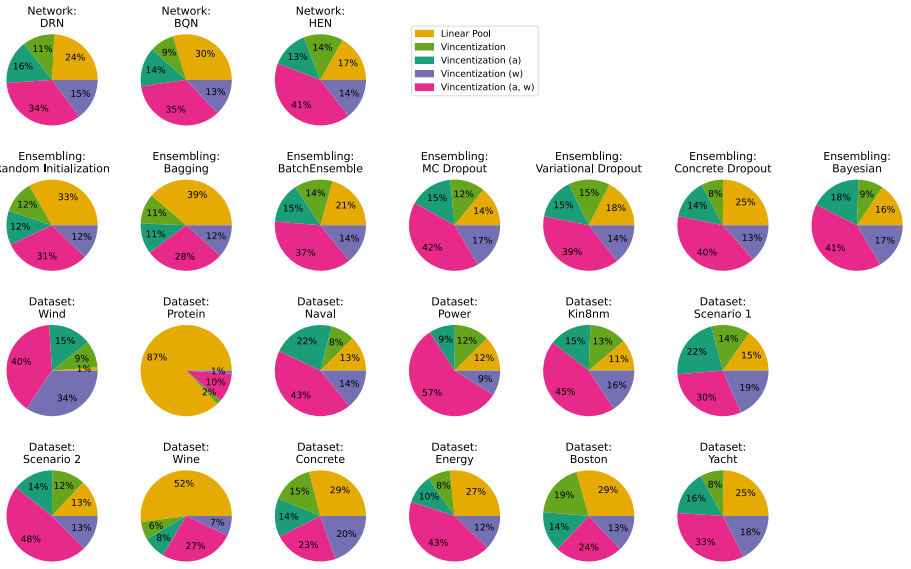

Fig. 12 Comparison of all aggregation methods: Pie charts showing the proportion of cases in which each of the aggregation methods is superior in terms of the CRPS dependent on the NN variant, ensembling strategy or data set. The data sets are ordered according to their size starting with the largest

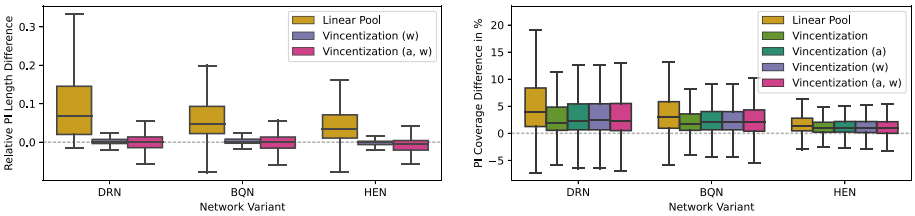

Fig. 13 Effect of aggregation on forecast confidence and calibration: Boxplots of the relative PI length differences (left) and the PI coverage differences (right) with respect to the DE dependent on the aggregation method and NN variant

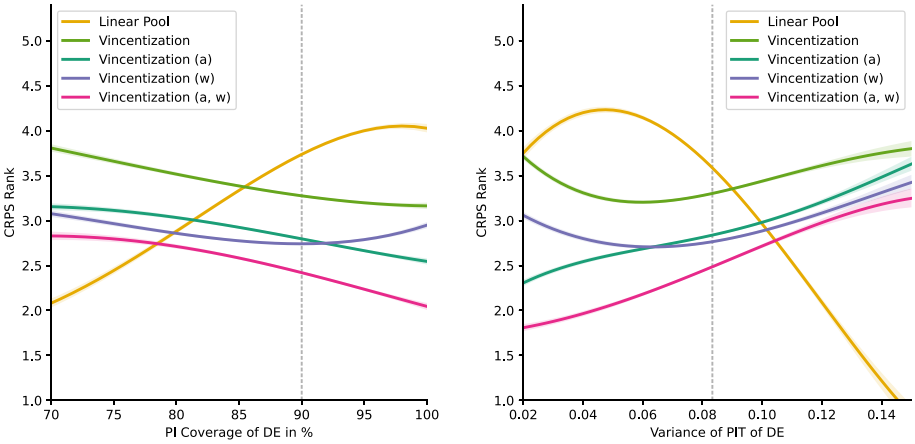

Fig. 14 Effect of ensemble characteristics on forecast aggregation: Polynomial regression curves of order 4 showing the relationship between the CRPS ranking of the aggregation methods and the PI coverage (left) respectively the dispersion (right) of the DE

in-depth analyses, the PI length is strongly connected to the PI coverage, which is illustrated analogously in the right panel of Fig. 13. Aligning with previous results, the PI coverage increases in a majority of the cases as all lower quartiles are positive. Figures 21 and 22 in Appendix D show similar plots dependent on the ensembling strategy and data set. Notably, V_a^w results in larger relative PI differences for ensembling strategies that rely on one base model, which might be a reason why V_a^w performs particularly well in these cases. For the data sets, the size has again an influence on the results as the differences become in general larger the smaller the data sets become.

Now, we investigate how the properties of the DE forecast affect the ranking of the methods. For this, we analyzed the dependence of the CRPS ranking of the aggregation methods depending on the PI coverage of the DE. The left panel of Fig. 14 shows a curve for each aggregation method based on a polynomial regression analysis, which was carried out on the evaluation data and models the relationship between the CRPS ranking and the PI coverage of the DE. The curves show that the LP performs better than the VI variants when the PI coverages are below the nominal level but becomes worse as the coverage increases. These findings agree with the in-depth analysis and supposed performance upsides of the LP based

on empirical evidence and its theoretical properties, as overconfident forecasts result in low PI coverages. For the VI variants, we observe the contrary effect.

However, we consider only one nominal level for the PI coverage in this study. Therefore, we additionally analyze the performance in terms of the dispersion, which we define as the variance of the PIT in Sect. 4.2. The right panel in Fig. 14 shows the results of an analogous regression analysis but based on the dispersion instead of the PI coverage. Recall that values below 0.0833 correspond to underconfidence and values above to overconfidence. The polynomial curves obtained by the regression analysis confirm the previous findings in that the LP works especially well for overconfident DE forecasts. Again, we observe the contrary effect for the VI variants. Notably, for calibrated DE forecasts, the analysis indicates that the VI variants result in a better performance.

Next to the dispersion of the DE forecasts, the diversity within the DE may also be a relevant factor for the performance of the methods. While we did not find a connection of the ensemble diversity to the ranking of the aggregation methods, we did for the connection to the CRPSS. Figure 15 shows the effect of the location diversity based on a regression analysis on the evaluation data. In general, the skill of the aggregation methods increases as the diversity increases. Still, there is large spread in the relationship between diversity and CRPSS as the wide distribution of the individual cases shows. Interestingly, in cases where the location diversity becomes larger than 0.4, we see an additional increase in skill. Hence, when the DE becomes more diverse, the improvement obtained by aggregation increases further. In these cases, the improvement by VI with parameter estimation is larger than that of the LP and $V_0^=$. At last, when the ensemble diversity becomes "too" large, the improvement vanishes. However, the conclusions drawn for cases with large diversity are based on a relatively small number of outliers. For the diversity in terms of the prediction uncertainty measured by the PI length and in terms of the performance measured by the CRPS (see Fig. 23 in Appendix D), we come to similar conclusions. As the performance diversity increases, the CRPSS of the aggregation methods increases, while the differences between the methods become larger. Recall that the CRPSS is calculated with respect to the mean score of the DE, hence the CRPSS is a relative and not an absolute performance measure.

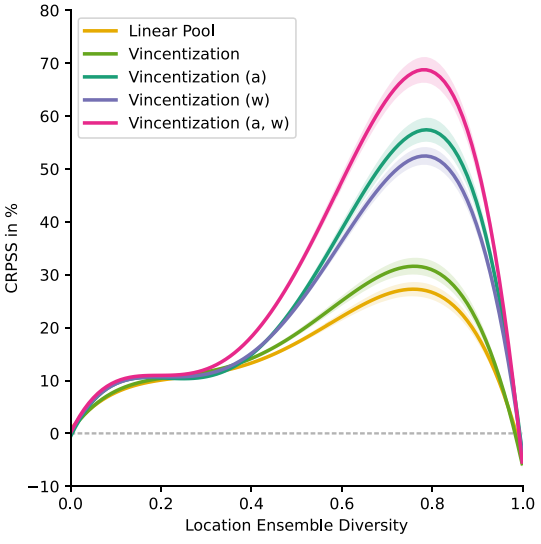

Fig. 15 Effect of ensemble characteristics on forecast aggregation: Polynomial regression curves of order 4 showing the relationship between the CRPSS of the aggregation methods and the location diversity of the DE

For the prediction uncertainty diversity, only a small effect is visible. Altogether, we conclude that location and performance diversity result in more improvement obtained from aggregation.

