Pattern Recognition 122 (2022) 108263

Contents lists available at ScienceDirect

Pattern Recognition

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

# Deep and interpretable regression models for ordinal outcomes

Lucas Kook[a,b,1], Lisa Herzog[a,b,1], Torsten Hothorn[a], Oliver Dürr[c], Beate Sick[a,b,*]

[a] Epidemiology, Biostatistics & Prevention Institute, University of Zurich, Hirschengraben 84, Zurich 8001, Switzerland
[b] Institute of Data Analysis and Process Design, Zurich University of Applied Sciences, Rosenstrasse 3, Winterthur 8400, Switzerland
[c] Institute for Optical Systems, Alfred-Wachtel Strasse 8, Konstanz University of Applied Sciences, Konstanz 78462, Germany

ARTICLE INFO

*Article history:*
Received 15 December 2020
Revised 12 August 2021
Accepted 18 August 2021
Available online 19 August 2021

*Keywords:*
Deep learning
Interpretability
Distributional regression
Ordinal regression
Transformation models

ABSTRACT

Outcomes with a natural order commonly occur in prediction problems and often the available input data are a mixture of complex data like images and tabular predictors. Deep Learning (DL) models are state-of-the-art for image classification tasks but frequently treat ordinal outcomes as unordered and lack interpretability. In contrast, classical ordinal regression models consider the outcome's order and yield interpretable predictor effects but are limited to tabular data. We present ordinal neural network transformation models (ontrams), which unite DL with classical ordinal regression approaches. ontrams are a special case of transformation models and trade off flexibility and interpretability by additively decomposing the transformation function into terms for image and tabular data using jointly trained neural networks. The performance of the most flexible ontram is by definition equivalent to a standard multi-class DL model trained with cross-entropy while being faster in training when facing ordinal outcomes. Lastly, we discuss how to interpret model components for both tabular and image data on two publicly available datasets.



## 1. Introduction

Many classification problems deal with classes that show a natural order. This includes for example patient outcome scores in clinical studies or movie ratings [1]. These ordinal outcome variables may not only depend on interpretable tabular predictors like age or temperature but also on complex input data such as medical images, textual descriptions, or spectra. Depending on the complexity of the input data and the concrete task, different analysis approaches have been established to tackle the ordinal problems.

Ordinal regression as a probabilistic approach has been studied for more than four decades [2]. The goal is to fit an interpretable regression model, which estimates the conditional distribution of an ordinal outcome variable $Y$ based on a set of tabular predictors. The ordinal outcome $Y$ can take values in a set of ordered classes and the tabular predictors are scalar and interpretable like age. Ordinal regression models provide a valid probability distribution instead of a single point estimate for the most likely outcome which is essential to reflect uncertainty in the predictions. Moreover, the estimated model parameters are interpretable as the effect a single predictor has on the outcome given the remaining predictors are held constant. This allows experts to assess whether the model corresponds to their field knowledge and provides the necessary trust for application in critical decision making. However, there is a trade-off between interpretability and model complexity. The higher the complexity of a model, the harder it becomes to directly interpret the individual model parameters.

Deep Learning (DL) approaches have gained huge popularity over the last decade and achieved outstanding performance on complex tasks like image classification and natural language processing [3]. The models take the raw data as input and learn relevant features during the training procedure by transforming the input into a latent representation, which is suitable to solve the problem at hand. This avoids the challenging task of feature engineering, which is necessary when working with statistical models. Yet, unlike statistical models, most DL models have a black box character, which makes it hard to interpret individual model components.

DL models for ordinal models typically do not integrate tabular predictors and yield interpretable effect estimates for tabular and image data at the same time.

This is a major disadvantage for example in fields like medicine which requires multiple data modalities for decision making but also a reliably interpretable model which quantifies the effects of the predictors on the outcome [4].

* Corresponding author at: Epidemiology, Biostatistics & Prevention Institute, University of Zurich, Hirschengraben 84, 8001 Zurich, Switzerland.
*E-mail address:* sick@zhaw.ch (B. Sick).
[1] Authors contributed equally.

https://doi.org/10.1016/j.patcog.2021.108263
0031-3203/

### 1.1. Our contribution

In this work we introduce ordinal neural network transformation models (ontrams), which unite classical ordinal regression with DL approaches while conserving the interpretability of statistical and flexibility of DL models. We use a theoretically sound maximum-likelihood based approach and reparametrize the categorical cross-entropy loss to incorporate the order of the outcome. This guarantees the estimation of a valid probability distribution. By definition, the reparameterized negative log-likelihood (NLL) loss is able to achieve the same prediction performance as a standard DL model trained with cross-entropy loss, but allows a faster training in case of an ordinal outcome. The main advantage of the proposed ontrams is that ontrams provide interpretable effect estimates for the different input data, which is not possible with other DL models.

