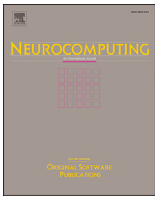
Neurocomputing 682 (2026) 133394

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Deep interpretable ensembles

Lucas Kook [a,*], Andrea Götschi [b], Philipp F.M. Baumann [c], Torsten Hothorn [b], Beate Sick [b,d]

[a] Institute for Statistics & Mathematics, Vienna University of Economics and Business, 1020 Vienna, Austria
[b] Epidemiology, Biostatistics & Prevention Institute, University of Zurich, 8001 Zurich, Switzerland
[c] KOF Swiss Economic Institute, ETH Zurich, 8092 Zurich, Switzerland
[d] Thurgau Institute for Digital Transformation, 8280 Kreuzlingen, Switzerland

ARTICLE INFO

Communicated by P.N. Suganthan

Keywords:
Deep ensembles
Interpretability
Normalizing flows
Transformation models
Uncertainty quantification

ABSTRACT

Ensembles improve prediction performance and uncertainty quantification by aggregating predictions from multiple models. In deep ensembling, these individual models can be black box neural networks or more interpretable models, such as neural additive models. However, interpretability of the ensemble members is generally lost when computing ensemble predictions. This is a crucial drawback of deep ensembles in high-stake decision fields, in which interpretable models are desired. We propose a novel transformation ensemble which aggregates probabilistic predictions with the guarantee to preserve interpretability and yield uniformly better predictions than the ensemble members on average. Transformation ensembles are tailored towards a class of normalizing flows, called deep transformation models, but are applicable to a wider range of probabilistic neural networks. In experiments on several publicly available data sets, we demonstrate that transformation ensembles perform on par with classical deep ensembles in terms of prediction performance, discrimination, and calibration. In addition, we demonstrate how transformation ensembles capture both aleatoric and algorithmic uncertainty, and produce minimax optimal predictions under certain conditions.

1. Introduction

The need for interpretable yet flexible and well-performing prediction models is great in high-stakes decision fields, such as medicine. Practitioners need to be able to understand how a model arrives at its predictions, and how confident these predictions are, to assess a model's trustworthiness. For this purpose, Rudin [44] proposes the use of intrinsically interpretable models: A model is deemed intrinsically interpretable if it possesses a transparent structure, such as sparsity or additivity with parameters that can be interpreted, *e.g.,* as log odds-ratios or log hazard-ratios. For instance, traditional statistical regression models, *e.g.,* linear regression, or the Cox proportional hazards model [10], and also decision trees fall in this category. However, contemporary applications may involve more complex data or require methods, for which intrinsic interpretability can only partly be achieved. For models that are not intrinsically interpretable there exist *post hoc* or model-agnostic explanation methods [for an overview, see [38]].

Nowadays the data available for prediction is often a mix of structured tabular and unstructured data, such as images, text, or speech. The prediction target can be continuous (*e.g.,* time-to-event), or discrete (*e.g.,* number of recurrences, or an ordinal score, such as tumour-grading). Hence, the ideal prediction model should in summary (i) be as interpretable as possible, (ii) yield accurate and calibrated probabilistic predictions, and (iii) be applicable to multi-modal input data and different kinds of outcomes. One group of models fulfilling these requirements are deep transformation models [2,32,45,47]. Deep transformation models estimate the conditional distribution of an outcome given the supplied data modalities. Transformation models are specified by a transformation function that maps between an a priori chosen latent distribution and the conditional outcome distribution [24]. Thus, transformation models are related to univariate conditional normalizing flows [41], where the transformation function corresponds to the inverse conditional flow and the latent distribution corresponds to the target distribution in normalizing flows (oftentimes the standard normal distribution). Optional structural assumptions on the transformation function, such as additivity, sparsity or linearity make the models more intrinsically interpretable. However, a single instance of such a model may suffer from overfitting and overconfident predictions or get stuck in local optima during training, which may lead to severe prediction errors when applied to novel data.

* Corresponding author.
Email address: lucas.kook@wu.ac.at (L. Kook).

https://doi.org/10.1016/j.neucom.2026.133394
Received 7 October 2024; Received in revised form 5 September 2025; Accepted 17 March 2026
Available online 21 March 2026

To build well-performing models, it is common practice to aggregate the predictions of several models, mitigating overconfident and potentially inaccurate predictions [9,13,14]. Aggregating the predictions of several models is referred to as ensembling, and the resulting final prediction model is called an ensemble. Deep ensembling, *i.e.,* aggregating predictions of (often as few as 3–5) deep neural networks that are fitted on the same data but with different random initializations, has been demonstrated to notably improve prediction performance [34]. We refer to this approach as classical deep ensembling. However, classical deep ensembles are black boxes even if their members are somewhat interpretable individually, because averaging probability predictions from highly non-linear models reduces explainability.

Alongside improved prediction performance, deep ensembles have been advocated to improve uncertainty quantification and robustness towards distributional shifts in test data via measuring heterogeneity between ensemble members [34]. However, this view has been challenged by attributing improved behaviour out-of-distribution to the use of more flexible single models rather than ensembles of simpler models [1].

In this paper, we propose transformation ensembles, which preserve structure and interpretability of their members. In addition, transformation ensembles improve predictions and allow algorithmic uncertainty quantification with distributional deep neural networks.

1.1. Our contribution

We present transformation ensembles as a novel way to aggregate predicted cumulative distribution functions (CDFs) derived from deep neural networks with a special focus on deep transformation models for (potentially) semi-structured data. We show that transformation ensembles (i) preserve structure and interpretability of their members by averaging predictions on a model-related latent scale which preserves additive model structures (see Section 3), (ii) improve the prediction performance akin to classical deep ensembles, and (iii) minimize worst-case prediction error in special cases. We showcase these properties theoretically and demonstrate the benefits of transformation ensembles empirically on several semi-structured data sets. With transformation ensembles we are able to provide empirical evidence for answering open questions in deep ensembling [1]. Namely, (iv) the increased flexibility of classical deep ensembles over their members does not seem to be necessary for improving prediction performance; (v) single more flexible models oftentimes match the prediction performance of an ensemble consisting of less flexible models. Lastly, (vi) ensembles allow a form of *algorithmic uncertainty* quantification. We choose the term algorithmic uncertainty because all ensemble members are trained on the same data and thus heterogeneity between members stems solely from random initialization and stochastic optimization of model parameters.

Structure of this paper. We discuss how transformation ensembles relate to classical deep ensembles and other ensembling strategies from the forecasting literature in Section 1.2. We give the necessary background on scoring rules, ensembling and transformation models in Section 2. Afterwards, we present transformation ensembles (Section 3) and demonstrate how they improve prediction performance similar to deep ensembles while preserving interpretability. We present case studies of deep ensembling on several real-world datasets with both continuous (Section 4) and discrete outcomes (Section 5). We conclude with a discussion of our results and potential directions for future research (Section 6).

1.2. Related work

We recap ensembling techniques from machine learning, deep learning and ensembles. Note that we focus on probabilistic prediction in this article, but not on point prediction. For completeness we mention some ensembles for point prediction, *i.e.,* classification and conditional means, below.

Ensembling in machine learning. Ensembling methods have been used for some decades now in statistics and machine learning [see [9] for an overview]. For instance, Hansen and Salamon [20] aggregated classifications from neural networks with different initializations by majority vote and Breiman [4] proposed to aggregate models or predictions obtained from bootstrap samples (bagging). In practice, decision trees as base models have been vastly successful and the random forest [6,51] is probably the most well-known ensemble algorithm. The main goal of these ensemble methods is to improve upon the individual members' prediction performance in terms of both calibration and sharpness, rather than to obtain interpretable models [*e.g.,* [7]]. Besides random forests, boosting is a highly versatile ensemble approach based on combining simple base-learners into an ensemble [36,46]. Both random forests and boosting have been formulated for transformation models [22,23]. A parallel line of research focuses on how to generally combine probabilistic forecasts other than conditional means [16], including aggregating densities or distribution functions, hazards [28], or quantile functions [42].

Classical deep ensembles. Classical deep ensembles combine predictions of several deep neural networks, by training several random instances of the same model on the same data and averaging their predictions [34]. Deep ensembling has been applied across a wide range of domains, including medical [55] and environmental sciences [52]. For a recent overview of deep ensembles in speech recognition, see Tanveer et al. [49]. Opitz et al. [40] and Ju et al. [30] average last-layer outputs before passing them through a softmax activation in multiclass classification problems. In contrast to the bagging algorithms discussed above, heterogeneity of deep ensemble members does not stem from bootstrapping the data, but rather from stochasticity in initializing and optimizing the neural networks. Recent deep learning architectures that are tailored towards interpretability are proposed in Izonin et al. [29].

Several contributions suggest that deep ensembles benefit prediction performance, uncertainty quantification and out-of-distribution generalization [53]. Abe et al. [1] question the extent to which these benefits hold. For instance, the authors suggest that more complex models may show a gain in prediction performance similar to classical ensembles. As in random forests, ensembling conditional mean functions instead of probabilistic predictions is also possible. However, we do not further address conditional mean ensembles in this article.

Quasi-arithmetic pooling with proper scoring rules and minimax optimality. Scoring rules are metrics designed to evaluate probabilistic predictions [15]. An ideal score judges predictions from a model by how faithful they are to the data-generating distribution and thus penalizes overconfident as well as too uncertain predictions. Thus, proper scoring rules [15] are a natural choice to evaluate probabilistic forecasts (see Appendix C for more detail). Neyman and Roughgarden [39] study the relation between proper scoring rules and different types of aggregations used for ensembling. In particular, for nominal outcomes, Neyman and Roughgarden [39] study ensembling densities, $p_1, \dots, p_M$ from M models based on *quasi-arithmetic pooling*,

$$p_M^g = g^{-1}\left(\sum_m w_m g \circ p_m\right), \tag{1}$$

where g is any continuous, non-decreasing function and w_m are non-negative weights summing to one. The authors prove that for nominal outcomes, certain combinations of scoring rules and ensembling methods are minimax optimal, guaranteeing that the worst-case prediction error as measured by the scoring rule is minimized (compared to the average prediction error). When, for example, evaluating ensemble predictions based on Brier's quadratic score [[8] see Appendix C for the definition], aggregating predictions with the arithmetic mean is minimax optimal, corresponding to g being the identity. For the negative log-likelihood, a geometric mean aggregation is minimax optimal, corresponding to g being the natural logarithm.

2. Background

We introduce classical ensembles and neural networks. Then, we give a short overview of deep transformation models as the backbone of our proposed transformation ensembles. Here, we focus on probabilistic predictions of an outcome $Y \in \mathcal{Y} \subseteq \mathbb{R}$ given predictors $\boldsymbol{D} \in \mathcal{D}$ based on observations from a joint distribution $\mathbb{P}_{(Y,\boldsymbol{D})}$. The outcome may be binary, ordinal, count-valued, or continuous. The predictors may be tabular or non-tabular, such as images or text, or both. Throughout the manuscript, we reserve $\mathbf{X} \in \mathcal{X} \subseteq \mathbb{R}^d$ for tabular, $\boldsymbol{T} \in \mathcal{T}$ for text and $\boldsymbol{B} \in \mathcal{B}$ for image data. The predictions are assumed to be in the form of a conditional CDF $F : \mathcal{Y} \times \mathcal{D} \to [0,1]$, where $(y, \boldsymbol{d}) \mapsto F(y \mid \boldsymbol{d}) := \mathbb{P}(Y \leq y \mid \boldsymbol{D} = \boldsymbol{d})$. Further, let $\mathcal{P}$ denote the set of all probability measures on $\mathcal{Y} \times D$.

2.1. Ensembles

Ensembles aggregate predictions from multiple models. We assume access to M conditional CDFs $F_1, \dots, F_M$ of M models. The most commonly used ensemble methods are the classical linear and log-linear ensembles, which we discuss in the following.