We end the overall evaluation by analyzing the effect of the ensemble size on the performance of the aggregation methods. Figure 16 shows the relative PI length differences of the aggregation methods and the DE dependent on the ensemble size. While the amplitudes of the relative PI length differences resemble those observed in Fig. 13, we find that the PI length of the LP forecasts is more dependent on the ensemble size than that of the other two variants. For ensembles of size 2 to 10, the PI length increase with the size of the ensemble. A similar effect was observed for both data sets in the in-depth analysis. For V_a^w, we find that the PI length also increases, but to a smaller extent. However, the spread of the relative PI length differences decreases slightly for the two VI variants that include parameter estimation. For LP, this is not the case. As in the in-depth analysis, most effects are observed up to ensembles of size 10.

The in-depth analysis showed that the improvement obtained by aggregation is saturated for ensembles of size 20. Here, we investigate whether this also holds for all benchmark data sets in general. For this, we compute the fraction of the potential improvement from the aggregation method that is already reached for the given ensemble size. The potential improvement is defined as the difference of the best CRPS value from all partitions and ensemble sizes of the corresponding setting with the CRPS value of the DE. For each ensemble size, we calculate the corresponding difference and set this in relation to the maximum improvement to compute the desired fraction. Figure 17 shows the fraction of the potential improvement dependent on the ensemble size. The plot reveals that an ensemble of size 2 improves upon the DE forecast by almost 50% with respect to the potential improvement for $\mathrm{V}_0^=$ and LP. For the VI variants with parameter estimation, the fraction is larger and almost 60% for V_a^w. A reason why the fraction is larger for the VI variants with parameter estimation is that the aggregated forecasts improve upon the DE additionally due to the corrections applied in the generalized VI framework. The improvement drastically increases by around 40% to ensembles of size 8, where we already have around 90% of the maximum possible improvement. Hence, by using an ensemble of size 8, one has already reached 90% of the potential improvement from aggregation. Afterwards, the improvement increases by around 1–2% for each step of 2 in the ensemble size. We did not observe any systematic differences in the dependence on the ensemble size for the NN variants and ensembling strategies.

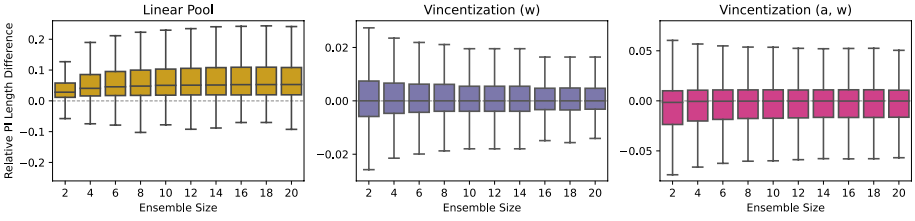

Fig. 16 Effect of aggregation on forecast confidence dependent on ensemble size: Boxplots of the relative PI length differences of the aggregation methods and the DE dependent on the ensemble size. Note the different scale of the y-axes

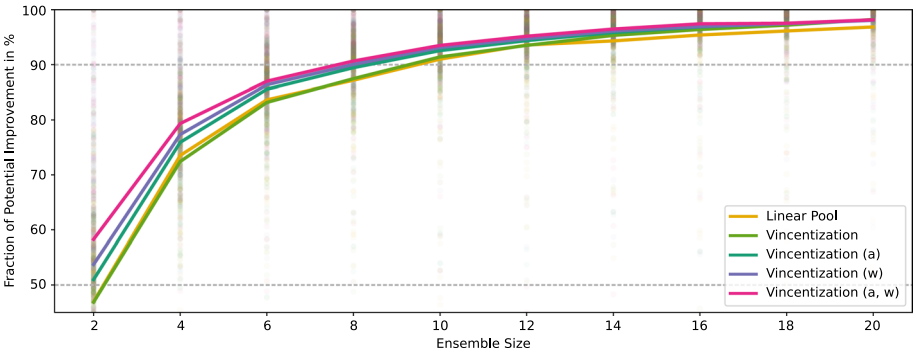

Fig. 17 Relationship between the fraction of the potential improvement of the aggregation methods and the ensemble size. The curve is based on the mean values of the individual fractions that are derived for each data set, ensembling strategy and NN variant in addition to the aggregation method and ensemble size

5 Discussion and Conclusions

We have conducted a systematic comparison of aggregation methods for the combination of distribution forecasts from ensembles of neural networks based on different ensembling strategies, so-called deep ensembles. In doing so, our work aims to reconcile and consolidate findings from the statistical literature on forecast combination and the machine learning literature on ensemble methods. Specifically, we propose a general Vincentization framework where quantile functions of the forecast distributions can be flexibly combined, and compare to the results of the widely used linear pool, where the probabilistic forecasts are linearly combined on the scale of probabilities. For deep ensembles of three variants of NN-based models for probabilistic forecasting that differ in the characterization of the output distribution, aggregation with both the LP and VI improves the predictive performance in a comprehensive evaluation on twelve data sets using seven ensembling strategies. The VI approaches frequently outperform the LP, but their ranking depends on the characteristics of the deep ensemble forecasts, especially whether the forecasts are under- or overconfident. For example, given ensemble members that are already calibrated or underconfident, the VI approaches are superior to the LP. While all approaches improve the predictive accuracy, the LP reduces the confidence of the forecasts resulting in (more) underconfident forecasts. If the individual forecast distributions are subject to systematic errors such as biases and dispersion errors, coefficient estimation via $\mathrm{V}_a^{=}$, V_0^w and V_a^w is able to correct these errors and improve the predictive performance considerably. While these combination approaches require the estimation of additional combination coefficients, the computational costs are negligible compared to the generation of the NN-based probabilistic forecasts and can be performed on the validation data without restricting the estimation of the NNs. However, the smaller the validation data set, the larger the variability in the actual improvement from aggregation. In terms of the ensembling strategies, we found that VI performs better than the LP for deep ensembles generated with one base-model, e.g., dropout or Bayesian NNs. In particular, VI with parameter estimation performs especially well for such deep ensembles.

Even though forecast combination generally improves the predictive performance, a lack of calibration of severely misspecified individual forecast distributions cannot be corrected

by the aggregation methods considered here. In the context of NNs and deep ensembles, the calibration of (ensemble) predictions and re-calibration procedures have been a focus of much recent research interest (Guo et al., 2017; Ovadia et al., 2019). For example, in line with the results of Gneiting and Ranjan (2013), deep ensemble predictions based on the LP were found to be miscalibrated and should be re-calibrated after the aggregation step (Rahaman & Thiery, 2020; Wu & Gales, 2021). A wide range of re-calibration methods, which simultaneously aggregate and calibrate the ensemble predictions (such as the $\mathrm{V}_a^{=}$, V_0^w and V_a^w approaches presented in Sect. 2.3 for VI), have been proposed in order to correct the systematic errors introduced by the LP in the context of probability forecasting for binary events (Allard et al., 2012). For example, the beta-transformed LP composites the CDF of a Beta distribution with the LP (Ranjan & Gneiting, 2010), and Satopää et al. (2014) propose to aggregate probabilities on a log-odds scale. Some of these approaches can in principle be extended to the case of forecast distributions considered here (Gneiting & Ranjan, 2013). However, the application of such recalibration approaches to LP-aggregated forecast distributions is notably more complex and computationally more expensive than the VI methods. Comparing the performance of such post-hoc corrections of forecasts combined via the LP to the VI approaches with learnable parameters considered here constitutes an interesting starting point for future research. For VI, more sophisticated approaches that allow the weights to depend on the quantile levels might improve the predictive performance (Fakoor et al., 2023). Further, moving from a linear combination function towards more complex transformations allowing for non-linearity might help to correct more involved calibration errors.

The focus on our study was on the effects of aggregating distribution forecasts from a given deep ensemble, and not on finding the overall best ensembling strategy to produce neural network-based probabilistic forecasts in the form of a deep ensemble. While we did not systematically compare the performance of the ensembling strategies, the naive ensemble generally seemed to perform best in terms of the CRPS. In particular, the naive ensemble seemed to be the most stable approach for generating a deep ensemble. That said, the other ensembling strategies have distinct advantages and have proven their effectiveness in other applications. Most importantly, aggregating forecasts did not have an effect on the ranking of the ensembling strategies.