We view ordinal regression models from a transformation model perspective [5,6]. This change of perspective is useful because it allows a holistic view on regression models, which easily extends beyond the case of ordinal outcomes. In transformation models the problem of estimating a conditional outcome distribution is translated into a problem of estimating the parameters of a monotonically increasing transformation function, which transforms the potentially complex outcome distribution to a simple, predefined distribution $F_Z$ of a continuous variable.

The goal of ontrams is to estimate a flexible outcome distribution based on a set of predictors including images and tabular data while keeping components of the model interpretable. ontrams are able to seamlessly integrate both types of data with varyingly complex interactions between the two, by taking a modular approach to model building. The data analyst can choose the scale on which to interpret image and tabular predictor effects, such as the odds or hazard scale, by specifying the simple distribution function $F_Z$. In addition, the data analyst has full control over the complexity of the individual model components. The discussed ontrams will contain at most three (deep) neural networks for the intercepts in the transformation function, the tabular and the image data. Together with the simple distribution function $F_Z$ the output of these neural networks will be used to evaluate the NLL loss. In the end, the NNs, which control the components of the model, are jointly fitted by standard deep learning algorithms based on stochastic gradient descent. In this work, we feature convolutional neural networks (CNNs) for complex input data like images. However, the high modularity of ontrams enables many more applications such as recurrent neural networks for text-based models.

### 1.2. Organization of this paper

We first give some theoretical background on multi-class classification and ordinal regression. Afterwards, related work is described in Section 2.3 to highlight the contributions of ontrams to the field. We then provide details about ontrams in Section 3. Subsequently, we describe the data sets, experiments, and models we use to study and benchmark ontrams (Section 4). We end this paper with a discussion of our results and juxtaposition of the different approaches in light of model complexity, interpretability, and predictive performance. We present further results in Appendix C and complement our discussion of different loss functions and evaluation metrics in Appendices E and F, respectively. Because most state-of-the-art approaches to ordinal outcomes are classifiers, we particularly highlight the distinction between ordinal classification and the proposed regression approach of ontrams in Appendix G.

## 2. Background

### 2.1. Multi-class classification

In DL approaches ordinal outcomes are frequently modeled in the same way as unordered outcomes using multi-class classification (MCC). That is, softmax is used as the last-layer activation and the loss function is the categorical cross-entropy. The cross-entropy corresponds to the negative log-likelihood and solely the probability assigned to the observed class is entering the loss as $\mathcal{L}_i(h; y_{ki}, \boldsymbol{x}_i) = \mathbb{P}(Y = y_{ki})$, which ignores the outcome's natural order (see also Appendix A).

### 2.2. Ordinal regression models

Ordinal regression aims to characterize the whole conditional distribution of an ordinal outcome variable given its predictors. Consider an ordered outcome variable $Y$ with $K$ possible values $y_1 < y_2 < \ldots < y_K$. The distribution of $Y$ is fully determined by its probability density function (PDF). However, unlike unordered outcomes an ordered outcome possesses a well defined cumulative distribution function (CDF) $F_Y(y_k) := \mathbb{P}(Y \leq y_k)$, which naturally contains the order. The likelihood contribution for an observation $(y_{ki}, \boldsymbol{x}_i)$ is given by the predicted probability for the observed class, which can be written as

$$\mathcal{L}_i = \mathbb{P}(Y = y_{ki}|\boldsymbol{x}_i) = \mathbb{P}(Y \leq y_{ki}|\boldsymbol{x}_i) - \mathbb{P}(Y \leq y_{(k-1)i}|\boldsymbol{x}_i), \tag{1}$$

for $k = 1, \ldots, K$ and $\mathbb{P}(Y \leq y_0) := 0$, $\mathbb{P}(Y \leq y_K) = 1$. Parametrizing the likelihood contributions using the CDF directly enables to incorporate the order of the outcome when formulating regression models for ordinal data (Section 3). It is worth noting that the loss is equivalent to the cross-entropy and merely uses a different parametrization to take the outcome's natural order into account.

Many ordinal regression models assume the existence of an underlying continuous latent variable (an unobserved quantity) $Z$. The ordinal outcome variable $Y$ is understood as a categorized version of $Z$ resulting from incomplete knowledge; we only know the classes in terms of the intervals in which $Z$ lies. Fitting an ordinal regression model based on the latent variable approach aims at finding cut points $h(y_k|\boldsymbol{x})$ at which $Z$ is separated into the assumed classes (see Fig. 1 B). Even if $Z$ can not be interpreted directly, using a latent variable approach has advantages, because the chosen distribution of $Z$ determines the interpretability of the terms in the transformation function (see Section 2.2.1).