Definition 1 (Classical linear ensemble). Let $F_1, \dots, F_M$ be M CDFs and $w_1, \dots, w_M$ be non-negative weights summing to one. The classical linear ensemble is defined as the point-wise weighted average of the M ensemble members $\bar{F}_M^{LIN} = \sum_m w_m F_m$. When using equal weights, the linear ensemble is equivalent to taking the arithmetic mean, $M^{-1} \sum_m F_m$.

The distribution $\bar{F}_M^{LIN}$ of the classical linear ensemble can be viewed as a mixture distribution with weights w_m. Note that for linear ensembles it does not matter whether one ensembles on the scale of the CDF or the probability density function (PDF), because for all $y \in \mathcal{Y}$ and $\boldsymbol{d} \in \mathcal{D}$, $\bar{f}_M^{LIN}(y \mid \boldsymbol{d}) = \sum_m w_m f_m(y \mid \boldsymbol{d}) = \frac{\mathrm{d}}{\mathrm{d}v} \bar{F}_M^{LIN}(v \mid \boldsymbol{d})|_{v=y}$ for continuous CDFs.

In this article we formulate all ensembles on the scale of the CDF if it is well-defined (*i.e.,* for random variables with at least an ordered sample space). Note, however, that in *e.g.,* multi-class classification, it is common to ensemble on the density scale. Here, the predicted probabilities for each class $p(k \mid \boldsymbol{d}) := \mathbb{P}(Y = y_k \mid \boldsymbol{D} = \boldsymbol{d})$ are aggregated via $\bar{p}_M(k \mid \boldsymbol{d}) = \sum_m w_m p_m(k \mid \boldsymbol{d}),\ k = 1, \dots, K$. Performing linear ensembling on the density scale with deep neural networks is called deep ensembling (Section 1.2).

For convex loss functions, it is well-known that the performance of the classical linear ensemble is always at least as good as the average performance of its members [Proposition 1, see also[1]].

Proposition 1. *Let $\mathcal{F}$ denote the set of all conditional CDFs on $\mathcal{Y} \times \mathcal{D}$. Let $F_1, \dots, F_M \in \mathcal{F}$ as CDFs with $w_1, \dots, w_M \in \mathbb{R}_+$ be non-negative weights summing to one. Let $L : \mathcal{F} \times \mathcal{Y} \times D \to \mathbb{R}$ be a convex loss function. Then, for all $y \in \mathcal{Y}$ and $\boldsymbol{d} \in \mathcal{D}$, we have $L\left(\sum_m w_m F_m, y, \boldsymbol{d}\right) \leq \sum_m w_m L\left(F_m, y, \boldsymbol{d}\right)$.*

The claim follows directly from Jensen's inequality by convexity of L. In particular, this holds for the negative log-likelihood (NLL), where, for continuous outcomes, $L : (F, y, \mathbf{X}) \mapsto -\log \frac{\mathrm{d}}{\mathrm{d}v} F(v \mid \boldsymbol{d})|_{v=y}$, and the ranked probability score (RPS) as loss functions (for definitions, see Appendix C).

The arithmetic mean is not the only way to aggregate forecasts. The geometric mean (arithmetic mean on the log-scale) is used for log-linear ensembles, as defined in the following.

Definition 2 (Classical log-linear ensemble). The log-linear ensemble is defined as the point-wise geometric mean of the M ensemble members $\bar{F}_M^{LOG} = \exp\left(\sum_m w_m \log F_m\right)$.

The log-linear ensemble, as defined here, is a special case of quasi-arithmetic pooling with $g = \log$ on the scale of the CDF, see Eq. (1). As mentioned above, commonly (*i.e.,* in classification problems with nominal outcomes) densities are aggregated in log-linear ensembles which require scaling by a constant such that the ensemble density integrates to one. Regardless of whether log PDFs or log CDFs are pooled, the ensemble will still score better in terms of negative log-likelihood than its members do on average (see Prop. 2 and 3 in Appendix D).

2.2. Transformation models with neural networks

Although our proposed ensembling scheme (see Definition 3) is applicable to any probabilistic deep neural network, it is most beneficial when all members are deep transformation models. Therefore, we briefly recap (conditional) transformation models [24] and their extensions involving deep neural networks [32,33,47]. Transformation models have originally been introduced as models for the conditional distribution of a univariate outcome $Y \in \mathcal{Y}$ given tabular predictors $\mathbf{X} \in \mathcal{X} \subseteq \mathbb{R}^d$. Transformation models allow to model flexible conditional outcome distributions without the need to pre-specify a parametric family of distributions, such as a Gaussian mixture. Instead, transformation models are specified by a latent distribution $F_Z : \mathbb{R} \to [0,1]$ with log-concave and continuous (w.r.t. Lebesgue measure) density f_Z, and a transformation function $h : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}$ which, for all $\mathbf{X} \in \mathcal{X}$, is monotonically increasing in $y \in \mathcal{Y}$. Finally, the conditional outcome distribution, $F(y \mid \mathbf{X}) := \mathbb{P}(Y \leq y \mid \mathbf{X} = \mathbf{X})$, is modelled as

$$F(y \mid \mathbf{X}) = F_Z(h(y \mid \mathbf{X})), \tag{2}$$

with $Z := h(Y \mid \mathbf{X})$ being a latent random variable with CDF F_Z which is assumed to be independent of $\mathbf{X}$, and the goal is to estimate the transformation h from data (Fig. 1).

In their general form, transformation models are distribution-free [3], since no predefined family has to be chosen for the conditional outcome distribution. Transformation models generalize and extend several well-known models, such as the normal linear regression model, cumulative ordinal regression, and the Cox proportional hazards model [25]. For instance, the normal linear regression model presupposes the conditional outcome distribution $N(\mathbf{X}^\top \boldsymbol{\varphi}, \sigma^2)$, and can be represented as in Eq. (2) with a standard normal as latent distribution $F_Z = \Phi$, and a transformation function $h : (y, \mathbf{X}) \mapsto \sigma^{-1} y - \sigma^{-1} \mathbf{X}^\top \boldsymbol{\varphi}$ that is linear in y. Here, the parameters $\boldsymbol{\varphi}$ can be interpreted as differences in conditional means, but the conditional outcome distribution is fixed to be normal. More general transformation retain the interpretability, but can leave restrictive distributional assumptions behind by specifying a more flexible transformation function. For instance, $h : (y, \mathbf{X}) \mapsto h_Y(y) - \mathbf{X}^\top \beta$ with a $h_Y : \mathcal{Y} \to \mathbb{R}$ that is non-linear and monotonically increasing, allows for non-normal outcome distributions (as for instance, the bounded continuous outcome in Fig. 1(A) but a similar interpretation of β as shifts in transformed conditional expectation. Other common choices for F_Z are the standard logistic, and standard minimum extreme value distribution, which lead to different interpretational scales (log-odds, or log-hazards, respectively). Transformation models are closely related to normalizing flows in deep learning [41].

Neural networks. Neural networks (NNs) are flexible compositions of linear and non-linear functions [43]. NNs are able to map inputs from complex spaces, such as (multi-dimensional) images, text, or other unstructured data to Euclidean spaces [18]. Feed-forward neural networks include, for instance (if neither hidden layers nor non-linear activation functions are involved), simple linear functions as a special case, $f : \mathbb{R}^p \to \mathbb{R}$, $f : \mathbf{X} \mapsto \mathbf{X}^\top \beta$, where $\beta \in \mathbb{R}^p$, which are commonly referred to as "linear predictors" in the statistics literature [50]. However, much more complex NNs can be formulated for tabular and non-tabular (unstructured) data, whose output can be mapped into $\mathbb{R}$ and then used together with out-of-the-box mini-batch gradient descent to augment classical statistical models, as described in the next paragraph. An example of a more complex NN is a convolutional NN which maps from an image $\mathsf{B} \in \mathbb{R}^{w \times h \times c}$, $w, h, c \in \mathbb{N}$ to, for instance, $\mathbb{R}$. In Appendix B, we give details on the NN architectures used in our experiments, including convolutional NNs and NNs with text-embedding. For a more detailed overview of deep learning and NN architectures, see Goodfellow et al. [18].

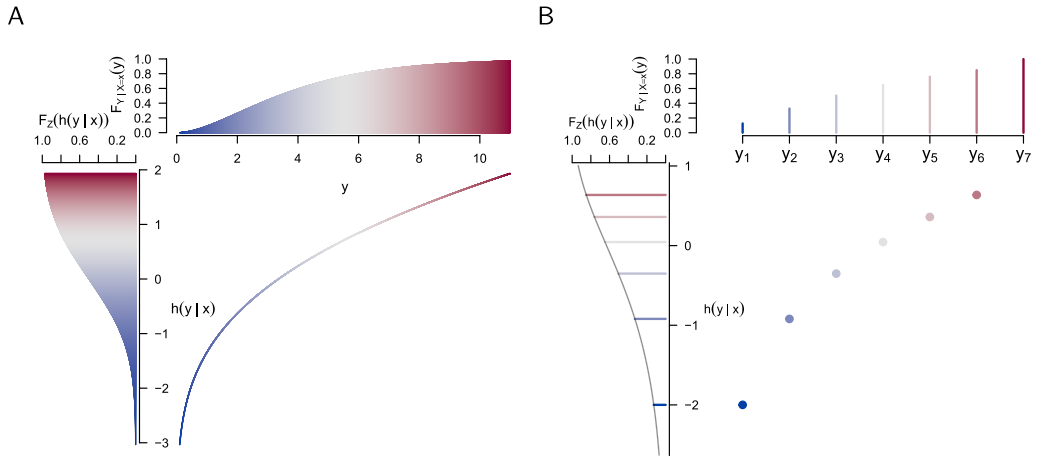


Fig. 1. Transformation models for continuous (A) and ordinal (B) outcomes are constructed such that $\mathbb{P}(Y \leq y \mid \mathbf{X}) = \mathbb{P}(h(Y \mid \mathbf{X}) \leq h(y \mid \mathbf{X}))$ holds. The monotone increasing continuous or discrete transformation function (lower right panel in A and B) is estimated from the data by minimizing an empirical proper score.

Deep transformation models. Recently, transformation models have been combined with neural networks [2,32,47] to overcome the limitation of classical statistical models of not being able to incorporate unstructured data without any feature engineering. Sick et al. [47] demonstrate how transformation models for continuous outcomes can be set up with neural networks to handle tabular and/or image data, however the authors focused solely on predictive power and not on interpretability. Baumann et al. [2] and Kook & Herzog et al. (2022) explore intrinsic interpretability of deep transformation models. Recently, Kook et al. [33] have established connections to normalizing flows [41]. In the latter, the authors propose models for semi-structured input data and ordinal outcomes (ONTRAMs). In ONTRAMs the transformation function is parameterized via neural networks and the model parameters are optimized jointly by minimizing the NLL. This way of combining traditional statistical methods with deep learning allows the construction of interpretable, yet powerful prediction models. For instance, having access to tabular data $\mathbf{X}$ and image data $\boldsymbol{b}$, we can formulate models for the conditional distribution of Y, such as

$$F(\cdot \mid \mathbf{X}, \boldsymbol{b}) = F_Z\left(h_Y(\cdot) - \mathbf{X}^\top \beta - \eta(\boldsymbol{b})\right), \quad \text{or}$$
$$F(\cdot \mid \mathbf{X}, \boldsymbol{b}) = F_Z\left(h_{Y,\boldsymbol{B}}(\cdot \mid \boldsymbol{b}) - \mathbf{X}^\top \beta\right).$$

The first model assumes a simple linear shift ($\mathbf{X}^\top\beta$) for the tabular data and a complex shift ($\eta(\boldsymbol{b})$) for the image data, while the second model allows full flexibility in the image data, but still assumes linear shift effects for the tabular data. When using $F_Z(z) = expit(z) = (1+\exp(-z))^{-1}$, the shift terms can be interpreted as log odds-ratios. Then the first model assumes proportional odds for both $\mathbf{X}$ and $\boldsymbol{b}$. The second model lifts this restriction on the image component. The components of the transformation function are controlled by (deep) neural networks, *i.e.,* a convolutional neural network for $\boldsymbol{b}$ and a single-layer NN for $\mathbf{X}$ (for details see Kook & Herzog et al. 2022). The following example illustrates how deep transformation models can be made more intrinsically interpretable compared to black-box deep neural networks.