When deciding how to generate a deep ensemble, the trade-off between computational costs and predictive performance plays an important role. Larger ensembles yield a better predictive performance but at the expense of increased computing time. In case of base-model approaches, the additional computational costs are in general significantly lower compared to the multi-model approaches, where multiple models need to be trained. However, the uncertainty in the training of the base model is not taken account for. To enhance predictive performance, one could follow both approaches by generating multiple base-models, which are then each used to generate their own sub-ensemble that need to aggregated altogether. While we have assumed equally weighted aggregation schemes, more sophisticated approaches that take the interplay of two ensemble-generating mechanisms into account might enhance the predictive performance.

To limit the scope of our investigations, we restricted our attention to ensembles of neural network-based forecasting methods. However, the forecast aggregation strategies could also be applied to other ensemble methods in machine learning such as variants of random forests for probabilistic forecasting, including quantile regression forests (Meinshausen,

2006), distributional regression forests (Schlosser et al., 2019) or distributional random forests (Cevid et al., 2022). A detailed investigation of the effects of forecast combination approaches for those ensemble methods, and comparisons to the deep ensembles considered here, constitutes an interesting starting point for future work. Further, we restricted our attention to univariate probabilistic forecasts. Considering multivariate distributional regression approaches such as distributional random forests (Cevid et al., 2022) or generative machine learning methods (Chen et al., 2024) is another interesting direction for future research. However, the lack of universally agreed-upon notions of multivariate quantiles and multivariate forecast distributions being often available in the format of simulated samples only, pose notable challenges to extension of the LP and VI approaches considered here.

Finally, we summarize five key recommendations for aggregating distribution forecasts from deep ensembles based on our results. First, in order to optimize the final predictive performance of the aggregated forecast, the individual component forecasts should be optimized as much as possible.[11] While forecast combination improves predictive performance, it generally did not effect the ranking of the different NN-variants for generating probabilistic forecasts, and can be unable to fix substantial systematic errors. Second, generating an ensemble with a size of 10 appears to be a sensible choice, with only minor improvements being observed for up to 20 members. This corresponds to the results in Fort et al. (2019) and ensemble sizes typically chosen in the literature (Lakshminarayanan et al., 2017; Rasp & Lerch, 2018), but the benefits of generating more ensemble members need to be balanced against the computational costs, and sometimes smaller ensembles have been suggested (Ovadia et al., 2019; Abe et al., 2022). Third, the choice of aggregation methods should take the dispersion of the individual ensemble member forecasts into account, whether they are too confident or not confident enough. For calibrated or underconfident forecasts, VI is favorable, for overconfident forecasts, the LP may be the better option. Fourth, parameter estimation via V_a^w frequently enhances predictive performance, especially for larger data sets or base-model ensembling strategies. The choice of the specific variant within the general VI framework depends on potential misspecifications of the individual component distributions. The fifth and last recommendation is of practical nature and concerns the standard practice of averaging model output. Our advice is to match forecast distribution and aggregation method such that averaging the model output results in the desired aggregation characteristics. Such a simple aggregation process is favorable especially for ensembling strategies that can generate large ensembles at low cost. Note that these conclusions, in particular the superiority of the quantile aggregation approaches, refer to the specific situation of deep ensembles considered here. The property of shape-preservation justifies the use of VI from a theoretical perspective in a setting where the ensemble members are based on the same model and data. If the ensemble members differ in terms of the model used to generate the forecast distribution or the input data they are based on, shape-preservation might not be desired. Instead, a model selection approach based on the LP, which allows for obtaining a multi-modal forecast distribution, might better represent the possible scenarios that may materialize.

[11] Abe et al. (2022) find that deep ensembles do not offer benefits compared to single larger (that is, more complex) NNs. Our results do not contradict their findings since we address a conceptually different question and argue that given the generation of a deep ensemble, the individual members' forecasts should be optimized as much as possible. In this situation, a single NN will generally not be able to match the predictive performance of the associated deep ensemble.

Appendix A: Network Setup

Here, we describe the setup of the NNs in more detail. First, note that the predictor variables are standardized based on the respective training data (i.e., separately for each partition) in a preprocessing step. Other than that, all NNs are trained over 500 epochs using early stopping with a patience of 10. The NNs are implemented in Python (3.10.6; Python Software Foundation 2020) via keras (2.10.0; Chollet and Others 2015) built on tensorflow (2.10.0; Abadi et al. 2015).

In case of the BatchEnsemble, we generate one ensemble of the maximum ensemble size, i.e., 20, and then use the first n members for aggregation of an ensemble of size n for each combination of NN variant, data set and partition. Further, the chosen batch sizes (see Table 4) refer to the effective batch sizes per ensemble member within the parallel training. If the required batch size for parallel training exceeds the size of the training set, we resample the missing data points. In case of the dropout variants and BNN, where the ensemble is generated on one base model, the direct NN output of one prediction corresponds to one ensemble member. For the dropout variants, we drop only the neurons in the hidden layers and not the input layer. In case of MC dropout, the chosen architectures (see Table 4) refer to the effective architecture, as we additionally scale up the number of neurons based on the dropout rate.

Regarding the NN variants, the BQN models use 99 equidistant quantile levels from 0.01 to 0.99 in the loss function. For HEN, the target variable is also standardized on the training data (analogous to the predictor variables). As described in Sect. 3.1.3, the bin edges are defined by quantiles of a standard normal distribution. Based on experiments on the validation set and previous applications, we use equidistant quantile levels within the interval $[0.05, 0.95]$ and a finer resolution (with respect to the quantile level) in the tails of the distribution. For the tails, we chose the 10 (fixed) bin edges $b_0 = \Phi^{-1}\left(10^{-16}\right)$, $b_1 = \Phi^{-1}\left(10^{-8}\right)$, $b_2 = \Phi^{-1}(0.0001)$, $b_3 = \Phi^{-1}(0.01)$, $b_4 = \Phi^{-1}(0.05)$, where Φ denotes the CDF of the standard normal distribution, and $b_{N+1-\ell} = 1 - b_\ell$ for $\ell = 0, \ldots, 4$. If the minimum (maximum) within the training data is smaller (larger) than the bin edge b_0 (b_{N+1}), we adapt the bin edge to this value minus (plus) a small threshold. The other $N - 9$ bin edges are then chosen as the quantiles at equidistant levels between 0.05 and 0.95.

Appendix B: Hyperparameter Tuning

For the hyperparameter tuning, we first note that we did not perform a separate hyperparameter tuning for bagging and BatchEnsemble, but instead use the hyperparameters obtained for the naive ensemble runs. This was done, as we tune the performance of an individual ensemble member, which are structurally identical for these three variants. As described in Sect. 4, we choose the hyperparameters that perform best on the first two random partitions. Performance was measured based on the mean CRPS and sanity checked with PIT histograms and the logarithmic score (negative log-likelihood). Unless a severe degree of mis-

calibration or strong deviations in the logarithmic score were detected (with respect to the competing hyperparameter sets), the hyperparameters with the lowest CRPS were chosen. The following variables and values were considered for hyperparameter tuning:

- Batch size (BA): 16, 32, 64, 256.
- Activation function (Actv): Relu, Softplus.
- Architectures (Arch): [64, 32], [512, 256], [64, 64, 32], [512, 512, 256], [512, 512, 256, 128]. In Table 4, we denote the architecture by the number of layers and the number of nodes in the first layer, e.g., 2–512 for [512, 256].
- Learning rate (LR): 0.001, 0.0005.
- Dropout rate (DR; for MC dropout): 5%, 10%, 20%, 50%, 80%.
- Prior (PR; for Bayesian NN): Uniform, standard normal, Laplace.
- Degree of Bernstein polynomials d (for BQN): 8, 12.
- Number of bins N (for HEN): 20, 30.

Tables 4 lists the chosen hyperparameter configurations. Recall that all NNs are trained over 500 epochs using early stopping with a patience of 10.

Appendix C: Description of the simulation studies

Scenarios 1 and 2 in Sect. 4.4 correspond to models 1 and 4 proposed in Li et al. (2021). The results for their other models do not provide additional insights and are thus not included here. Note that we reduce the size of the test set from 10,000 to 1,000 for computational reasons.