Moreover, the latent variable approach enables to understand ordinal regression as a special case of parametric transformation models, which were recently developed in statistics [5] and are applicable to a wide range of outcomes with natural extensions to classical machine learning techniques such as random forests and boosting. Transformation models are able to model highly flexible outcome distributions while simultaneously keeping specific model components interpretable. In transformation models the conditional outcome distribution of $(Y|\boldsymbol{x})$ is modeled by transforming the outcome variable $(Y|\boldsymbol{x})$ to a variable $(Z|\boldsymbol{x})$ with known (simple) CDF $F_Z$, like the Gaussian or logistic distribution. Transformation models in general are thus defined by

$$F_Y(y|\boldsymbol{x}) = F_Z(h(y|\boldsymbol{x})), \tag{2}$$

and all models in our proposed framework of ontrams are of this form.

The goal is then to fit a monotonically increasing transformation function $h$, which maps the observed outcome classes $(y_k|\boldsymbol{x})$ to the conditional cut points

$$h(y_k|\boldsymbol{x}),\ k = 1, \ldots, K-1, \tag{3}$$

of the latent variable $Z$, as illustrated in Fig. 1. In the example in Fig. 1 the outcome can take five classes and the $K-1$ cut

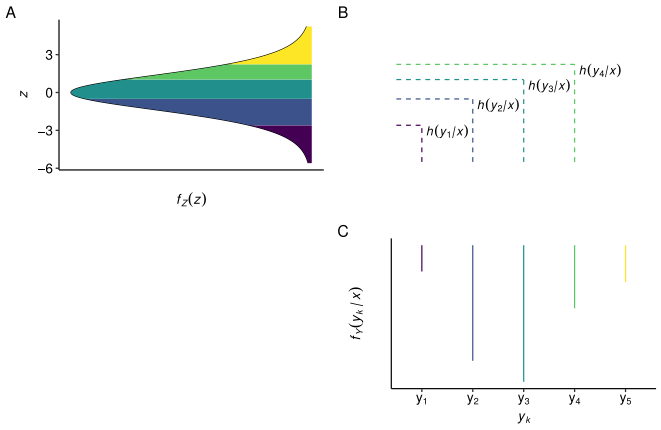


**Fig. 1.** Transformation model likelihoods for a model with ordinal outcome. Panel C shows the conditional density of $Y$ given $\boldsymbol{x}$, which gets mapped onto the density of the latent variable $Z$ (A) via the transformation function $h$ (B). The likelihood contributions are in fact probabilities and given by the area under the density of $Z$ between two consecutive cut points in the transformation function. Note that $h(y_5|\boldsymbol{x}) = +\infty$ does not show on the plot for the transformation function, but is evident from the yellow (upper) area under the density of $Z$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

points $h(y_1|\boldsymbol{x})$, $h(y_2|\boldsymbol{x})$, $h(y_3|\boldsymbol{x})$, and $h(y_4|\boldsymbol{x})$ have to be estimated. The first class of $Y$ on the scale of $Z$ is given by the interval $(-\infty, h(y_1|\boldsymbol{x})]$, the fifth class as $(h(y_4|\boldsymbol{x}), +\infty)$, so often the conventions $h(y_0|\boldsymbol{x}) = -\infty$ and $h(y_K|\boldsymbol{x}) = +\infty$ are used. The likelihood contribution of a given observation $(y_{ki}, \boldsymbol{x}_i)$ can now be derived from the CDF of $Z$ instead of $Y$ and is given by

$$\begin{aligned}\mathcal{L}_i(h; y_{ki}, \boldsymbol{x}_i) &= \mathbb{P}(Y = y_{ki}|\boldsymbol{x}_i) = F_Y(y_{ki}|\boldsymbol{x}_i) - F_Y(y_{(k-1)i}|\boldsymbol{x}_i) \\ &= F_Z(h(y_{ki}|\boldsymbol{x}_i)) - F_Z(h(y_{(k-1)i}|\boldsymbol{x}_i)).\end{aligned} \tag{4}$$

The single likelihood contributions are the heights of the steps in the CDF or equivalently the area under the density of the latent variable $Z$ between two consecutive cut points (*cf.* Fig. 1 B, C). Note that two consecutive cut points enter the likelihood, such that the natural order of the outcome is used to parametrize the likelihood, although the likelihood contribution is given by the probability of the observed class alone. Consequently, minimizing the negative log-likelihood

$$-\ell(h; y_{1:n}, \boldsymbol{x}_{1:n}) = -\sum_{i=1}^{n} \log \mathcal{L}_i(h; y_{ki}, \boldsymbol{x}_i) \tag{5}$$

estimates the conditional outcome distribution of $(Y|\boldsymbol{x})$ by estimating the unknown parameters of the transformation function. Note that in principle this formulation allows us to directly incorporate uncertain observations, for instance, an observation may lie somewhere in $[y_k, y_{k+2}]$, $k \leq K-2$ if a rater is uncertain about the quality of a wine or a patient rates their pain in between two classes.