Example 1 (Interpretability of deep transformation models). We are interested in modelling the conditional distribution of revenue of a movie ($Y \in \mathbb{R}_+$) given a popularity rating of the movie ($X \in \mathbb{R}_+$, higher values indicate higher popularity) and a movie review ($\boldsymbol{T} \in \mathcal{T}$, text data). First, consider the continuous-outcome logistic regression model [35], which for all $y \in \mathbb{R}$ and $x \in \mathbb{R}_+$ is given by,

$$\mathbb{P}(Y \leq y \mid X = x) \approx expit(\boldsymbol{a}_{\text{Bs},P}(\log(1+y))^\top \boldsymbol{\vartheta} - x\beta),$$
$$\text{s.t. } \vartheta_1 \leq \vartheta_2 \leq \cdots \leq \vartheta_{P+1},$$

where $\boldsymbol{a}_{\text{Bs},P} : \mathbb{R} \to \mathbb{R}^{P+1}$ denotes a basis of polynomials in Bernstein form [12]. The constraints are needed to ensure monotonicity of the modelled cumulative distribution function [25,37]. Here, $\exp(\beta)$ is interpretable as the odds ratio of a movie bringing higher revenue rather than lower revenue, when the movie's popularity increases by one unit (for all revenues $y \in \mathbb{R}$ simultaneously). Now we extend the model by including the movie reviews,

$$\mathbb{P}(Y \leq y \mid X = x, \boldsymbol{T} = \boldsymbol{t}) \approx expit(\boldsymbol{a}_{\text{Bs},P}(\log(1+y))^\top \boldsymbol{\vartheta} - x\beta - \eta(\boldsymbol{t})),$$

where $\eta : \mathcal{T} \to \mathbb{R}$ denotes a NN which maps the non-tabular text input to the real numbers. The interpretation of β remains the same, when keeping the movie review constant. Further, exponentiated differences in the output of η for two movie reviews $\boldsymbol{t}_0, \boldsymbol{t}_1 \in \mathcal{T}$ are now also interpretable as odds ratios, $\exp(\eta(\boldsymbol{t}_1) - \eta(\boldsymbol{t}_0))$, given the two movies are equally popular. If one is not willing to assume proportional odds (*i.e.,* β is the log odds ratio for all possible realizations of Y), the model can be made more flexible by including the features (tabular and text) as complex intercept terms [32]. Lastly, note that the model components $\boldsymbol{\vartheta}$, β, η can be controlled via constrained NNs. The constraints on $\boldsymbol{\vartheta}$ can be implemented with a custom last-layer activation (see Appendix B), $x\beta$ is a simple feed-forward NN without any hidden layers, a linear activation function and no bias term, and with a recurrent neural network [21] for η. We fit the above models and apply classical and transformation ensembling in Section 4.

3. Transformation ensembles

Deep transformation models can be used as flexible, yet interpretable prediction models with semi-structured data, making them an attractive choice of model class. To further improve their predictive power, one could use classical deep ensembling. However, when pooling the CDFs of transformation models linearly, the ensemble loses the structural assumptions of its individual members (*e.g.,* proportional odds), and thus intrinsic interpretability is lost in general (see Fig. 2(A). For instance, the average of two Gaussian densities is generally not Gaussian anymore and neither unimodal nor necessarily symmetric. To mitigate the black-box character of classical ensembles, we propose *transformation ensembles*. Transformation ensembles are specifically tailored towards transformation models for which predictions are aggregated on the scale of the transformation function h. That is, for all $\boldsymbol{d} \in \mathcal{D}$, the transformation ensemble with members $F_m(\cdot \mid \boldsymbol{d}) = F_Z\left(h_m(\cdot \mid \boldsymbol{d})\right)$ is given by $\tilde{F}_M^{TRF}(\cdot \mid \boldsymbol{d}) = F_Z\left(\sum_m w_m h_m(\cdot \mid \boldsymbol{d})\right)$.

Definition 3 (Transformation ensemble). Let $F_1, \ldots, F_M$ be CDFs and $w_1, \ldots, w_M$ be non-negative weights summing to one. Let $F_Z : \mathbb{R} \to [0,1]$

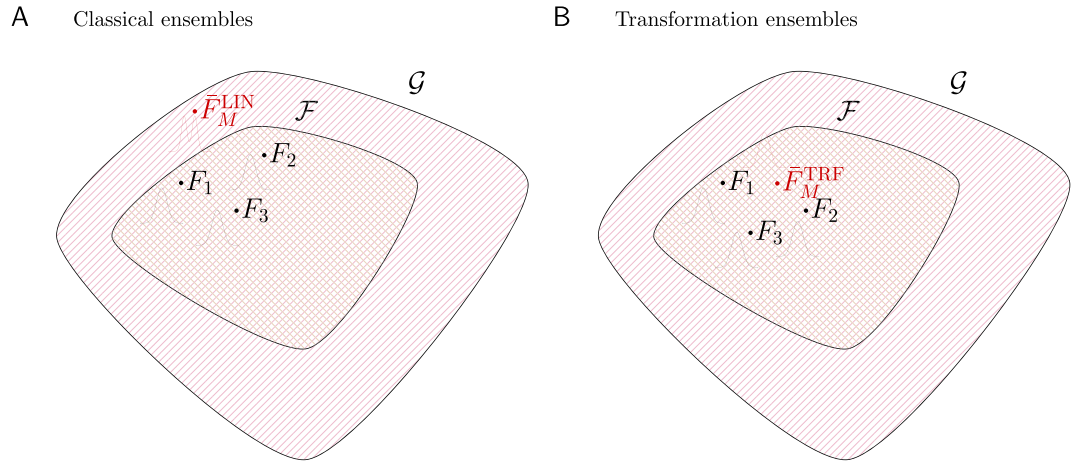


Fig. 2. Comparing classical and transformation ensembles. A: In classical ensembles, structural assumptions on the individual $F_m \in \mathcal{F}$ do not necessarily carry over to the ensemble $\bar{F}_M^{LIN} \in \mathcal{G} \supset \mathcal{F}$. Note, that there exist special cases in which the classical ensemble may remain in $\mathcal{F}$, for instance if all ensemble members are equal, $F_m = F$, $\forall m$. For example normal densities corresponding to the ensemble members can produce a multimodal ensemble density under classical ensembling, thus leaving the model class. B: The transformation ensemble, on the other hand, ensures that$\forall M$ $F_m \in \mathcal{F} \implies \bar{F}_M^{TRF} \in \mathcal{F}$. For the example densities, the transformation ensemble corresponds to averaging the members' transformation functions, ensuring a normal ensemble density. Further, if the transformation function is linear (as in normal linear regression), the transformation ensemble corresponds to averaging the (scaled) regression coefficients, yielding another normal linear regression model.

be a continuous CDF with quantile function ${F_Z}^{-1}$ and continuous log-concave density. For all $\boldsymbol{d} \in \mathcal{D}$, the transformation ensemble is defined as

$$\bar{F}_M^{TRF}(\cdot \mid \boldsymbol{d}) = F_Z\left(\sum_m w_m {F_Z}^{-1} \circ F_m(\cdot \mid \boldsymbol{d})\right). \tag{3}$$

Further, if for all $\boldsymbol{d} \in \mathcal{D}$, each member is a transformation model, *i.e.*, $F_m(\cdot \mid \boldsymbol{d}) = F_Z\left(h_m(\cdot \mid \boldsymbol{d})\right)$, the transformation ensemble simplifies to $\bar{F}_M^{TRF}(\cdot \mid \boldsymbol{d}) = F_Z\left(\sum_m w_m h_m(\cdot \mid \boldsymbol{d})\right)$. For continuous outcomes, the transformation ensemble density is given by the transformation model density evaluated at the average transformation function, *i.e.*, $\bar{f}_M(y \mid \boldsymbol{d}) = f_Z(\bar{h}_M(y \mid \boldsymbol{d}))\bar{h}'_M(y \mid \boldsymbol{d})$, where $\bar{h}_M(\cdot \mid \boldsymbol{d}) = \sum_m w_m h_m(\cdot \mid \boldsymbol{d})$ and for all $y \in \mathcal{Y}$, $\bar{h}'_M(y \mid \boldsymbol{d}) = \frac{\mathrm{d}}{\mathrm{d}v}\bar{h}_M(v \mid \boldsymbol{d})|_{v=y}$.

Note that transformation ensembles in (3) require only the existence of the CDF of each ensemble member. These CDFs may stem from any deep neural network, which need not necessarily be a transformation model. A transformation ensemble can still be constructed after picking any continuous CDF F_Z.

A transformation ensemble of transformation models with the same F_Z is again a transformation model (with the same F_Z as its members, see Fig. 2). Thus, transformation ensembles remain as intrinsically interpretable as their members, while still producing provably better than average predictions, in case their members are transformation models (see Prop. 1 in Appendix D for a formal statement). This property is especially interesting for more special cases, such as linear and semi-structured deep transformation models, as illustrated in the following example.

Example 2 (Semi-structured transformation models). Consider a model for $Y \mid \mathbf{X} = \mathbf{X}, \boldsymbol{B} = \boldsymbol{b}$, given by $F_m(\cdot \mid \mathbf{X}, \boldsymbol{b}) = F_Z \circ h_m(\cdot \mid \mathbf{X}, \boldsymbol{b})$, where $F_Z(z) = 1 - \exp(-\exp(z))$ is the standard minimum extreme value CDF and $h_m(\cdot \mid \mathbf{X}, \boldsymbol{b}) := \boldsymbol{a}(\cdot)^\top \boldsymbol{\vartheta}_m - \mathbf{X}^\top \beta_m - \eta_m(\mathsf{B})$. The model assumes proportional hazards for both structured ($\mathbf{X}$) and unstructured data ($\boldsymbol{b}$). The resulting transformation ensemble retains the proportional hazard assumption, because for all $y \in \mathcal{Y}$, $\bar{h}_M(y \mid \mathbf{X}, \boldsymbol{b}) = \boldsymbol{a}(y)^\top \bar{\boldsymbol{\vartheta}}_M - \mathbf{X}^\top \bar{\beta}_M - \bar{\eta}_M(\boldsymbol{b})$ averages the predicted *log cumulative-hazards*. In contrast, random survival forests [28] aggregate cumulative hazards without focusing on interpretability. Note that the outputs of the convolutional neural networks, $\eta_m(\boldsymbol{b})$, are averaged, not their weights. The classical linear ensemble would not preserve the model structure, because there exist $y \in \mathcal{Y}$, $\mathbf{X} \in \mathcal{X}$ and $\boldsymbol{b} \in \mathcal{B}$, such that $\sum_m w_m F_Z(h_m(y \mid \mathbf{X}, \boldsymbol{b})) \neq F_Z\left(\sum_m w_m h_m(y \mid \mathbf{X}, \boldsymbol{b})\right)$ due to the non-linearity of F_Z.

In Proposition 2 below, we show that the NLL of the transformation ensemble is uniformly better than the average NLL of its members. This is analogous to the improved prediction performance of classical linear and log-linear ensembles w.r.t. the NLL. A proof is given in Appendix D.