The first simulation scenario we consider is a linear model with normally distributed errors. Based on a random vector of predictors $\boldsymbol{X} \in \mathbb{R}^5$, which serves as the input of the NNs, and the random coefficient vectors $\boldsymbol{\beta_1}, \boldsymbol{\beta_2} \in \mathbb{R}^5$, which are fixed for each run of the simulation and unknown to the forecaster, the target variable Y is calculated via

$$Y = \boldsymbol{X}^T \boldsymbol{\beta_1} + \epsilon \cdot \exp\left(\boldsymbol{X}^T \boldsymbol{\beta_2}\right),$$

where $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I_5})$, $\boldsymbol{\beta_1} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I_5})$, $\boldsymbol{\beta_2} \sim \mathcal{N}\left(\boldsymbol{0}, 0.45^2 \boldsymbol{I_5}\right)$ and $\epsilon \sim \mathcal{N}(0, 1)$. In the second scenario, we consider a skewed distribution with a nonlinear mean function. The target variable Y is defined by

$$Y = 10 \sin(2\pi X_1 X_2) + 20 (X_3 - 0.5)^2 + 10 X_4 + 5 X_5 + \epsilon,$$

where $\boldsymbol{X} = (X_1, \ldots, X_5)^T$, $X_1, \ldots, X_5 \overset{iid}{\sim} \mathcal{U}(0, 1)$, and $\epsilon \sim \text{SkewNormal}(0, 1, -5)$.

Table 4 Hyperparameter choices for the case studies in Sect. 4

	DRN					BQN						HEN					
	Arch	Actv	BA	LR	DR/PR	*d*	Arch	Actv	BA	LR	DR/PR	*N*	Arch	Actv	BA	LR	DR/PR
`Naive ensemble/bagging/BatchEnsemble`																	
Gusts	2–512	Soft	32	.0010	–	12	3–512	Soft	64	.0010	–	20	4–512	Soft	64	.0005	–
Scenario 1	2–512	Soft	16	.0005	–	8	4–512	Soft	64	.0010	–	30	4–512	Soft	16	.0005	–
Scenario 2	2–512	Soft	256	.0005	–	8	3–512	Soft	16	.0005	–	30	4–512	Soft	64	.0005	–
Protein	4–512	Relu	16	.0005	–	8	4–512	Relu	32	.0010	–	30	4–512	Relu	32	.0010	–
Naval	4–512	Relu	16	.0010	–	8	2–512	Relu	16	.0005	–	30	3–64	Relu	16	.0005	–
Power	3–64	Relu	16	.0005	–	12	3–64	Relu	256	.0005	–	30	4–512	Relu	16	.0005	–
Kin8nm	4–512	Soft	16	.0005	–	8	3–64	Soft	16	.0005	–	20	3–64	Soft	16	.0010	–
Wine	2–512	Relu	16	.0010	–	12	3–512	Relu	16	.0010	–	30	3–64	Relu	16	.0005	–
Concrete	3–512	Relu	16	.0010	–	8	3–512	Relu	16	.0005	–	20	4–512	Relu	16	.0005	–
Energy	2–512	Relu	16	.0010	–	12	2–512	Relu	16	.0005	–	30	4–512	Soft	16	.0010	–
Boston	3–512	Relu	16	.0010	–	8	4–512	Relu	16	.0005	–	30	4–512	Relu	16	.0010	–
Yacht	2–64	Soft	16	.0010	–	12	3–64	Soft	16	.0005	–	30	3–512	Soft	16	.0010	–
`MC dropout`																	
Gusts	2–512	Soft	32	.0005	20%	8	3–64	Soft	16	.0005	10%	20	2–64	Soft	32	.0005	10%
Scenario 1	2–512	Soft	16	.0010	5%	8	3–512	Soft	16	.0010	5%	30	3–512	Soft	16	.0010	5%
Scenario 2	2–512	Relu	16	.0010	5%	12	3–512	Relu	16	.0005	5%	30	4–512	Relu	32	.0005	5%
Protein	4–512	Relu	32	.0005	20%	8	4–512	Relu	64	.0010	10%	30	4–512	Relu	64	.0005	5%
Naval	2–64	Relu	16	.0005	5%	8	4–512	Relu	16	.0010	10%	30	3–512	Relu	16	.0005	5%
Power	2–512	Soft	16	.0010	5%	8	2–512	Soft	16	.0005	5%	30	3–512	Relu	32	.0010	5%
Kin8nm	3–512	Soft	32	.0005	5%	8	4–512	Soft	16	.0005	5%	30	4–512	Soft	16	.0005	5%
Wine	2–512	Relu	32	.0010	5%	8	2–512	Relu	16	.0005	5%	30	3–64	Relu	64	.0010	5%
Concrete	4–512	Relu	16	.0005	5%	12	3–512	Relu	16	.0010	5%	20	4–512	Relu	16	.0010	5%
Energy	2–512	Relu	16	.0010	5%	12	3–512	Relu	16	.0005	5%	30	4–512	Soft	32	.0010	10%
Boston	4–512	Relu	16	.0005	10%	12	3–512	Relu	32	.0010	5%	30	4–512	Relu	16	.0010	5%
Yacht	2–512	Relu	16	.0005	5%	12	2–512	Relu	16	.0010	5%	30	3–512	Soft	16	.0005	5%

Table 4 (continued)

	DRN					BQN						HEN					
	Arch	Actv	BA	LR	DR/PR	*d*	Arch	Actv	BA	LR	DR/PR	*N*	Arch	Actv	BA	LR	DR/PR
`Variational dropout`																	
Gusts	2–64	Soft	16	.0010	–	12	2–64	Soft	16	.0005	–	20	3–64	Soft	16	.0010	–
Scenario 1	2–64	Soft	16	.0010	–	8	2–64	Soft	16	.0005	–	30	2–64	Soft	16	.0010	–
Scenario 2	3–64	Relu	16	.0010	–	12	2–64	Relu	16	.0010	–	30	3–512	Soft	16	.0005	–
Protein	2–64	Relu	16	.0010	–	12	3–512	Relu	32	.0005	–	30	2–512	Relu	16	.0005	–
Naval	2–64	Soft	64	.0005	–	12	3–512	Relu	16	.0010	–	20	3–512	Soft	16	.0005	–
Power	2–64	Relu	32	.0005	–	8	3–64	Soft	16	.0010	–	30	4–512	Soft	16	.0005	–
Kin8nm	2–64	Relu	16	.0010	–	12	2–64	Soft	16	.0005	–	30	3–64	Soft	16	.0010	–
Wine	2–64	Relu	16	.0010	–	8	2–64	Soft	16	.0005	–	30	2–64	Relu	16	.0005	–
Concrete	2–64	Soft	16	.0010	–	12	2–64	Relu	16	.0005	–	30	2–512	Relu	256	.0005	–
Energy	2–64	Relu	16	.0005	–	8	2–64	Relu	16	.0010	–	30	2–64	Relu	32	.0005	–
Boston	2–64	Soft	16	.0005	–	12	2–64	Relu	16	.0010	–	30	2–64	Relu	256	.0010	–
Yacht	2–64	Relu	16	.0010	–	12	2–512	Relu	32	.0005	–	30	2–64	Relu	16	.0010	–
`Concrete dropout`																	
Gusts	3–64	Soft	256	.0005	–	8	3–512	Soft	32	.0005	–	20	3–64	Soft	16	.0005	–
Scenario 1	3–64	Soft	16	.0010	–	8	3–64	Soft	16	.0010	–	30	3–64	Soft	32	.0010	–
Scenario 2	3–64	Soft	16	.0010	–	8	3–64	Soft	16	.0010	–	30	3–64	Relu	16	.0005	–
Protein	4–512	Relu	32	.0005	–	12	4–512	Relu	256	.0010	–	30	4–512	Relu	64	.0005	–
Naval	2–512	Relu	256	.0005	–	8	2–512	Relu	64	.0005	–	30	3–64	Relu	16	.0005	–
Power	3–512	Relu	16	.0005	–	12	3–512	Relu	64	.0010	–	30	4–512	Relu	16	.0010	–
Kin8nm	3–512	Relu	64	.0005	–	12	3–512	Relu	32	.0005	–	30	3–64	Soft	16	.0010	–
Wine	3–512	Relu	32	.0005	–	8	3–512	Relu	32	.0005	–	30	4–512	Relu	32	.0005	–
Concrete	4–512	Relu	16	.0010	–	8	3–512	Relu	32	.0010	–	20	4–512	Relu	16	.0005	–
Energy	3–512	Relu	64	.0010	–	8	2–512	Relu	16	.0010	–	30	3–512	Relu	32	.0010	–
Boston	3–512	Relu	16	.0010	–	12	3–512	Relu	64	.0010	–	20	4–512	Relu	16	.0005	–
Yacht	4–512	Relu	16	.0010	–	12	3–512	Relu	16	.0010	–	20	3–512	Soft	32	.0010	–