#### 2.2.1. Interpretability in proportional odds models

Interpretability of a transformation model depends on the choice of the distribution $F_Z$ of the latent variable $Z$ and the transformation function $h$. A summary of common interpretational scales is given in Table 1.

Here, we demonstrate interpretability through the example of a proportional odds model, which is well known in statistics [7]. For the distribution of $Z$ we choose the standard logistic distribution (denoted by $F_L$), whose CDF is given by $F_Z(z) = F_L(z) := (1 + \exp(-z))^{-1}$. The transformation function $h$ is parametrized as

$$h(y_k|\boldsymbol{x}) = \vartheta_k - \sum_{j=1}^{J} \beta_j x_j = \vartheta_k - \boldsymbol{x}^\top \boldsymbol{\beta}, \; j = 1, \ldots, J. \tag{6}$$

**Table 1**
Interpretational scales of shift terms induced by $F_Z$ [7]. Most link functions have been studied in the context of proportional odds model neural networks and a classification loss [8]. More details concerning the interpretational scales are given in Appendix D.

| $F_Z$ | $F_Z^{-1}$ | Symbol | Interpretation of shift terms |
|---|---|---|---|
| Logistic | logit | $F_L$ | log odds-ratio |
| Gompertz | cloglog | $F_{\text{MEV}}$ | log hazard-ratio |
| Gumbel | loglog | $F_{\text{Gumbel}}$ | log hazard-ratio for $Y_r = K + 1 - Y$ |
| Normal | probit | $\Phi$ | not interpretable directly |

A transformation model with such a transformation function is called linear shift model, since a change $\Delta x_j$ in a single predictor $x_j$ causes a linear shift of size $\beta_j \Delta x_j$ in the transformation function.

The popularity of the transformation model with $F_Z = F_L$ is due to the insightful interpretation of the parameter $\beta_j$ as a log odds-ratio

$$\log \text{OR}_{\boldsymbol{x} \to \boldsymbol{x}'} = \log\left(\frac{\text{odds}(Y > y_k|\boldsymbol{x}')}{\text{odds}(Y > y_k|\boldsymbol{x})}\right) = \beta_j, \tag{7}$$

where $\text{odds}(Y > y_k|\boldsymbol{x}) := \mathbb{P}(Y > y_k|\boldsymbol{x})/\mathbb{P}(Y \leq y_k|\boldsymbol{x})$. This is depicted in Fig. 2 for a positive valued $\beta$, where the effect of increasing $x$ by one unit increases the odds for the outcome to belong to a higher class. Specifically, the odds of the outcome being in a higher class than $y_k$ is increased by a factor of $\exp(\beta_j)$, which holds for each $y_k$. However, the resulting conditional distribution changes in a more complex way (Fig. 2 A). Because the effect of $\beta$ is the same for each class boundary these models are referred to as proportional odds models [7]. This corresponds to the shape of the transformation function $h$ being fixed. A more detailed derivation is given in Appendix D.

### 2.3. Related work

We summarize related work in the field of deep ordinal regression and classification and interpretable machine learning.

Prediction models for ordinal outcomes have been studied in machine learning as extensions of different popular methods like Gaussian Processes [9], support vector machines [10], and neural networks [11]. With the advent of deep learning, various approaches have been proposed to tackle classification and regression tasks with ordinal outcomes, which we describe in more detail in the following. Note that we refer to models, which aim to predict a valid entire conditional outcome distribution as ordinal *regression* models, whereas models, which focus on the predicted class label will be referred to as ordinal *classification* models.

For instance, the commonly used multi-class classification model with softmax last layer activation (see Section 2.1) is a regression model (*i.e.,* multinomial regression), whereas most of the state-of-the-art approaches described below are ordinal classifiers. In the following we discuss literature on ordinal classification and literature related to different aspects of our work, *i.e.,* ordinal regression models, transformation models, and interpretability.