Proposition 2. *Let $\mathcal{H}$ be the class of transformation functions (see Proposition 1 for a formal definition). Let F_Z be a fixed latent distribution, $h_1, \dots, h_M \in \mathcal{H}$, and $w_1, \dots, w_M$ be non-negative weights summing to one. Let $NLL : \mathcal{F} \times \mathcal{Y} \times \mathcal{D} \to \mathbb{R}$ denote the negative log-likelihood, i.e., for continuous outcomes $(F_Z \circ h, y, \boldsymbol{d}) \mapsto -\log f_Z(h(y \mid \boldsymbol{d}))\frac{\mathrm{d}}{\mathrm{d}v}h(v \mid \boldsymbol{d})|_{v=y}$. Then, for all $y \in \mathcal{Y}$, $\boldsymbol{d} \in \mathcal{D}$, we have $NLL\left(F_Z(\sum_m w_m h_m(\cdot \mid \boldsymbol{d})), y, \boldsymbol{d}\right) \leq \sum w_m NLL\left(F_Z \circ h_m(\cdot \mid \boldsymbol{d}), y, \boldsymbol{d}\right)$.*

Tuning ensemble weights. Ensemble prediction performance can be further improved by tuning the ensemble's weights instead of equal weighting or top-K ensembling, in which the K members with the best validation loss are used to form the ensemble. The ensemble weights can be tuned such that the prediction performance w.r.t. a proper score is optimized on a hold-out data set. Tuning the composition of an ensemble in such a way is related to stacking [5,54]. The hold-out set could be another validation set or a validation fold in a nested cross-validation. Concretely, the optimal weights are obtained by solving

$$\begin{aligned} &\min_{w \geq 0} \sum_{i=1}^{n} s(\bar{F}_{M,i}^{w}, y_i), \\ &\text{s.t.} \sum_m w_m = 1, \end{aligned} \tag{4}$$

for some proper score s (see Appendix C) and predictions from the weighted ensemble $\bar{F}_{M,i}^{w}$. Tuning the ensemble weights may be expected to improve overall ensemble performance especially for models which are difficult to train and easily get stuck in local optima, because their contribution to the overall prediction is discounted via lower weights.

3.1. Minimax optimality of binary transformation ensembles

Commonly practitioners aim for the best performing model. However, in many applications optimal worst-case prediction errors are sought [*e.g.,* in robust statistics, [26]]. For instance, in forecasting extreme weather events practitioners may be willing to sacrifice average prediction performance to mitigate worst-case prediction errors [19]. There is a strong similarity between transformation ensembles in Eq. (3) and quasi-arithmetic pooling in Eq. (1), for which minimax properties were shown for nominal outcomes [39]. While quasi-arithmetic pooling is defined for densities of nominal outcomes, transformation ensembles act on the CDF of outcomes with an ordered sample space. However, for the special case of binary outcomes both aggregation methods coincide. Hence, for binary outcomes transformation ensembles are guaranteed to minimize worst-case prediction error in terms of NLL. The result follows from Theorem 4.1 in Neyman and Roughgarden [39].

Corollary 1 (Minimax optimality of transformation ensembles). *Let $p_1, \dots, p_M$ be predicted probabilities for success in a binary outcome, i.e., $p_m = \mathbb{P}_m(Y = 1 \mid D)$ and $w_1, \dots, w_M$ be non-negative weights summing to one. Then $\bar{p}_M^{TRF} = expit(\sum_m w_m logit(p_m))$ minimizes*

$$\max_y NLL(p, y) - \sum_{m=1}^{M} w_m NLL(p_m, y).$$

In words, when working with binary outcomes, using transformation ensembles with $F_Z = expit$ minimizes the worst-case prediction error measured in terms of negative log-likelihood. For high-stakes applications in which worst case risk ought to be minimized, transformation ensembles with $F_Z = expit$ can be a good choice. Note that the result is independent of the type of ensemble members, *i.e.,* for minimax optimality to hold, the members do not need to be transformation models. The proof of Cor. 1 is given in Appendix D. There, we also prove minimax optimality of linear pooling in terms of RPS and Brier score.

4. Case study with a continuous outcome: the `movies` dataset

We juxtapose classical (linear and log-linear) and transformation ensembles in terms of their key features, interpretability, prediction performance and uncertainty quantification. We illustrate these based on a publicly available dataset of movie ratings, described below.

Movies data. The `movies` dataset [31] contains information on movies released before 2017 and contains the continuous tabular features "revenue" and "popularity" and textual movie reviews. We use four non-overlapping subsets of the movie data, holding 2194, 244, 128, and 642 movies, for fitting the model, early stopping, tuning ensemble weights, and evaluating the performance on test data, respectively. We pre-process popularity using $\log(1+x)$ due to its skewness, and model the conditional distribution of movie revenues exceeding \$10,000. Further details on pre-processing are given in Appendix B.

4.1. Interpretability

We build a semi-structured model for predicting movie revenue based on popularity and the textual review. We use the deep transformation model introduced in Example 1 (Section 2.2) with the continuous outcome movie revenue (Y), the continuous popularity score as tabular feature (X), and the textual review representation ($\boldsymbol{T}$) as unstructured input: $F(y \mid x, \boldsymbol{t}) := F_Z\left(\boldsymbol{a}_{\text{Bs},10}(\log(1+y)) + x\beta + \eta(\boldsymbol{t})\right)$. Because we choose a standard logistic latent distribution F_Z, the additive shift terms can be interpreted as log odds-ratios. For instance, when dropping one sentence from an original review $\boldsymbol{t}$ to obtain an altered review $\boldsymbol{t}'$ (and keeping popularity constant), then $e^{\eta(\boldsymbol{t}')-\eta(\boldsymbol{t})}$ is the factor by which the odds for a higher log-revenue changes when observing $\boldsymbol{t}'$ instead of using $\boldsymbol{t}$.

Results. We now fit $M = 5$ deep transformation models to the movies data to build classical and transformation ensembles. In Fig. 3, we show the results. Panel A shows a visualization of the marginal relationship between log-popularity and log-revenue, the outcome. For two movies in the test data (color coded with purple and green) panel B shows the predicted PDFs of the five members along with the corresponding transformation ensemble and classical ensemble. The transformation ensemble is constructed by averaging the transformation function of the five members resulting in $\bar{h}_M(y \mid x, \boldsymbol{t}) = \boldsymbol{a}_{\text{Bs},10}(\log(1+y))^\top \bar{\boldsymbol{\vartheta}}_M + x\bar{\beta}_M + \bar{\eta}(\boldsymbol{t})$.

Interpretability of the effect of popularity and textual reviews on log-revenue directly carries over from the members to the transformation ensemble, because the additive structure of the transformation function is preserved under transformation ensembling (Proposition 1). This interpretability is lost in the classical ensemble because F_Z is nonlinear. However, classical ensembles may achieve a better prediction performance due to their potential to leave the model class [1] of the underlying transformation model (see Fig. 2), which is investigated next.

4.2. Prediction performance

The classical linear ensemble is not guaranteed to be in the same model class as its members. It is an open question whether this increased flexibility is necessary for the improvement in prediction performance which ensembling guarantees. If the increased model complexity was necessary, transformation ensembles would be unable to compete with the classical ensemble, because they remain in the same model class as their members. Hence, transformation ensembles provide a direct way to resolve this open question, which we illustrate now on the movies data. In Section 5.3 we present empirical evidence that the increased complexity is not necessary in typical applications (see *e.g.,* Figs. 4 and 6). All types of ensembles show considerable improvement over the average performance of their members, while all types of ensemble perform roughly on par. In addition, while increasing model complexity (*i.e.,* increasing the number of parameters) increases prediction performance, ensembling improves this performance for all model complexities under study (see Section 5.3).

Results. Turning back to the `movies` example, Fig. 3(D) shows the negative log-density for the two picked-out test movies. The NLL contribution differs for each of the five members, while the classical linear and transformation ensemble are quite close at the observed log-revenue. Averaged over all test movies, the transformation ensemble achieved an NLL of 0.89 (95% bootstrap confidence interval (BCI): $[0.83, 0.95]$) for equal weights and 0.85 (95% BCI: $[0.79, 0.91]$) for tuned weights. The classical ensemble performed similarly to the unweighted transformation ensemble with an NLL of 0.89 (95% BCI: $[0.83, 0.95]$), and all ensemble models performed better than the average of all single model NLLs, 0.91 (95% BCI: $[0.85, 0.97]$).

Even when applying classical ensembles to transformation model members, we can use the transformation ensemble approach to judge whether the ensemble deviates from the structural assumptions of its members. To do so, we transform the M predicted CDFs to the scale of the empirical transformation function, $\{h_m(y_i \mid \mathbf{X}_i) = {F_Z}^{-1} \circ F_m(y_i \mid \mathbf{X}_i)\}_{i=1}^n$, and plot those against the likewise transformed ensemble predictions $\{{F_Z}^{-1} \circ \bar{F}_M(y_i \mid \mathbf{X}_i)\}_{i=1}^n$, where $\bar{F}_M(y_i \mid \mathbf{X}_i)$ is for example the classical linear ensemble (see Fig. 3(C). Observations close to the diagonal indicate that the ensemble does not deviate far from the model class of its members. In this example, only few observations deviate from the diagonal, indicating that the ensemble deviates little from the members' model class.

4.3. Uncertainty quantification

We focus on algorithmic and aleatoric uncertainty, as defined next. Algorithmic uncertainty results from repeating the stochastic fitting procedure with random parameter initializations and the non-convex loss-landscape of deep neural networks while keeping the data fixed, which leads to the heterogeneity between the M ensemble members (see

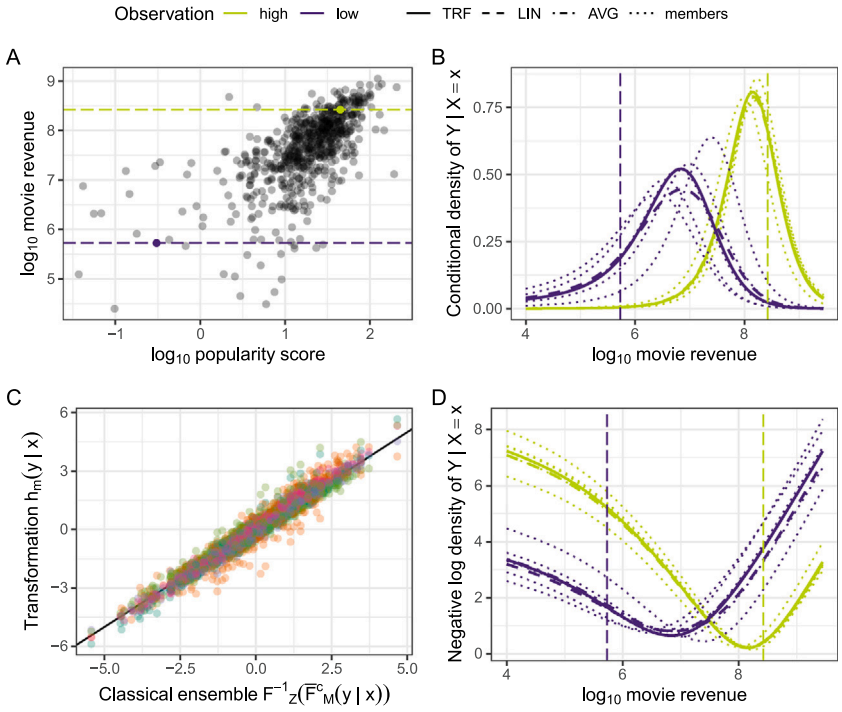


Fig. 3. Transformation (TRF) and classical ensembles (LIN, [34]) applied to the `movies` dataset. Two test movies are picked for detailed visualization and are labeled as "high" and "low". A: Marginal scatterplot of revenue against popularity score. The horizontal dashed lines mark the observed values of the revenue of these two picked movies. B: Predicted PDFs of members (dashed lines) and ensembles (solid lines) for the two chosen movies. C: Visual comparison of how far a classical linear ensemble deviates from the model class of its members on the scale of F_Z^{-1}, in this case the logit scale. The five colors indicate the five ensemble members. D: Negative log-density of members and ensembles of the two picked movies when assuming different values for the observed revenue. NLL contributions of the really observed revenues are indicated by the horizontal dashed lines. AVG denotes the average negative log-density of the five members and lies very close to TRF, making them hard to distinguish visually.

for example Fig. 3(B)–(C). Algorithmic uncertainty, as a form of model-uncertainty, can be subsumed under epistemic uncertainty. Aleatoric uncertainty refers to the variation inherent to the data generating process. Conditional on modeling assumptions, each ensemble member captures aleatoric uncertainty in the predicted conditional outcome distribution (see Fig. 3(B).