Table 4 (continued)

	DRN					BQN						HEN					
	Arch	Actv	BA	LR	DR/PR	*d*	Arch	Actv	BA	LR	DR/PR	*N*	Arch	Actv	BA	LR	DR/PR
`Bayesian`																	
Gusts	3–64	Soft	32	.0005	Lapl	12	3–512	Soft	64	.0010	Norm	30	4–512	Soft	64	.0010	Lapl
Scenario 1	2–64	Soft	32	.0010	Unif	12	3–512	Soft	64	.0010	Norm	30	3–512	Soft	64	.0010	Norm
Scenario 2	2–512	Soft	64	.0005	Unif	12	2–512	Soft	16	.0005	Unif	30	4–512	Soft	64	.0005	Norm
Protein	3–512	Relu	16	.0010	Norm	8	4–512	Relu	32	.0005	Norm	30	2–512	Relu	32	.0005	Norm
Naval	3–512	Soft	32	.0010	Unif	8	3–512	Soft	32	.0010	Unif	30	4–512	Relu	16	.0005	Norm
Power	2–64	Relu	16	.0005	Unif	8	3–64	Relu	256	.0005	Lapl	30	4–512	Relu	64	.0010	Norm
Kin8nm	4–512	Relu	64	.0005	Norm	8	3–64	Soft	64	.0010	Unif	30	3–512	Relu	32	.0005	Lapl
Wine	2–64	Relu	16	.0010	Norm	8	3–64	Relu	16	.0010	Norm	30	3–64	Relu	16	.0010	Lapl
Concrete	3–512	Relu	64	.0005	Unif	12	3–512	Relu	16	.0010	Norm	20	4–512	Relu	16	.0010	Norm
Energy	3–512	Relu	16	.0010	Norm	8	3–512	Relu	16	.0005	Unif	30	3–512	Soft	64	.0005	Unif
Boston	3–512	Relu	16	.0010	Norm	12	3–512	Relu	32	.0010	Norm	30	2–512	Soft	256	.0010	Unif
Yacht	4–512	Relu	16	.0005	Unif	8	2–512	Relu	64	.0010	Unif	30	4–512	Relu	16	.0010	Norm

For the notation, see Appendix B

Appendix D: Additional Figures

See Figs. 18, 19, 20, 21, 22, 23.

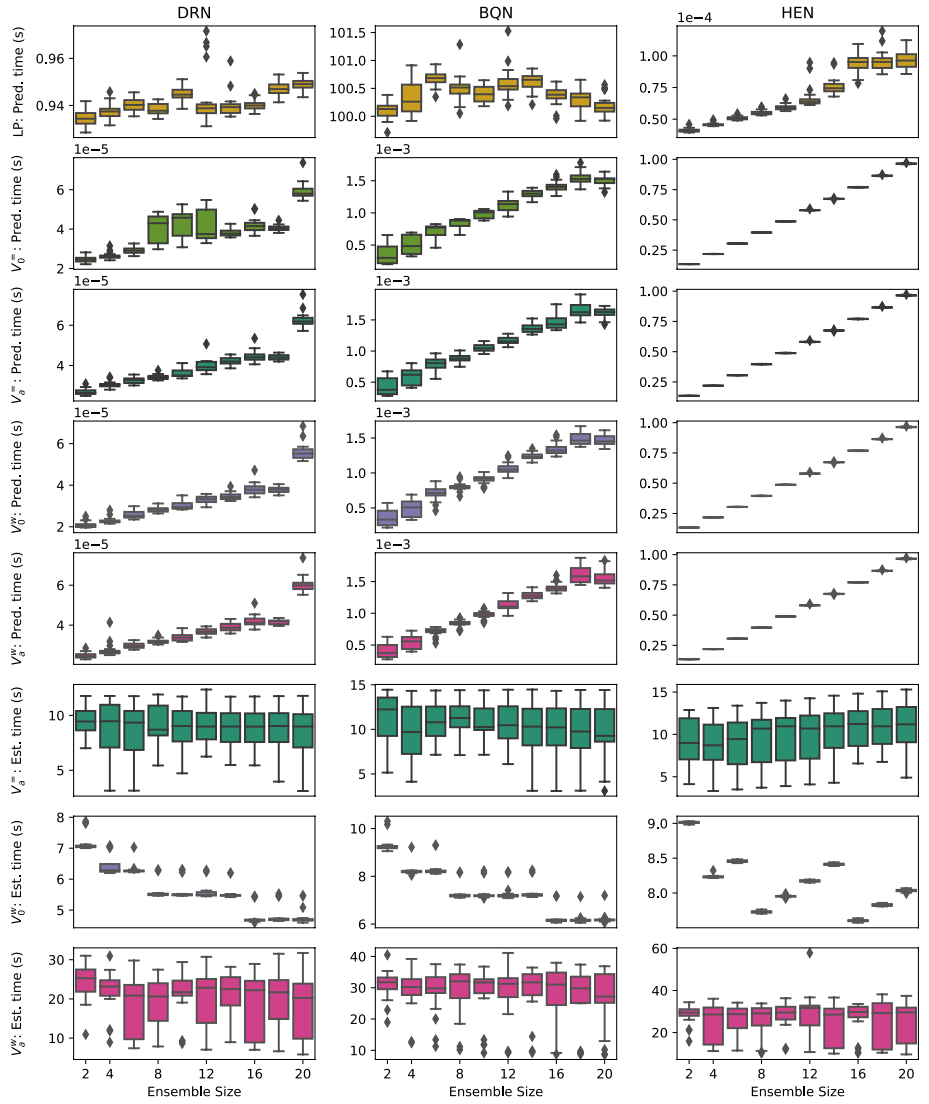

Fig. 18 Analysis of computational costs of aggregation for Kin8nm and Bayesian DEs: Boxplots of the average prediction and estimation times in seconds (for aggregation of the entire data set) dependent on the ensemble size. The Kin8nm data set contains on average 819 test (prediction) and 1,475 validation samples (estimation)

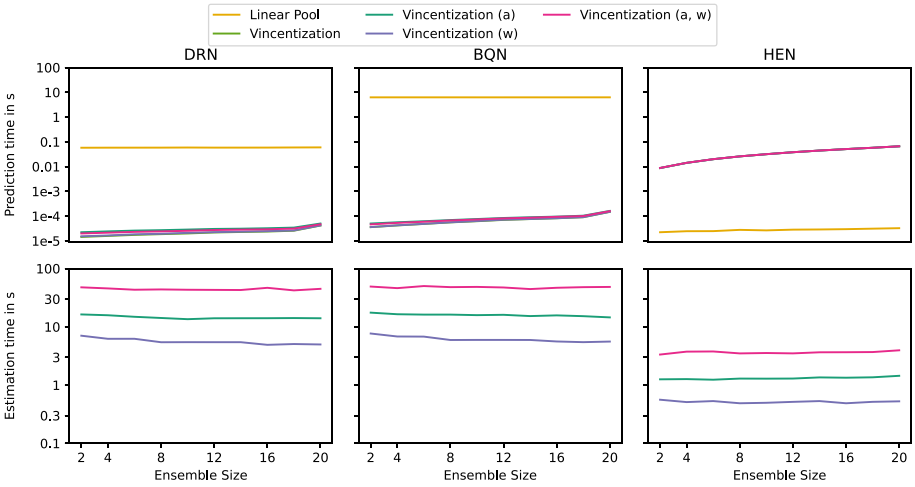

Fig. 19 Analysis of computational costs of aggregation for Boston and Bagging DEs: Average prediction and estimation times in seconds (for aggregation of the entire data set) dependent on the ensemble size. Note the logarithmic scale on the *y*-axis. The Boston data set contains on average 51 test (prediction) and 91 validation samples (estimation)