*Ordinal classification*

Deep learning approaches to ordinal regression and classification problems range from using an ordinal metric for the evaluation of multi-class classification models to the construction of novel ordinal loss functions and dummy encodings. The earliest approaches made use of the equivalence of an ordinal prediction problem with outcome $Y \in \{y_1 < \ldots < y_K\}$, to the $K-1$ binary classification problems given by $\mathbb{1}(Y \leq y_k)$, $k = 1, \ldots, K$ [12], which is still being used in applications such as age estimation [13].

Cheng et al. [14] devised a cumulative dummy encoding for the ordinal response where for $Y = y_k$ we have $y_i = 1$ if $i \leq k$ and 0

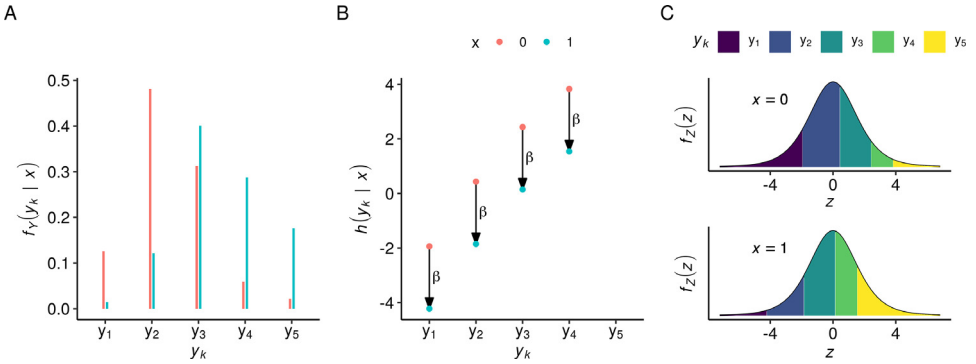


**Fig. 2.** The conditional probability density, transformation function and latent representation of the ordinal outcome $Y$ with 5 classes depending on a single predictor $x$ which is increased by $\Delta x = 1$ from 0 to 1. The density of $(Y|x)$ for $x = 0$ and $x = 1$ is shown in A. The simple linear shift model $h(y_k|\boldsymbol{x}) = \vartheta_k - \beta \cdot x$ imposes a downward shift of the transformation function by $\beta$ when increasing the predictor from $x = 0$ to $x = 1$ (B). The shift in the transformation function translates into a shift in the conditional cut points $h(y_k|\boldsymbol{x})$ under the density of the latent variable $Z$ (C). Shifting the transformation function downwards results in higher probabilities of $Y$ belonging to a higher class. (For a colour version of this figure, the reader is referred to the web version of this article.)

otherwise. Cheng et al. [14] then suggest a sigmoid activation for the last layer of dimension $K$, together with two loss functions (relative entropy and a squared error loss). Similar approaches remain highly popular in application. For instance, [15] extend the approach to rank-consistent ordinal predictions. The problem of rank inconsistency, however, is confined to the $K$-rank and similar approaches and does not appear in ordinal regression models, such as the ones we propose.

Xie and Pun [16] used a similar dummy encoding to train $K - 1$ binary classifiers, which share a common CNN trunk for image feature extraction but possess their own fully connected part. This allows flexible feature extraction while reducing model complexity substantially due to weight sharing. Weight sharing is a natural advantage of models which are trained with an ordinal loss function instead of multiple binary losses, which we describe next. A comparison of ONTRAM against the method described in [16] can be found in Appendix G.

Recently, the focus shifted towards novel ordinal loss functions involving Cohen's kappa, which was first proposed by de La Torre et al. [17] and subsequently used in "proportional odds model (POM) neural networks" [18]. POM neural networks and their extensions to other cumulative link functions in [8] are closely related to ONTRAMs, proposed in this paper, because they constitute a special case in which the class-specific intercepts do not depend on input data (see Section 3). The crucial difference between POM NNs (as proposed in [18]) and ONTRAMs is the quadratic weighted Cohen's kappa (QWK) loss function in POM NNs, compared to a log-likelihood loss in ONTRAMs. Although POM NNs predict a full conditional outcome distribution, their focus lies on optimizing a classification metric (QWK). The idea is to penalize misclassifications that are further away from the observed class stronger than misclassifications that are closer to the observed class. In contrast, in regression approaches, the goal is to predict a valid probability distribution across all classes. We give more detail on and compare our proposed method against the QWK loss in Appendices E and G, respectively. We use QWK-based models as an example to address the general problem arising when comparing classification and regression models, which address different questions and hence optimize distinct target functions.