Throughout this manuscript, transformation and classical ensembles are both based on the same members, but differ in the way the members are combined, which results in slightly different ensemble distributions. To assess and compare how well the overall uncertainty is captured in the different ensemble approaches, we compare the NLL and the calibration on the test set (see Section 5.3), revealing very similar performance. However, the transformation ensemble has the advantage, that it allows to assess the uncertainty of interpretable model parts. For instance, if the transformation function includes a linear predictor $\mathbf{X}^\top \beta_m$, $m = 1, \dots, M$, the ensemble estimate $\bar{\beta}_M$ together with the individual estimates conveys the algorithmic uncertainty. Asymptotic or bootstrap confidence intervals can be computed for each component in $\bar{\beta}_M$ (see Fig. 5). However, refitting deep ensembles is computationally very expensive, rendering experiments for empirically investigating coverage of these intervals infeasible.

5. Case studies with discrete outcomes

We evaluate transformation ensembles in terms of prediction performance and calibration on several publicly available, semi-structured data sets with binary and ordinal outcomes. Results for a continuous outcome have been shown in Section 4. We compare transformation ensembles, that preserve structure and interpretability of its members, to state-of-the-art ensembling methods (linear, log-linear ensembles, see Section 2.1) where the structure of the members is not preserved. In the following, we describe the different data sets and models used in our experiments.

5.1. Data sets

Melanoma. The publicly available melanoma data set [27] contains skin lesion color images of dimension $128 \times 128 \times 3$ along with age information of 33,058 patients. The response is binary and highly imbalanced (98.23% of all skin lesions are benign and 1.77% malignant).

UTKFace. The publicly available UTKFace data set contains facial images from people of various age groups. Here, age is treated as an ordinal outcome using seven categories: 0–3 ($n = 1'894$), 4–12 ($n = 1'519$), 13–19 ($n = 1'180$), 20–30 ($n = 8'068$), 31–45 ($n = 5'433$), 46–61 ($n = 3'216$) and > 61 ($n = 2'395$) [11]. In addition, the data set contains sex (female, male) as a feature. To compare against the results reported in Kook et al. [32], we use the same cropped version of the images reported therein. We simulate 10 additional covariates using the same simulation scheme as in Kook et al. [32] with effect sizes $\pm \log 1.2$ for $X_{\{2,3\}}$, $\pm \log 1.5$ for $X_{\{5,6\}}$, and 0 for the remaining covariates.

We present results for the MNIST data set in Appendix E.

5.2. Models

For all data sets, we train several (ordinal) neural network transformation models (see Section 2.2) of varying intrinsic interpretability and flexibility. The data sets feature binary and ordinal outcomes, all of which can be handled by deep transformation models. All models use $F_Z = expit$, *i.e.,* a logistic latent distribution.

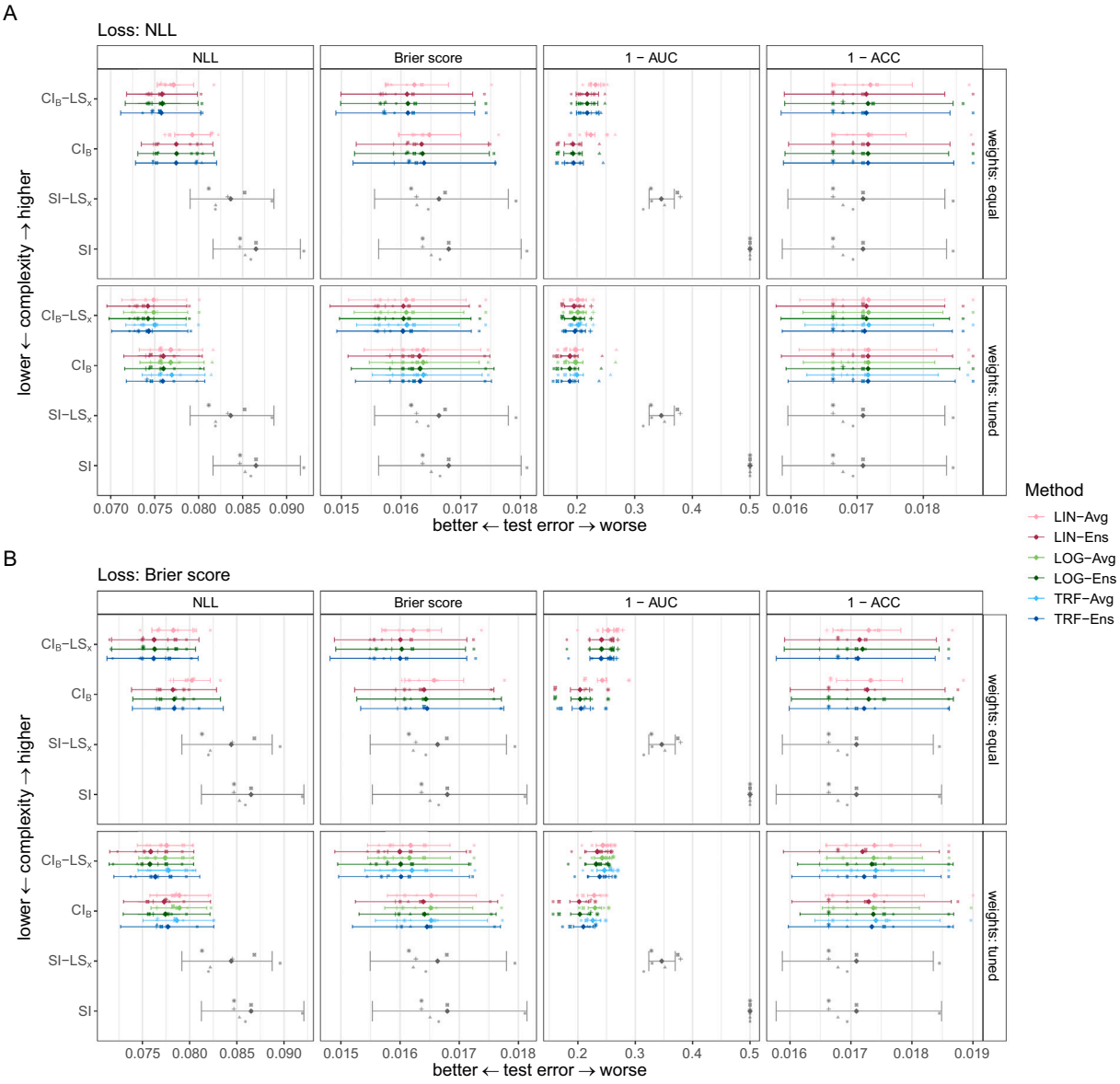


Fig. 4. Performance estimates on the melanoma data set. The classical linear (LIN-Ens, [34]), classical log-linear (LOG-Ens), and transformation (TRF-Ens) ensemble test error are shown for negative log-likelihood (NLL, ↓), Brier score (↓), discrimination error (1 − AUC, ↓) and classification error (1 − ACC, ↓). The average ensemble test error and 95% bootstrap confidence intervals are depicted for six random splits (indicated by different symbols) of the data. In the upper panels ensemble members are weighted equally and in the lower panels weights are tuned to minimize validation loss. In case of equal weights, the average coincides for all ensemble types (LIN-Avg). Models are fitted by minimizing NLL (A) or Brier score (B). Note the different scales for A and B. Exact numbers are reported in Tab. E2. Model abbreviations are defined in Table 1.

The degree of intrinsic interpretability can be controlled by parametrizing the transformation function in different ways (see Table 1). We follow the model nomenclature in Kook et al. [32], where transformation functions consist of an intercept (SI: simple intercept, *i.e.,* intercepts do not depend on input data, CI: complex intercept, intercepts depend on input data) and potentially shift terms (LS: linear shift, additive linear predictor of tabular features, CS: complex shift, additive but flexible predictor depending on either tabular or image data). A subscript indicates which part of the model depends on which input modality. Note that for models with binary outcomes, the CI_B and SI-CS_B are equivalent, unless the complex shift term is restricted to have mean of zero. Therefore, we only fit the CI_B version of these models which are reparametrizations of classical image convolutional networks with softmax last-layer activation.

Training. All models were fitted by minimizing NLL or RPS, both of which are proper scores (Appendix C). In contrast to NLL, RPS is a global score which is bounded and explicitly takes the natural order of the outcome into account. For binary responses, RPS reduces to the Brier score. We apply all ensemble methods to five instances of the same model for six random splits of each data set. Training procedures and model architectures are described in Appendix B in more detail.

Evaluation. For all data sets, we evaluate the prediction performance via proper scores (NLL, RPS or Brier score, Section 2) and discriminatory performance (accuracy, AUC and Cohen's quadratic weighted kappa). We investigate uncertainty quantification of the ensembles using calibration plots.

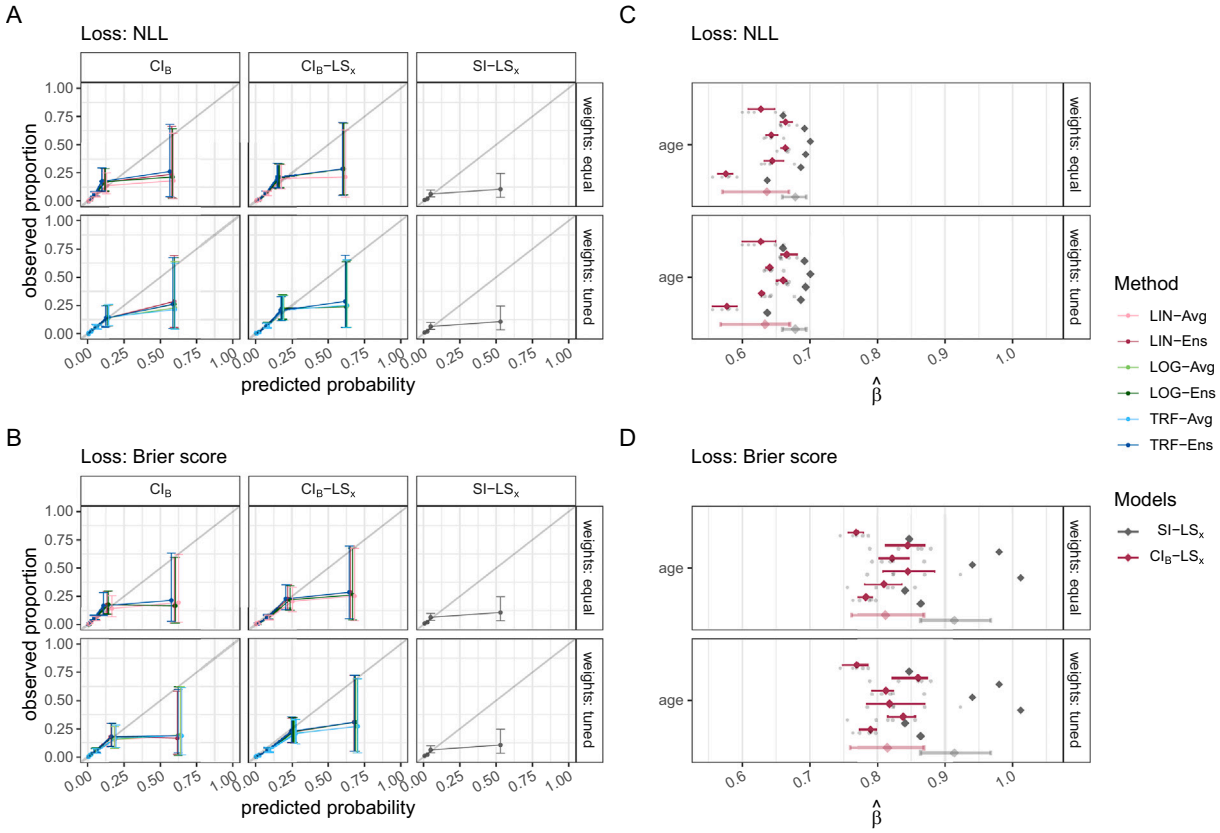


Fig. 5. Calibration plots (A and B) and coefficient estimates (C and D) for different models fitted on the melanoma data set. Calibration with 95% confidence intervals averaged across splits are depicted for the classical linear (LIN-Ens, [34]), classical log-linear (LOG-Ens), and transformation (TRF-Ens) ensemble in A and B. The predicted probabilities for a malign lesion are split at four empirical quantiles (0.5, 0.9, 0.99, 0.999) to compute the proportion of a malign lesion in the five resulting bins. For the SI-LS$_x$, the last two bins are merged. Panels C and D show log odds-ratios ($\hat{\beta}$) and 95% bootstrap confidence intervals for the standardized age predictor in the models SI-LS$_x$ and CI$_B$-LS$_x$ for each of the six random splits. The average log odds-ratio across splits is shown as transparent diamond along with a 95% confidence interval. Individual log odds-ratios of the five ensemble members in each split are shown as transparent dots for the model CI$_B$-LS$_x$. In the upper panels ensemble members are weighted equally and in the lower panels weights are tuned to minimize validation loss. Model abbreviations are defined in Table 1.