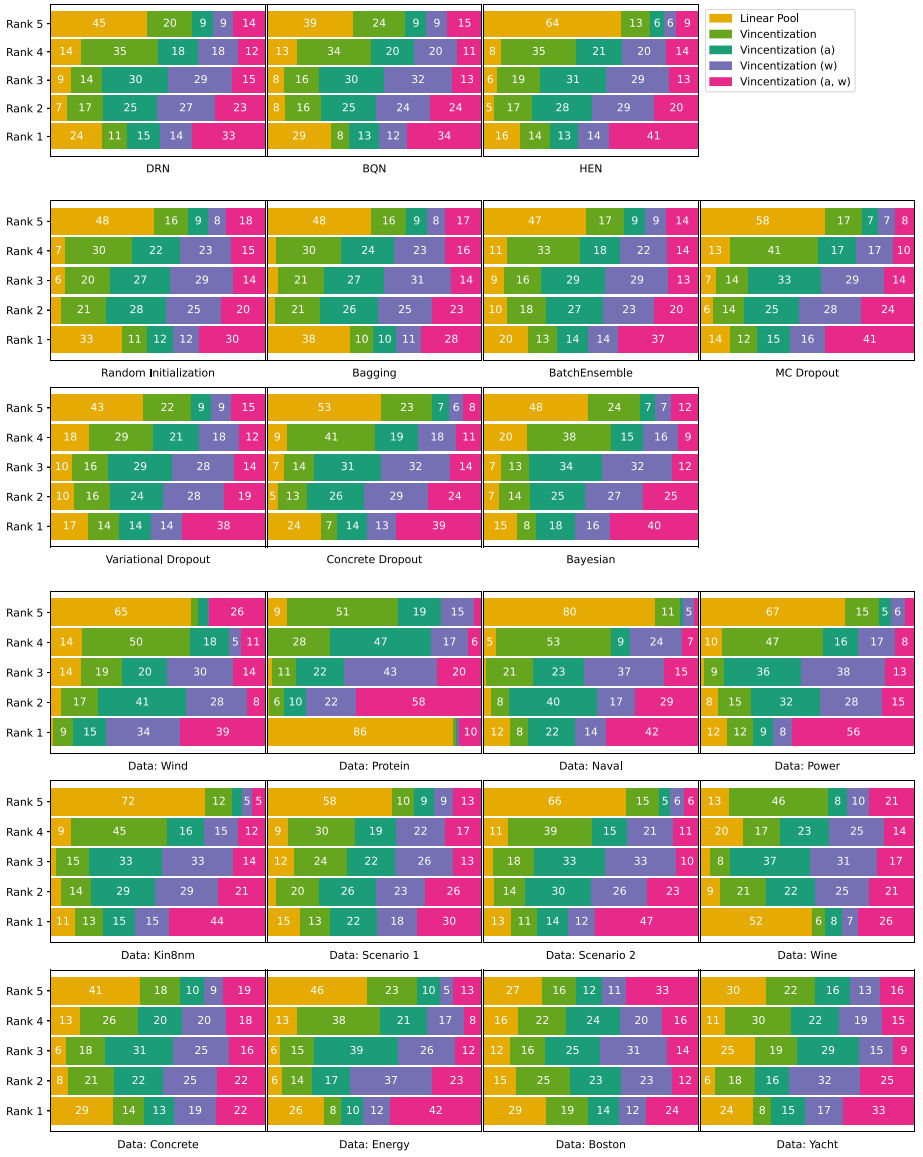

Fig. 20 Comparison of aggregation methods: Stacked bar plots showing the relative distribution of the CRPS ranking dependent on the NN variant, ensembling strategy and data set. Percentages below 5% are not labeled. The data sets are ordered according to their size starting with the largest

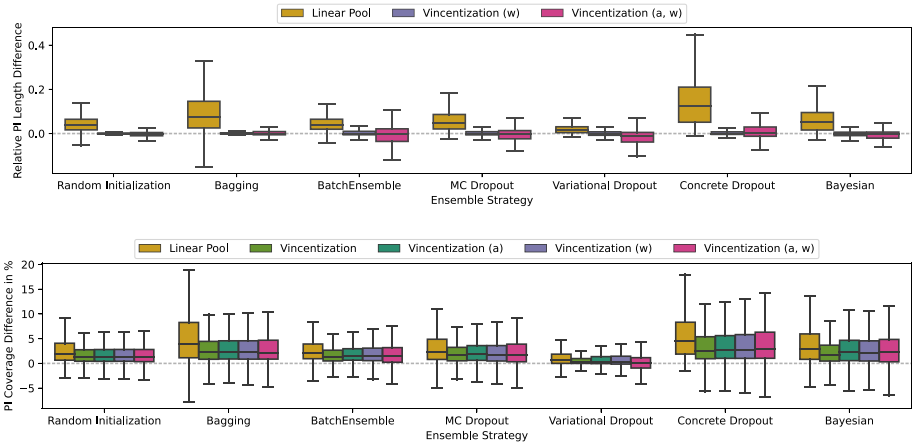

Fig. 21 Forecast sharpness and calibration per ensembling strategy: Boxplots of the relative PI length differences (left) and the PI coverage differences (right) with respect to the DE dependent on the aggregation method and ensembling strategy

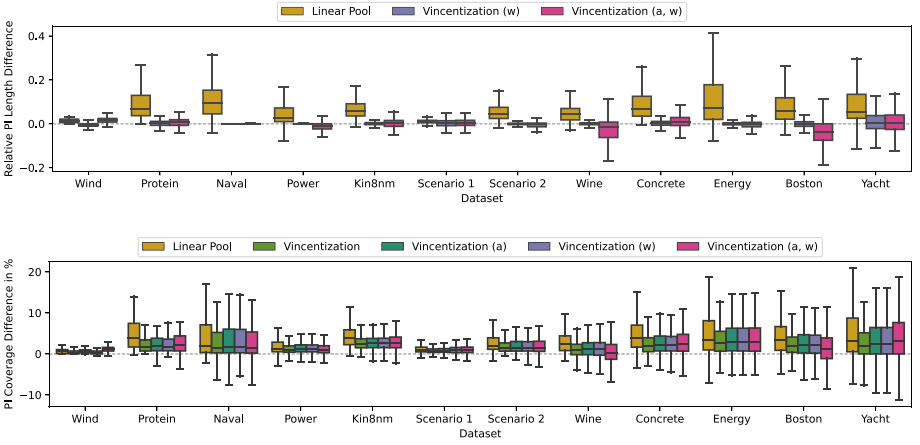

Fig. 22 Forecast sharpness and calibration per data set: Boxplots of the relative PI length differences (left) and the PI coverage differences (right) with respect to the DE dependent on the aggregation method and data set

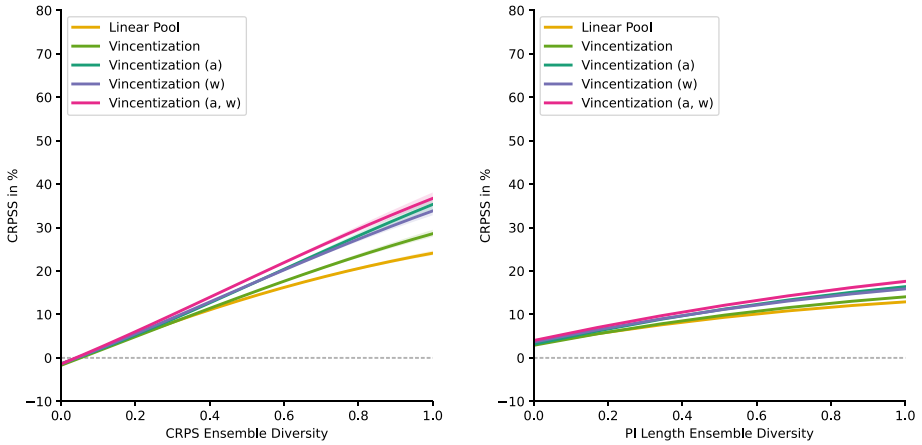

Fig. 23 Effect of ensemble diversity on improvement by aggregation: Polynomial regression curves of order 4 showing the relationship between the CRPSS and the prediction performance (left) resp. uncertainty (right) diversity of the aggregation methods

Acknowledgements We thank Daniel Wolffram, Eva-Maria Walz, Nina Horat, Anja Mühlemann, Alexander Jordan and Tilmann Gneiting for helpful comments and discussions. Further, we thank three anonymous reviewers, whose constructive comments helped to improve an earlier version of this paper.