*Ordinal regression* Lastly, [19] took a probabilistic approach using Gaussian processes with an ordinal likelihood similar to the cumulative probit model (cumulative ordinal model with $F_Z = \Phi$) and a model formulation similar to POM neural networks. We address further related work concerning technical details in Section 3, such as the explicit formulation of constraints in the loss function.

*Transformation models* Deep conditional transformation models have very recently been applied to regression problems with a continuous outcome [6]. Sick et al. [6] parametrized the transformation function as a composition of linear and sigmoid transformations and a flexible basis expansion that ensures monotonicity of the resulting transformation function. The authors applied deep transformation models to a multitude of benchmark data sets with a continuous outcome and demonstrated a performance that was comparable to or better than other state-of-the-art models. However, in one of the benchmark data set the authors treated a truly ordinal outcome as continuous, as done by all the other benchmark models. This is indicative for the lack of deep learning models for ordered categorical regression.

*Interpretability* In general, deep learning models suffer from a lack of interpretability of the predictions they make [3]. In DL models related to image data, interpretability is mostly referred to as highlighting parts of the image that explain the respective prediction. Often, surrogate models are build on top of the black-box model's predictions, which are easier to interpret. One such model is LIME [20]. For problems with an ordinal outcome, [21] comment on the limited interpretability of the ensemble of neural networks in the $K$-rank approach described above and propose to use a mimic learning technique, which combines the ensemble with a more directly interpretable model. In the present work we take a different approach to interpretability rooted in statistical regression models. The interpretability of the effect of individual input features is given by the fitted model parameters in an additive transformation function, which is a common modelling choice for achieving interpretability [4]. We give more detail in Section 2.2.1 and Appendix D.

## 3. Ordinal neural network transformation models

Here, we present ordinal neural network transformation models, which unite cumulative ordinal regression models with deep neural networks and seamlessly integrate complex data like images (B) and/or tabular data ($\boldsymbol{x}$). At the heart of an ONTRAM lies a parametric transformation function $h(y_k|\boldsymbol{x}, \mathrm{B})$, which transforms the ordinal outcome $y_k$ to cut points of a continuous latent variable and controls the interpretability and flexibility of the model (see Fig. 1). The ordering of the outcome is incorporated in the ONTRAM loss function by defining it via the cumulative distribution function

$$\mathrm{NLL} := -\frac{1}{n}\sum_{i=1}^{n} \log\left(F_Z(h(y_{ki}|\boldsymbol{x}_i, \mathrm{B}_i)) - F_Z(h(y_{(k-1)i}|\boldsymbol{x}_i, \mathrm{B}_i))\right). \tag{8}$$

In the following we describe the terms of the parametric transformation function and their interpretability. The parameters of these

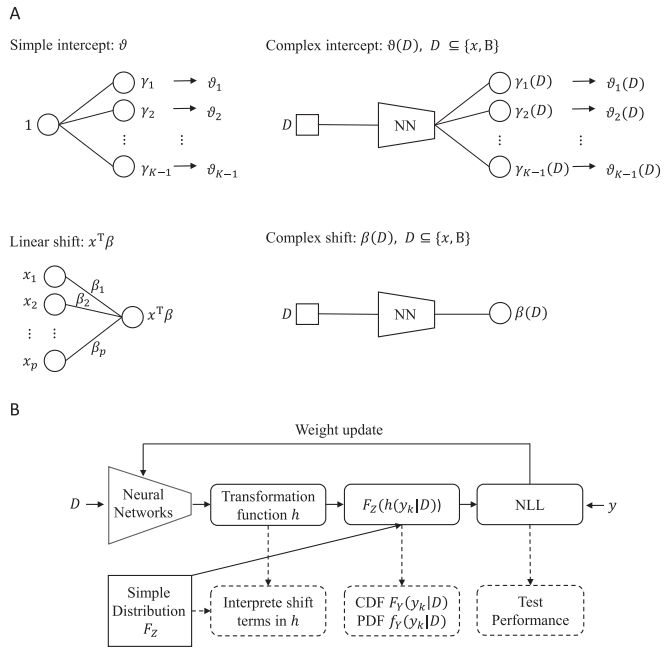


**Fig. 3.** Architecture of ontrams. A: The modular building blocks of the transformation function $h$ are controlled by NNs. Simple intercept and linear shift terms are modeled using a single-layer neural network. Complex intercept and complex shift terms are allowed to depend on the input data in a more complex manner and may be a fully connected or convolutional NN depending on the type of the input data. The input data $D$ can be images $\mathrm{B}$ or tabular predictors $\boldsymbol{x}$. B: The output of the NNs control the additive components of the transformation function $h$. Together with the choice of $F_Z$, $h$ determines the full model, from which the likelihood can be evaluated. During training time (solid lines) the weights of all model components are trained jointly by minimizing the NLL. After training (dashed lines) the shift terms in the transformation function can be interpreted, the conditional outcome distribution can be predicted and the NLL can be evaluated for a given test set.

terms are controlled by NNs, which are jointly fitted in an end-to-end fashion by minimizing the NLL (Fig. 3).