5.3. Results and discussion

We apply linear, log-linear, and transformation ensembling to data sets with a binary, and ordered outcome, respectively (see Section 5.1). We refer to the three ensemble types as LIN-Ens [34], LOG-Ens, and TRF-Ens in all figures, respectively.[1] The focus of the comparison is on the performance difference between ensemble types, but also between the different models, which vary in their degree of flexibility and interpretability (see Table 1). Average ensemble test performance and bootstrap confidence intervals based on six random splits of the data are shown for all models. Besides prediction performance, we discuss interpretability and calibration of the different models and ensembles.

Melanoma. Four models of different flexibility and interpretability are fitted (see Table 1) with the melanoma data in order to predict the conditional probabilities of the unbalanced binary outcome benign or malignant lesion given a person's age and an image of the skin lesion.

We now see empirically that all ensemble methods (LIN-Ens, LOG-Ens, TRF-Ens) result in a higher test performance w.r.t. the two proper scores NLL (as shown in Proposition 2) and RPS compared to their members' average (Avg, see Fig. 4). When additionally tuning the ensemble weights on the validation set, test prediction and discrimination performance improve further. Test performance indicates for all ensemble methods that both input modalities, a person's age and appearance of the lesion, aid in predicting the risk of a malignant *vs.* a benign skin lesion (see Fig. 4 or Tab. E2). While the image data (CI$_B$) seems to be most important for prediction, including age (CI$_B$-LS$_x$) further improves prediction performance (NLL, Brier score). The unconditional model (SI) achieves an NLL and Brier score close to that of the model based on age alone (SI-LS$_x$). However, SI-LS$_x$ lacks discriminatory ability (AUC), which suggests that the seemingly high performance of the unconditional model is due to the highly imbalanced outcome. Performance of the individual ensemble members and performance relative to the SI-LS$_x$ model are shown in Appendix E.

For the best performing CI$_B$-LS$_x$ model all three ensemble methods improve compared to the average performance of the individual models and result in similarly low test error (NLL, Brier score). The positive effect of optimizing the ensemble weights seems to be more pronounced when there is more variation in the individual members' performance (note the larger benefit for the CI$_B$ model than for the CI$_B$-LS$_x$ model in Fig. E3). A reduction in between-split variation is observed, when tuning the ensemble weights. Whether NLL or Brier score is optimized during training influences test performance only slightly for this data set. Especially for AUC, optimizing NLL instead of Brier score yields slightly better results (compare Fig. 4(A) and B).

Calibrated predictions are hard to achieve when the outcome is highly imbalanced. For predicted probabilities below 0.2, all models seem to be well calibrated (Fig. 5(A) and B). However, the SI-LS$_x$ model over-predicts the probability for the rare outcome (malign lesion). Note

[1] All code for reproducing the results is available on GitHub https://github.com/LucasKook/interpretable-deep-ensembles.

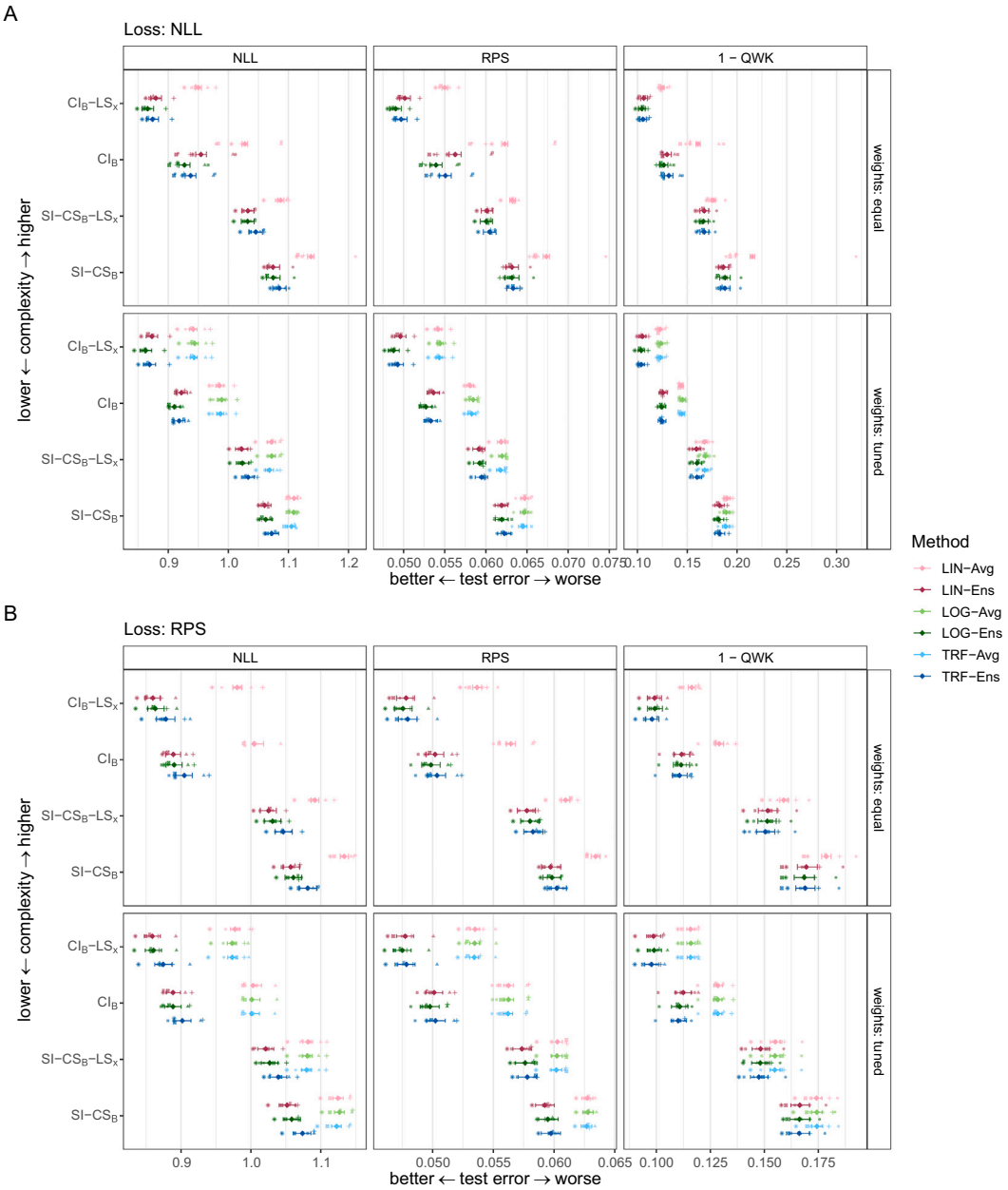


Fig. 6. Performance estimates on the UTKFace data set. The classical linear (LIN-Ens, [34]), classical log-linear (LOG-Ens), and transformation (TRF-Ens) ensemble test error is shown for negative log-likelihood (NLL, ↓), ranked probability score (RPS, ↓) and discrimination error measured by Cohen's quadratic weighted kappa (1 – QWK, ↓). The average ensemble test error and 95% bootstrap confidence intervals are depicted for six random splits (indicated by different symbols) of the data. In the upper panels ensemble members are equally weighted for constructing the ensemble and in the lower panels weights are tuned to minimize validation loss. In case of equal weights, the average coincides for all ensemble types (LIN-Avg). Models are fitted by minimizing NLL (A) or RPS (B). Note the different scales for A and B. The results for the SI and SI-LS$_x$ models are shown in Appendix E. Exact numbers are reported in Tab. E3. Model abbreviations are defined in Table 1.

that the last bin includes only 0.1% of observations and the model never predicts probabilities of a malign lesion larger than 0.13. When including the image data, calibration improves somewhat and uncertainty increases. Again, there is no pronounced difference between equal and tuned weights, ensemble types, and ensembles and individual members, or NLL and Brier score loss.

Lastly, since age is included as a tabular predictor, individual models and transformation ensembles thereof produce directly interpretable estimates for the log odds of a malign *vs.* a benign lesion upon increasing age by one standard deviation (Fig. 5(C) and D). In all models, increasing age is associated with a higher risk of the lesion being malign. In transformation ensembles, the pooled estimate is simply the (weighted) average of the members' estimates. Note that the log odds-ratios obtained by minimizing NLL and RPS agree in direction, but differ slightly in magnitude. In particular, the maximum RPS solution in SI-LS$_x$ is more uncertain than the maximum likelihood solution (grey dots in Fig. 5(C) *vs.* D).

UTKFace. We fit six models of different flexibility and interpretability (see Table 1) to the UTKFace data in order to predict the conditional distribution of age categories given a person's face and/or sex and simulated tabular data. Fig. 6 shows the average ensemble test error with bootstrap confidence intervals for four more complex models (CI$_B$-LS$_x$, CI$_B$, SI-CS$_B$-LS$_x$, SIcsb, see Table 1) and each ensembling method.

Table 1
Overview of data sets, outcomes, and models fitted thereon. The melanoma data set comes with a binary outcome for which we can fit a semi-structured model (CI_B-LS_x), two models using only one of the two modalities, and the unconditional model (SI). For UTKFace, age is split into ordered categories and several models of varying complexity are fitted. All models use $F_Z = expit$ as the target distribution and hence all components of the transformation function are interpretable on the log-odds scale. CI: Complex intercept. CS: Complex shift. SI: Simple intercept. LS: Linear shift. A subscript indicates the input data for a model term, *e.g.,* CS_B is a neural network with image input modelling a complex shift effect. We only aggregate models with transformation functions that share the same structure to preserve interpretability.

Data set	Outcome	Type	Model name	Transformation function
Melanoma	benign/malign	Binary	CI_B-LS_x	$\vartheta(B) - \mathbf{X}^\top\beta$
			CI_B	$\vartheta(B)$
			SI-LS_x	$\vartheta - \mathbf{X}^\top\beta$
			SI	ϑ
UTKFace	age groups	Ordinal	CI_B-LS_x	$\vartheta_k(B) - \mathbf{X}^\top\beta$
			SI-CS_B-LS_x	$\vartheta_k - \eta(B) - \mathbf{X}^\top\beta$
			CI_B	$\vartheta_k(B)$
			SI-CS_B	$\vartheta_k - \eta(B)$
			SI-LS_x	$\vartheta_k - \mathbf{X}^\top\beta$
			SI	ϑ_k

Test performance of the two simplest models (SI, SI-LS_x) can be found in the Appendix E along with the estimated coefficients for age and the ten simulated tabular predictors. Performance of the individual ensemble members and performance relative to the SI-LS_x model are also shown in Appendix E.