Author Contributions B.S. and S.L. contributed to the study conception and design. Data collection, programming and analysis were performed by B.S. and L. K.. The first draft of the manuscript was written by B.S., S.L. commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. The research leading to these results has been done within the project C5 "Dynamical feature-based ensemble postprocessing of wind gusts within European winter storms" of the Transregional Collaborative Research Center SFB/TRR 165 "Waves to Weather" funded by the German Research Foundation (DFG). Sebastian Lerch gratefully acknowledges support by the Vector Stiftung through the Young Investigator Group "Artificial Intelligence for Probabilistic Weather Forecasting".

Data Availability The simulated data and the Wind data set can be obtained from the Github-repository https://github.com/benediktschulz/ADDE. Further, the repository describes how to obtain the other data sets.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Consent for publication All authors have given their consent for publication.

References

Aastveit, K. A., Mitchell, J., Ravazzolo, F., & Van Dijk, H. K. (2019). *The evolution of forecast density combinations in economics*. Oxford University Press.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*. http://tensorflow.org/

Abe, T., Buchanan, E.K., Pleiss, G., Zemel, R., & Cunningham, J.P. (2022). Deep ensembles work, but are they necessary? In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., & Oh, A. (eds.) *Advances in neural information processing systems*, vol. 35, pp. 33646–33660. https://proceedings.neurips.cc/paper_files/paper/2022/file/da18c47118a2d09926346f33bebde9f4-Paper-Conference.pdf

Allard, D., Comunian, A., & Renard, P. (2012). Probability aggregation methods in geoscience. *Mathematical Geosciences, 44*(5), 545–581. https://doi.org/10.1007/s11004-012-9396-3

Baran, S., & Lerch, S. (2016). Mixture EMOS model for calibrating ensemble forecasts of wind speed. *Environmetrics, 27*(2), 116–130.

Baran, S., & Lerch, S. (2018). Combining predictive distributions for the statistical post-processing of ensemble forecasts. *International Journal of Forecasting, 34*(3), 477–496.

Berkowitz, J. (2001). Testing density forecasts, with applications to risk management. *Journal of Business & Economic Statistics, 19*(4), 465–474.

Bishop, C. M. (1994). *Mixture density networks*. Technical report, available at https://publications.aston.ac.uk/id/eprint/373/1/NCRG_94_004.pdf.

Bojer, C. S., & Meldgaard, J. P. (2021). Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting, 37*(2), 587–603.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140. https://doi.org/10.1007/bf00058655

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Bremnes, J. B. (2020). Ensemble postprocessing using quantile function regression based on neural networks and Bernstein polynomials. *Monthly Weather Review, 148*(1), 403–414. https://doi.org/10.1175/mwr-d-19-0227.1

Busetti, F. (2017). Quantile aggregation of density forecasts. *Oxford Bulletin of Economics and Statistics, 79*(4), 495–512. https://doi.org/10.1111/obes.12163

Cevid, D., Michel, L., Näf, J., Bühlmann, P., & Meinshausen, N. (2022). Distributional random forests: Heterogeneity adjustment and multivariate distributional regression. *Journal of Machine Learning Research, 23*(333), 1–79.

Chapman, W. E., Monache, L. D., Alessandrini, S., Subramanian, A. C., Ralph, F. M., Xie, S.-P., Lerch, S., & Hayatbini, N. (2022). Probabilistic predictions from deterministic atmospheric river forecasts with deep learning. *Monthly Weather Review, 150*(1), 215–234. https://doi.org/10.1175/mwr-d-21-0106.1

Chen, J., Janke, T., Steinke, F., & Lerch, S. (2024). Generative machine learning methods for multivariate ensemble postprocessing. *The Annals of Applied Statistics, 18*(1), 159–183. https://doi.org/10.1214/23-AOAS1784

Chollet, F., Others (2015). Keras. https://keras.io

Clare, M. C. A., Jamil, O., & Morcrette, C. J. (2021). Combining distribution-based neural networks to predict weather forecast probabilities. *Quarterly Journal of the Royal Meteorological Society, 147*(741), 4337–4357. https://doi.org/10.1002/qj.4180

Cramer, E. Y., Ray, E. L., Lopez, V. K., Bracher, J., Brennen, A., Rivadeneira, A. J. C., Gerding, A., Gneiting, T., House, K. H., & Huang, Y. (2022). Others: Evaluation of individual and ensemble probabilistic forecasts of COVID-19 mortality in the United States. *Proceedings of the National Academy of Sciences of the United States of America, 119*(15), 2113561119. https://doi.org/10.1073/pnas.2113561119

Dietterich, T. G. (2000). Ensemble methods in machine learning. In: *Lecture notes in computer science*, pp. 1–15. https://doi.org/10.1007/3-540-45014-9_1.

D'Isanto, A., & Polsterer, K. L. (2018). Photometric redshift estimation via deep learning-generalized and pre-classification-less, image based, fully probabilistic redshifts. *Astronomy & Astrophysics, 609*, 111.

Durasov, N., Bagautdinov, T., Baque, P., & Fua, P. (2021). Masksembles for uncertainty estimation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13539–13548.

Fakoor, R., Kim, T., Mueller, J., Smola, A. J., & Tibshirani, R. J. (2023). Flexible model aggregation for quantile regression. *Journal of Machine Learning Research, 24*(162), 1–45.

Fort, S., Hu, H., & Lakshminarayanan, B. (2019). *Deep ensembles: A loss landscape perspective*. Preprint, available at. https://doi.org/10.48550/arXiv.1912.02757

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In: *Proceedings of the 13th international conference on machine learning*, pp. 148–156.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: *Proceedings of the 33rd international conference on machine learning*, pp. 1050–1059.

Gal, Y., Hron, J., & Kendall, A. (2017). Concrete dropout. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. (eds.) *Advances in neural information processing systems*, vol. 30. https://proceedings.neurips.cc/paper_files/paper/2017/file/84ddfb34126fc3a48ee38d7044e87276-Paper.pdf

Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., & Suganthan, P. N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence, 115*, Article 105151. https://doi.org/10.1016/j.engappai.2022.105151

Gasthaus, J., Benidis, K., Wang, Y., Rangapuram, S. S., Salinas, D., Flunkert, V., & Januschowski, T. (2019). Probabilistic forecasting with spline quantile function RNNs. In: *Proceedings of the twenty-second international conference on artificial intelligence and statistics*, pp. 1901–1910.

Genest, C. (1992). Vincentization revisited. *The Annals of Statistics, 20*(2), 1137–1142. https://doi.org/10.1214/aos/1176348676

Ghazvinian, M., Zhang, Y., Seo, D.-J., He, M., & Fernando, N. (2021). A novel hybrid artificial neural network - Parametric scheme for postprocessing medium-range precipitation forecasts. *Advances in Water Resources, 151*, Article 103907. https://doi.org/10.1016/j.advwatres.2021.103907

Gneiting, T., Balabdaoui, F., & Raftery, A. E. (2007). Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society Series B: Statistical Methodology, 69*(2), 243–268. https://doi.org/10.1111/j.1467-9868.2007.00587.x

Gneiting, T., & Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application, 1*(1), 125–151. https://doi.org/10.1146/annurev-statistics-062713-085831

Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association, 102*(477), 359–378. https://doi.org/10.1198/016214506000001437

Gneiting, T., & Ranjan, R. (2013). Combining predictive distributions. *Electronic Journal of Statistics, 7*(1), 1747–1782. https://doi.org/10.1214/13-EJS823

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In: *Proceedings of the 34th international conference on machine learning*, pp. 1321–1330.

Havasi, M., Jenatton, R., Fort, S., Liu, J. Z., Snoek, J., Lakshminarayanan, B., Dai, A. M., & Tran, D. (2021). Training independent subnetworks for robust prediction. In: *International conference on learning representations*. https://openreview.net/forum?id=OGg9XnKxFAH

Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., & Weinberger, K.Q. (2017). *Snapshot ensembles: Train 1, get m for free*. Preprint, available at https://doi.org/10.48550/arXiv.1704.00109

Januschowski, T., Wang, Y., Torkkola, K., Erkkilä, T., Hasson, H., & Gasthaus, J. (2022). Forecasting with trees. *International Journal of Forecasting, 38*(4), 1473–1481. https://doi.org/10.1016/j.ijforecast.2021.10.004

Jospin, L. V., Laga, H., Boussaid, F., Buntine, W., & Bennamoun, M. (2022). Hands-on Bayesian neural networks-a tutorial for deep learning users. *IEEE Computational Intelligence Magazine, 17*(2), 29–48. https://doi.org/10.1109/MCI.2022.3155327

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? In: *Proceedings of the 31st international conference on neural information processing systems*, pp. 5580–5590

Kingma, D. P., Salimans, T., & Welling, M. (2015). Variational dropout and the local reparameterization trick. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., & Garnett, R. (eds.) *Advances in neural information processing systems*, vol. 28. https://proceedings.neurips.cc/paper_files/paper/2015/file/bc7316929fe1545bf0b98d114ee3ecb8-Paper.pdf

Kobyzev, I., Prince, S. J. D., & Brubaker, M. A. (2021). Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(11), 3964–3979. https://doi.org/10.1109/TPAMI.2020.2992934

Koliander, G., El-Laham, Y., Djuric, P. M., & Hlawatsch, F. (2022). Fusion of probability density functions. *Proceedings of the IEEE, 110*(4), 404–453. arXiv:2202.11633.