*Modularity* The transformation function $h$ determines the complexity and interpretability of an ontram. The simplest transformation function with only one tabular feature is given by $h(y_k|\boldsymbol{x}) = \vartheta_k - \beta \cdot x$ (see Fig. 2). In general a transformation function is modularly composed of an intercept term, optionally followed by additive shift terms, which depend in a more or less complex manner on different input data and are controlled by NNs (see Fig. 3).

The intercept term controls the shape of the transformation function:

1. Simple intercepts (SI) $\vartheta_k$, $k = 1, \ldots, K-1$ are unconditional, *i.e.,* the shape of the transformation function is independent of the input data. SIs can be modeled as a single layer neural network with $K-1$ output units and linear activation function. The input is given by 1. The outputs are given by $\gamma_1, \ldots, \gamma_{K-1}$ controlling the intercepts (see Fig. 3).
2. Complex intercepts (CI), on the other hand, depend on the input data, which may be tabular data, image data or a combination of both, yielding $\vartheta_k(\boldsymbol{x})$, $\vartheta_k(\mathrm{B})$, or $\vartheta_k(\boldsymbol{x}, \mathrm{B})$, respectively. CIs enable more complex transformation functions, whose shape may vary with the input. Depending on the type of input data, CIs are modeled using a multi-layer fully connected neural network, a convolutional neural network or a combination of both. Analogous to SI terms, the number of output units in the last layer is equal to $K-1$ with linear activation function, yielding $\gamma_1(\boldsymbol{x}, \mathrm{B}), \ldots, \gamma_{K-1}(\boldsymbol{x}, \mathrm{B})$ depending on the input (see upper right panel in Fig. 3).

To ensure that the transformation function is non-decreasing, the outputs $\gamma_1, \ldots, \gamma_{K-1}$ of simple and complex intercept models are transformed before entering the likelihood via

$$\begin{aligned} &\vartheta_k = \vartheta_1 + \textstyle\sum_{i=2}^{k} \exp(\gamma_i), \quad k = 2, \ldots, K-1, \\ &\vartheta_0 = -\infty,\ \vartheta_1 = \gamma_1,\ \vartheta_K = +\infty. \end{aligned} \tag{9}$$

The addition of $\vartheta_0 = -\infty$ and $\vartheta_K = +\infty$ is important for computing the loss as described in Section 2. Enforcing a monotone increasing transformation function via Eq. (9), such that $\vartheta_0 < \vartheta_1 \leq \ldots < \vartheta_K$, has been done similarly in the literature. In what [14] call threshold models, $\gamma_i$ is squared instead of taking the exponential to ensure the intercept function is non-decreasing [8,19]. A different but related approach is to softly penalize the loss for pair-wise rank inconsistencies using a hinge loss [22,23]. Note that the special case $\vartheta_k(\boldsymbol{x}, \mathrm{B})$ already includes both tabular and image data. That is, the transformation function and therefore the outcome distribution is allowed to change with each input $\boldsymbol{x}$ and $\mathrm{B}$, which represents the most flexible model possible. In fact, this most flexible ontram is equivalent to a MCC model with softmax as last-layer activation function and a categorical cross-entropy loss, albeit parametrized differently to take the order of the outcome into account.

Shift terms impose data dependent vertical shifts on the transformation function (see *e.g.,* Fig. 2):

1. Linear shift (LS) terms $\boldsymbol{x}^\top\boldsymbol{\beta}$ are used for tabular features and are directly interpretable (see Section 2.2.1). The components of the parameter $\boldsymbol{\beta}$ can be modeled as the weights of a single layer neural network with input $\boldsymbol{x}$, one output unit with linear activation function and without a bias term (see lower left panel in Fig. 3).
2. Complex shift (CS) terms depend on tabular predictors or image data. Complex shift terms are modeled using flexible dense and/or convolutional NNs with input $\boldsymbol{x}$ and/or $\mathrm{B}$, and a single output unit with linear activation (see lower right panel in Fig. 3). Similar to linear shift terms, the output of $\beta$ and $\eta$ can be interpreted as the log odds of belonging to a higher class, compared to all lower classes, if $F_Z = F_{\mathrm{L}}$. Again, this effect is common to all class boundaries. In contrast to a linear shift term, we can model a complex shift for each tabular predictor $\beta(x_j)$ akin to a generalized additive model. Alternatively, we can model a single complex shift $\beta(\boldsymbol{x})$ for all predictors, which allows for higher order interactions between the predictors. This way, the interpretation of an effect of a single predictor is lost in favour of higher model complexity.