Similar to the results found in Kook et al. [32] the most flexible model including both image and tabular data (CI_B-LS_x) performs best across both proper scores and discrimination metrics. Assuming proportional odds for the image term does not seem to be appropriate, given the improved prediction performance when loosening this assumption (CI_B-LS_x *vs.* SI-CS_B-LS_x and CI_B *vs.* SIcsb). All models including the image data perform considerably better than the benchmark model including only tabular data (SI-LS_x, see Fig. E5).

When comparing the different ensembling methods we again observe comparable performance of the transformation ensemble to that of the classical ensembles for most test metrics. This difference is even more pronounced when ensemble weights are tuned to minimize validation loss (lower panels of Fig. 6). As observed in the other data sets, tuning the ensemble weights reduces the variability in performance. As expected, models fitted by minimizing the RPS result in a lower test RPS. However, the NLL of these models is also on par with or even better than when optimizing the NLL. This provides some empirical evidence that opti-

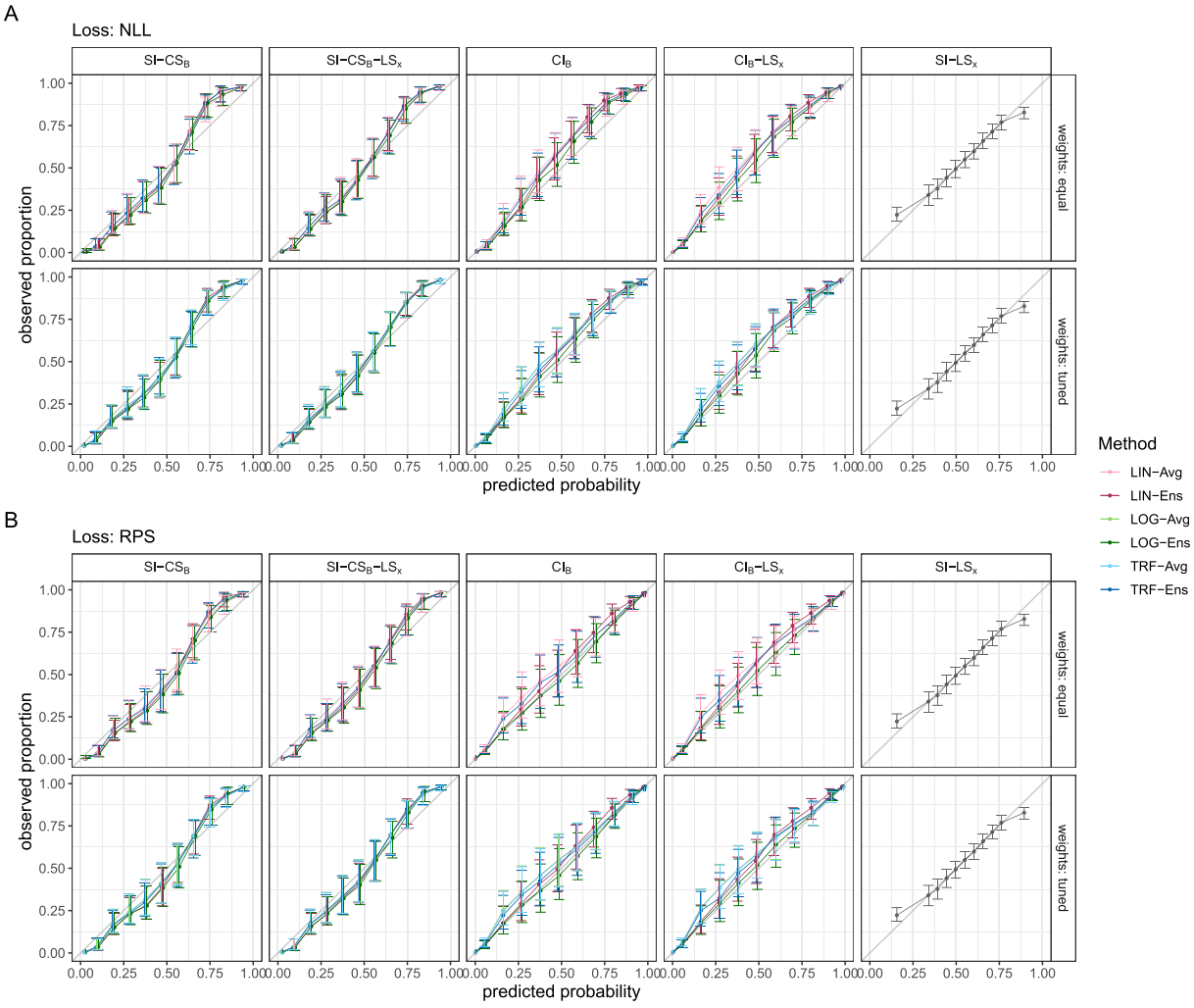


Fig. 7. Calibration plots for different models fitted on the UTKFace data set. The average calibration across six random splits and 95% confidence intervals are depicted of the classical linear (LIN, [34]), classical log-linear (LOG), and transformation (TRF) ensemble. The predicted probabilities are split at the 0.5, 0.95 quantiles and nine equidistant cut points in-between to calculate the observed event rate in each bin. In the upper panels ensemble members are equally weighted for constructing the ensemble and in the lower panels weights are tuned to minimize validation loss. Models are fitted by minimizing NLL (A) or RPS (B). Model abbreviations are defined in Table 1.

mizing the RPS may lead to more stable training for ordered outcomes, because it is bounded and global [17]. Further, transformation ensembles consistently score the best in terms of QWK across all models, losses and tuned/non-tuned weights (see also Tab. E3 in the appendix).

On the UTKFace data, models are fairly calibrated after training (Fig. 7). The SI-LS$_x$ model, including only tabular data, over-predicts large probabilities. Upon including image data as complex intercepts, calibration again improves notably (CI$_B$, CI$_B$-LS$_x$), whereas under-prediction can be observed when modelling image contributions as complex shifts (SI-CS$_B$, SI-CS$_B$-LS$_x$). This can again be interpreted as evidence that assuming proportional odds is unreasonable for this data set. CI$_B$ and CI$_B$-LS$_x$ models are better calibrated when fitted with the RPS loss. Again, no pronounced difference could be observed between the two weighting schemes, or between ensembles and individual models. However, the log-linear ensemble produces the best-calibrated predictions in the CI$_B$ and CI$_B$-LS$_x$ models.

For the ten simulated tabular predictors and sex the individual models with linear shift terms and the corresponding transformation ensemble yield directly interpretable log odds-ratios. In Fig. E7, estimated log odds-ratios are depicted together with bootstrap confidence intervals. The CI$_B$-LS$_x$ models' estimates are close to the estimates of the SI-LS$_x$ models, whereas the estimates of the SI-CS$_B$-LS$_x$ model experience some shrinkage. The estimate for sex (female) changes sign upon inclusion of the image data, potentially due to collinearity between learned image features and the tabular indicator variable. Again, the weighted average of the ensemble members' estimated coefficients is in fact the estimate in the transformation ensemble, which is not the case for any other type of ensemble discussed in this paper.

6. Summary and outlook

Transformation ensembles bridge the gap between two common goals, prediction performance and interpretability. Not only are they guaranteed to score better than their ensemble members do on average, but in addition they preserve model structure and thus possess the same intrinsic interpretability as their members. As a consequence, transformation ensembles allow one to directly assess algorithmic uncertainty in the intrinsically interpretable model parameters. On multiple data sets we demonstrate that transformation ensembles improve both probabilistic and discriminatory performance measures. Transformation ensembles perform on par with classical ensembling approaches on all data sets. Thus transformation ensembles present a viable alternative to classical ensembling in terms of prediction and discrimination.

Preserving the intrinsic interpretability of its members is the most crucial benefit of transformation ensembles. Practitioners in fields where decisions are commonly based on multimodal and semi-structured data, such as medicine, require transparent, well-performing and intrinsically interpretable models [44]. As our results suggest, the increased flexibility of the model class when using classical ensemble techniques may often not be necessary. Instead, the more interpretable transformation ensemble performs at least on par. In addition, transformation ensembles simply pool interpretable model parameters in additive linear predictors using a (weighted) average. This does not only yield transformation ensembles with the same model structure and interpretability, but also allows to assess the algorithmic uncertainty in linear shift terms. This is in line with intuition but lacks theoretical justification when using any other type of ensemble. For complex shift and complex intercept terms which are not intrinsically interpretable, data analysts may still use transformation ensembles and apply *post hoc* explainability methods to those model components [[38] Ch. 6].

Transparency requires clear communication of data and model uncertainty. Transformation ensembles estimate algorithmic uncertainty both in function space (transformation function, CDF) and parameter space (parameters of additive linear predictors). We use ensemble calibration to judge the quality of both aleatoric and algorithmic uncertainty simultaneously. In terms of calibration, no ensemble type led to a pronounced improvement over individual models' average. For both data sets including tabular data (melanoma, UTKFace), including the images alongside tabular data improved calibration. In clinical prediction modelling, where calibrated predictions are of high importance, it may thus be advantageous to model image contributions or to aggregate re-calibrated [[48] Ch. 15.3.5] versions of the models.

Optimizing the ensemble weights on a hold out set, is a simple way to reduce variability between random splits of the data. If only little data is available for training, a cross validation scheme may yield similar benefits when averaging the weights over the cross validation folds.

In this article, we mainly focused on interpretability, prediction and calibration. Apart from benefits in prediction performance and uncertainty quantification, deep ensembles have been found to be robust when evaluated out of distribution, *e.g.,* under distributional shifts [for a discussion, see [1]]. We leave for future research how transformation ensembles compare to classical linear ensembling in terms of robustness towards distributional shifts and related issues in epistemic uncertainty estimation. The focus of this work is on supervised learning and we did not consider (transformation) ensembles for unsupervised learning tasks, such as image segmentation, and leave investigation of transformation ensembles for such tasks for future work. Further theoretical directions include whether minimax optimality of transformation ensembles for binary outcomes extends to ordered and continuous outcomes.

High prediction performance with calibrated predictions, interpretability, and protection against worst-case prediction errors are fundamental to the endeavor of making deep neural networks and artificial intelligence more trustworthy. This paper sheds light on how to effectively combine probabilistic predictions from deep neural networks and their calibration, while preserving interpretability and, in special cases, guaranteeing worst-case robustness.

CRediT authorship contribution statement

Lucas Kook: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Andrea Götschi:** Writing – review & editing, Writing – original draft, Software, Formal analysis. **Philipp F.M. Baumann:** Formal analysis, Conceptualization. **Torsten Hothorn:** Writing – review & editing, Supervision, Investigation. **Beate Sick:** Writing – review & editing, Writing – original draft, Supervision, Resources, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We thank Jeffrey Adams, Oliver Dürr, Lisa Herzog, David Rügamer and Kelly Reeve for their valuable comments on the manuscript. The research of LK and BS was supported by Novartis Research Foundation (FreeNovation 2019) and by the Swiss National Science Foundation (grant no. S-86013-01-01 and S-42344-04-01). TH was supported by the Swiss National Science Foundation (SNF) under the project "A Lego System for Transformation Inference" (grant no. 200021_184603). We further would like to thank all reviewers and the associate editor for their helpful feedback through which this manuscript has improved.

Supplementary data

Supplementary data to this article can be found online at doi:10.1016/j.neucom.2026.133394.

Data availability

All data used are openly available and a link is provided.

References

[1] T. Abe, E.K. Buchanan, G. Pleiss, R. Zemel, J.P. Cunningham, Deep ensembles work, but are they necessary? in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems, vol. 35, Curran Associates, Inc., 2022, pp. 33646–33660.