Krüger, F., Lerch, S., Thorarinsdottir, T., & Gneiting, T. (2021). Predictive inference based on Markov chain Monte Carlo output. *International Statistical Review, 89*(2), 274–301.

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In: *Advances in neural information processing systems*, pp. 6403–6414.

Lampinen, J., & Vehtari, A. (2001). Bayesian approach for neural networks-review and case studies. *Neural Networks, 14*(3), 257–274. https://doi.org/10.1016/S0893-6080(00)00098-8
Lichtendahl, K. C., Grushka-Cockayne, Y., & Winkler, R. L. (2013). Is it better to average probabilities or quantiles? *Management Science, 59*(7), 1594–1611. https://doi.org/10.1287/mnsc.1120.1667
Li, R., Reich, B. J., & Bondell, H. D. (2021). Deep distribution regression. *Computational Statistics and Data Analysis, 159*, 107203. https://doi.org/10.1016/j.csda.2021.107203
Marcjasz, G., Narajewski, M., Weron, R., & Ziel, F. (2023). Distributional neural networks for electricity price forecasting. *Energy Economics, 125*, Article 106843. https://doi.org/10.1016/j.eneco.2023.106843
Matheson, J. E., & Winkler, R. L. (1976). Scoring rules for continuous probability distributions. *Management Science, 22*(10), 1087–1096.
Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research***7**(6).
Mohammed, A., & Kora, R. (2023). A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University - Computer and Information Sciences, 35*(2), 757–774. https://doi.org/10.1016/j.jksuci.2023.01.014
Mühlematter, D. J., Halbheer, M., Becker, A., Narnhofer, D., Aasen, H., Schindler, K., & Turkoglu, M. O. (2024). *LoRA-ensemble: Efficient uncertainty modelling for self-attention networks*. Preprint, available at https://doi.org/10.48550/arXiv.2405.14438.
Nix, D. A., & Weigend, A. S. (1994). Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE international conference on neural networks (ICNN'94)*, 1, 55–60.
Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., & Snoek, J. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In: *Advances in neural information processing systems*, p. 12.
Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Taieb, S. B., Bergmeir, C., Bessa, R. J., Bijak, J., & Boylan, J. E. (2022). Forecasting: Theory and practice. *International Journal of Forecasting, 38*(3), 705–871. https://doi.org/10.1016/j.ijforecast.2021.11.001
Python Software Foundation: Python software (2020). https://www.python.org/
Rahaman, R., & Thiery, A. H. (2020). Uncertainty quantification and deep ensembles. In: *Advances in neural information processing systems*.
Ranjan, R., & Gneiting, T. (2010). Combining probability forecasts. *Journal of the Royal Statistical Society. Series B: Statistical Methodology, 72*(1), 71–91. https://doi.org/10.1111/j.1467-9868.2009.00726.x
Rasp, S., & Lerch, S. (2018). Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review, 146*(11), 3885–3900. https://doi.org/10.1175/MWR-D-18-0187.1
Ratcliff, R. (1979). Group reaction time distributions and an analysis of distribution statistics. *Psychological Bulletin, 86*(3), 446–461. https://doi.org/10.1037/0033-2909.86.3.446
Ren, Y., Zhang, L., & Suganthan, P. N. (2016). Ensemble classification and regression-recent developments, applications and future directions. *IEEE Computational Intelligence Magazine, 11*(1), 41–53.
Satopää, V. A., Baron, J., Foster, D. P., Mellers, B. A., Tetlock, P. E., & Ungar, L. H. (2014). Combining multiple probability predictions using a simple logit model. *International Journal of Forecasting, 30*(2), 344–356. https://doi.org/10.1016/j.ijforecast.2013.09.009
Schlosser, L., Hothorn, T., Stauffer, R., & Zeileis, A. (2019). Distributional regression forests for probabilistic precipitation forecasting in complex terrain. *Annals of Applied Statistics, 13*(3), 1564–1589.
Schulz, B., & Lerch, S. (2022). Machine learning methods for postprocessing ensemble forecasts of wind gusts: A systematic comparison. *Monthly Weather Review, 150*(1), 235–257. https://doi.org/10.1175/mwr-d-21-0150.1
Schulz, B., & Lerch, S. (2024). *Machine learning methods for postprocessing ensemble forecasts of wind gusts: Data*. Karlsruhe Institute of Technology. Karlsruhe Institute of Technology. https://doi.org/10.35097/afEBrMYqNrxxvrLX. https://radar.kit.edu/radar/en/dataset/afEBrMYqNrxxvrLX
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*(1), 1929–1958.
Stone, M. (1961). The opinion pool. *The Annals of Mathematical Statistics, 32*(4), 1339–1342. https://doi.org/10.1214/aoms/1177704873
Taylor, J. W., & Taylor, K. S. (2021). Combining probabilistic forecasts of covid-19 mortality in the united states. *European Journal of Operational Research*.
Thomas, E. A. C., & Ross, B. H. (1980). On appropriate procedures for combining probability distributions within the same family. *Journal of Mathematical Psychology, 21*(2), 136–152. https://doi.org/10.1016/0022-2496(80)90003-6
Turkoglu, M. O., Becker, A., Gündüz, H. A., Rezaei, M., Bischl, B., Daudt, R. C., D'Aronco, S., Wegner, J., & Schindler, K. (2022). Film-ensemble: Probabilistic deep learning via feature-wise linear modulation. *Advances in Neural Information Processing Systems, 35*, 22229–22242.
Vincent, S. B. (1912). The functions of the Vibrissae in the behavior of the white rat. *Animal Behavior Monographs*, 1.

Vogel, P., Knippertz, P., Fink, A. H., Schlueter, A., & Gneiting, T. (2018). Skill of global raw and postprocessed ensemble predictions of rainfall over northern tropical Africa. *Weather and Forecasting, 33*(2), 369–388. https://doi.org/10.1175/WAF-D-17-0127.1

Wang, X., Hyndman, R. J., Li, F., & Kang, Y. (2023). Forecast combinations: An over 50-year review. *International Journal of Forecasting, 39*(4), 1518–1547. https://doi.org/10.1016/j.ijforecast.2022.11.005

Wasserman, L. (2004). *All of statistics: A concise course in statistical inference*. Springer Texts in Statistics. Springer, New York. https://doi.org/10.1007/978-0-387-21736-9

Wen, Y., Tran, D., & Ba, J. (2020). *BatchEnsemble: An alternative approach to efficient ensemble and lifelong learning*.

Wolffram, D. (2021). *Building and evaluating forecast ensembles for COVID-19 deaths*. M.Sc. thesis, Karlsruhe Institute of Technology.

Wu, X., & Gales, M. (2021) *Should ensemble members be calibrated?* Preprint, available at https://doi.org/10.48550/arXiv.2101.05397.

Zhou, Z.-H., Wu, J., & Tang, W. (2002). Ensembling neural networks: Many could be better than all. *Artificial Intelligence, 137*(1–2), 239–263.

Authors and Affiliations

Benedikt Schulz[1] · Lutz Köhler[2] · Sebastian Lerch[2,3]

✉ Benedikt Schulz
benedikt.schulz.info@gmail.com

Lutz Köhler
koehler.lutz@outlook.de

Sebastian Lerch
lerch@mathematik.uni-marburg.de

[1] Institute for Stochastics, Karlsruhe Institute of Technology (KIT), Englerstr. 2, 76128 Karlsruhe, Germany

[2] Institute of Statistics, Karlsruhe Institute of Technology (KIT), Blücherstr. 17, 76185 Karlsruhe, Germany

[3] Department of Mathematics and Computer Science, Philipps-Universität Marburg, Hans-Meerwein-Straße 6, 35043 Marburg, Germany