[2] P.F.M. Baumann, T. Hothorn, D. Rügamer, Deep conditional transformation models, in: Machine Learning and Knowledge Discovery in Databases. Research Track, Springer International Publishing, 2021, pp. 3–18, https://doi.org/10.1007/978-3-030-86523-8_1

[3] C.B. Bell, A characterization of multisample distribution-free statistics, Ann. Math. Stat. 35 (2) (1964) 735–738, https://doi.org/10.1214/aoms/1177703571

[4] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140, https://doi.org/10.1007/BF00058655

[5] L. Breiman, Stacked regressions, Mach. Learn. 24 (1) (1996) 49–64, https://doi.org/10.1007/BF00117832

[6] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32, https://doi.org/10.1023/A:1010933404324

[7] L. Breiman, Statistical modeling: the two cultures, Stat. Sci. 16 (3) (2001) 199–231, https://doi.org/10.1214/ss/1009213726

[8] G.W. Brier, Verification of forecasts expressed in terms of probability, Mon. Weather Rev. 78 (1) (1950) 1–3, https://doi.org/10.1175/1520-0493(1950)078%3C0001:VOFEIT%3E2.0.CO;2

[9] P. Bühlmann, Bagging, boosting and ensemble methods, in: Handbook of Computational Statistics, Springer, 2012, pp. 985–1022, https://doi.org/10.1007/978-3-642-21551-3_33

[10] D.R. Cox, Regression models and Life-Tables, J. R. Stat. Soc. Ser. B 34 (2) (1972) 187–202, https://doi.org/10.1111/j.2517-6161.1972.tb00899.x

[11] A. Das, A. Dantcheva, F. Bremond, Mitigating bias in gender, age and ethnicity classification: a multi-task convolution neural network approach, ECCVW 2018 - Eur. Conf. Comput. Vis. Workshops (2018), https://doi.org/10.1007/978-3-030-11009-3_35

[12] R.T. Farouki, The Bernstein Polynomial basis: a Centennial retrospective, Comput. Aided Geom. Des. 29 (6) (2012) 379–419, https://doi.org/10.1016/j.cagd.2012.03.001

[13] B. Fu, X. He, Y. Liang, T. Deng, H. Li, H. He, M. Jia, D. Fan, F. Wang, Examination of the performance of Asel and Mpvit algorithms for classifying mangrove species of multiple natural reserves of Beibu Gulf, South China, Ecol. Indic. 154 (2023) 110870, https://doi.org/10.1016/j.ecolind.2023.110870

[14] B. Fu, X. Sun, Y. Li, Z. Lao, T. Deng, H. He, W. Sun, G. Zhou, Combination of super-resolution reconstruction and sga-net for marsh vegetation mapping using multi-resolution multispectral and hyperspectral images, Int. J. Digit. Earth 16 (1) (2023) 2724–2761, https://doi.org/10.1080/17538947.2023.2234340

[15] T. Gneiting, A.E. Raftery, Strictly proper scoring rules, prediction, and estimation, J. Am. Stat. Assoc. 102 (477) (2007) 359–378, https://doi.org/10.1198/016214506000001437

[16] T. Gneiting, R. Ranjan, Combining predictive distributions, Electron. J. Stat. 7 (none) (2013) 1747–1782, https://doi.org/10.1214/13-EJS823

[17] T. Gneiting, A.E. Raftery, A.H. Westveld III, T. Goldman, Calibrated probabilistic forecasting using ensemble model output statistics and minimum crps estimation, Mon. Weather Rev. 133 (5) (2005) 1098–1118, https://doi.org/10.1175/MWR2904.1

[18] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.

[19] E.J. Gumbel, Statistics of Extremes, Columbia University Press, 1958.

[20] L.K. Hansen, P. Salamon, Neural network ensembles, IEEE Trans. Pattern Anal. Mach. Intell. 12 (10) (1990) 993–1001, https://doi.org/10.1109/34.58871

[21] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735

[22] T. Hothorn, Transformation boosting machines, Stat. Comput. 30 (1) (2020) 141–152, https://doi.org/10.1007/s11222-019-09870-4

[23] T. Hothorn, A. Zeileis, Predictive distribution modeling using transformation forests, J. Comput. Graph. Stat. 30 (4) (2021) 1181–1196, https://doi.org/10.1080/10618600.2021.1872581

[24] T. Hothorn, T. Kneib, P. Bühlmann, Conditional transformation models, J. R. Stat. Soc. Ser. B Stat. Methodol. 76 (1) (2014) 3–27, https://doi.org/10.1111/rssb.12017

[25] T. Hothorn, L. Möst, P. Bühlmann, Most likely transformations, Scand. J. Stat. 45 (1) (2018) 110–134, https://doi.org/10.1111/sjos.12291

[26] P.J. Huber, Robust Statistics, vol. 523, John Wiley & Sons, 2004.

[27] International Skin Imaging Collaboration, Siim-Isic 2020 Challenge dataset, 2020. https://challenge2020.isic-archive.com/.

[28] H. Ishwaran, U.B. Kogalur, E.H. Blackstone, M.S. Lauer, Random survival forests, Ann. Appl. Stat. 2 (3) (2008) 841–860, https://doi.org/10.1214/08-AOAS169

[29] I. Izonin, R. Tkachenko, K. Yemets, M. Havryliuk, An interpretable ensemble structure with a non-iterative training algorithm to improve the predictive accuracy of healthcare data analysis, Sci. Rep. 14 (1) (2024) 12947, https://doi.org/10.1038/s41598-024-61776-y

[30] C. Ju, A. Bibaut, M. van der Laan, The relative performance of ensemble methods with deep convolutional neural networks for image classification, J. Appl. Stat. 45 (15) (2018) 2800–2818, https://doi.org/10.1080/02664763.2018.1441383

[31] Kaggle, The `movies` dataset, 2017. https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset.

[32] L. Kook, L. Herzog, T. Hothorn, O. Dürr, B. Sick, Deep and interpretable regression models for ordinal outcomes, Pattern Recognition 122 (2022) 108263, https://doi.org/10.1016/j.patcog.2021.108263

[33] L. Kook, C. Kolb, P. Schiele, D. Dold, M. Arpogaus, C. Fritz, P.F.M. Baumann, P. Kopper, T. Pielok, E. Dorigatti, D. Rügamer, How inverse conditional flows can serve as a substitute for distributional regression, in: The 40th Conference on Uncertainty in Artificial Intelligence, 2024, https://openreview.net/forum?id=jd5DhbTsde.

[34] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in: Advances in Neural Information Processing Systems, vol. 30, Curran Associates, Inc., 2017.

[35] T. Lohse, S. Rohrmann, D. Faeh, T. Hothorn, Continuous outcome logistic regression for analyzing body mass index distributions, F1000Research 6 (2017) 1933, https://doi.org/10.12688/f1000research.12934.1

[36] W. Lu, L. Li, Boosting method for nonlinear transformation models with censored survival data, Biostatistics 9 (4) (2008) 658–667, https://doi.org/10.1093/biostatistics/kxn005

[37] A.C. McLain, S.K. Ghosh, Efficient sieve maximum likelihood estimation of time-transformation models, J. Stat. Theory Pract. 7 (2013) 285–303, https://doi.org/10.1080/15598608.2013.772835

[38] C. Molnar, Interpretable Machine Learning, Lulu.com, 2020.

[39] E. Neyman, T. Roughgarden, From proper scoring rules to max-min optimal forecast aggregation, Oper. Res. 71 (6) (2023) 2175–2195, https://doi.org/10.1287/opre.2022.2414

[40] M. Opitz, H. Possegger, H. Bischof, Efficient model averaging for deep neural networks, in: Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, Springer, 2017, pp. 205–220, https://doi.org/10.1007/978-3-319-54184-6_13

[41] G. Papamakarios, E. Nalisnick, D.J. Rezende, S. Mohamed, B. Lakshminarayanan, Normalizing flows for probabilistic modeling and inference, J. Mach. Learn. Res. 22 (57) (2021) 1–64.

[42] R. Ratcliff, Group reaction time distributions and an analysis of distribution statistics, Psychol. Bull. 86 (3) (1979) 446, https://doi.org/10.1037/0033-2909.86.3.446

[43] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (6) (1958) 386, https://doi.org/10.1037/h0042519

[44] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. 1 (5) (2019) 206–215, https://doi.org/10.1038/s42256-019-0048-x

[45] D. Rügamer, P.F. Baumann, T. Kneib, T. Hothorn, Probabilistic time series forecasts with autoregressive transformation models, Stat. Comput. 33 (2) (2023) 37, https://doi.org/10.1007/s11222-023-10212-8

[46] M. Schmid, T. Hothorn, Boosting additive models using component-wise p-splines, Comput. Stat. Data Anal. 53 (2) (2008) 298–311, https://doi.org/10.1016/j.csda.2008.09.009

[47] B. Sick, T. Hothorn, O. Dürr, Deep transformation models: tackling complex regression problems with neural network based transformation models, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, https://doi.org/10.1109/icpr48806.2021.9413177

[48] E.W. Steyerberg, Clinical Prediction Models, Springer, 2019.

[49] M. Tanveer, A. Rastogi, V. Paliwal, M.A. Ganaie, A.K. Malik, J. Del Ser, C.-T. Lin, Ensemble deep learning in speech signal tasks: a review, Neurocomputing 550 (2023) 126436, https://doi.org/10.1016/j.neucom.2023.126436

[50] G. Tutz, Regression for Categorical Data, vol. 34, Cambridge University Press, 2011, https://doi.org/10.1017/CBO9780511842061

[51] S. Wager, S. Athey, Estimation and inference of heterogeneous treatment effects using random forests, J. Am. Stat. Assoc. 113 (523) (2018) 1228–1242, https://doi.org/10.1080/01621459.2017.1319839

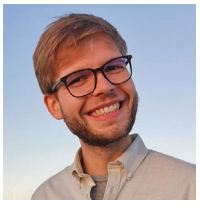
[52] J. Wang, G. Song, A deep spatial-temporal ensemble model for AIR quality prediction, Neurocomputing 314 (2018) 198–206, https://doi.org/10.1016/j.neucom.2018.06.049

[53] A.G. Wilson, P. Izmailov, Bayesian deep learning and a probabilistic perspective of generalization, Adv. Neural Inf. Process. Syst. 33 (2020) 4697–4708, http://arxiv.org/abs/2002.08791.

[54] D.H. Wolpert, Stacked generalization, Neural Netw. 5 (2) (1992) 241–259, https://doi.org/10.1016/S0893-6080(05)80023-1

[55] L. Zhang, G. Yu, D. Xia, J. Wang, Protein–protein interactions prediction based on ensemble deep neural networks, Neurocomputing 324 (2019) 10–19, https://doi.org/10.1016/j.neucom.2018.02.097. Deep Learning for Biological/Clinical Data.

Author biography



Lucas Kook is Assistant Professor of Statistics at the Vienna University of Economics and Business. He received his PhD in Biostatistics at the University of Zurich in 2023. He develops statistical methods for tackling predictive and causal inference problems within complex data structures.

Andrea Goetschi is a biostatistician at the SCQM foundation.

Philipp F.M. Baumann obtained his PhD degree at the ETH Zurich and is now a data scientist at Amazon.

Torsten Hothorn is Professor of Biostatistics at the Epidemiology, Biostatistics and Prevention Institute of the University of Zurich. He received a Ph.D. in Statistics from the University of Dortmund in 2003. His research focus is on developing exible regression models for analyzing biomedical data.

Beate Sick is a Professor at TIDIT where she leads the Probabilistic AI Research Group (PAIR) and is coaffiliated at UZH. She did her PhD in physics at ETHZ. Then she was responsible for the bioinformatics at the DNA Array facility of UNIL and EPFL. Currently, she is working on deep learning approaches for medical research.