*Interpretability and flexibility* In the following, we will present a non-exhaustive collection of ontrams integrating both tabular and image data. We start to introduce the least complex model with the highest degree of interpretability and end with the most complex model with the lowest degree of interpretability.

The simplest ontram conditioning on tabular data $\boldsymbol{x}$ and image data $\mathrm{B}$ is given by

$$h(y_k|\boldsymbol{x}, \mathrm{B}) = \vartheta_k - \boldsymbol{x}^\top\boldsymbol{\beta} - \eta(\mathrm{B}), \tag{10}$$

where $\vartheta_k$ is a simple intercept corresponding to class $k$, $\boldsymbol{\beta}$ is the weight vector of a single layer NN as described above and $\eta(\mathrm{B})$ the output of a CNN (Fig. 3 A). In this case, $\boldsymbol{\beta}$ and $\eta$ can be interpreted as cumulative log odds-ratios when choosing $F_Z = F_{\mathrm{L}}$ (see Section 2.2.1). The above model can be made more flexible, yet less interpretable, by substituting the linear predictor for a more complex neural network $\beta$, such that

$$h(y_k|\boldsymbol{x}, \mathrm{B}) = \vartheta_k - \beta(\boldsymbol{x}) - \eta(\mathrm{B}), \tag{11}$$

where $\beta(\boldsymbol{x})$ is now a log odds ratio function that allows for higher order interactions between all predictors in $\boldsymbol{x}$. For instance, one may be interested in the odds ratio $\mathrm{OR}_{\mathrm{B}\to\mathrm{B}'}$ of belonging to a

higher category when changing an image B to B′ and holding all other variables constant. As a special case, complex shifts include an additive model formulation in the spirit of generalized additive models (GAMs) by explicitly parametrizing the effect of each predictor $x_j$ with a single neural network $\beta_j$

$$h(y_k|\boldsymbol{x}, \mathrm{B}) = \vartheta_k - \sum_{j=1}^{J} \beta_j(x_j) - \eta(\mathrm{B}),\ j = 1, \ldots, J. \tag{12}$$

For $F_Z = F_{\mathrm{L}}$ the complex shift term $\beta_j(x)$ can be interpreted as a log-odds ratio for the outcome to belong to a higher class than $y_k$ compared to the scenario where $\beta_j(x) = 0$, all other predictors kept constant.

Another layer of complexity can be added by allowing the intercept function $\vartheta_k$ for $Y = y_k$, to depend on the image

$$h(y_k|\boldsymbol{x}, \mathrm{B}) = \vartheta_k(\mathrm{B}) - \beta(\boldsymbol{x}). \tag{13}$$

In this transformation function we call $\vartheta_k(\mathrm{B})$ complex intercept, because the intercept function is allowed to change with the image (Fig. 3 A). One does not necessarily have to stop here. Including both the image and the tabular data in a complex intercept

$$h(y_k|\boldsymbol{x}, \mathrm{B}) = \vartheta_k(\boldsymbol{x}, \mathrm{B}) \tag{14}$$

represents the most flexible model whose likelihood is equivalent to the one used in MCC models, solely with a different parametrization. Consequently, solely the most flexible ontrams achieve on-par performance compared with deep classifiers trained using the cross-entropy loss, while the less flexible ontrams are attractive because of their easier interpretability. In fact, we illustrate empirically that a minor trade-off in predictive performance leads to a considerable ease in interpretation.

*Computational details* The parameters of an ontram are jointly trained via stochastic gradient descent. The parameters enter the loss function via the outputs of the simple/complex intercept and shift terms modeled as neural networks (see Fig. 3 A). The gradient of the loss with respect to all trainable parameters is computed via automatic differentiation in the `TensorFlow` framework. Note that any pre-implemented optimizer can be used and that there are no constraints on the architecture of the individual components besides their last-layer dimension and activation function.

## 4. Experiments

We perform several experiments on data with an ordinal outcome to evaluate and benchmark ontrams in terms of prediction performance and interpretability. For the experiments we use two publicly available data sets as presented in the following section. In addition, we simulate tabular predictors to assess estimation performance for the effect estimates in ontrams.

### 4.1. Data

*UTKFace* UTKFace contains more than 23,000 images of faces belonging to all age groups [dataset 24]. The ordinal outcome is determined by age using the classes baby (0–3, $n_0 = 1894$), child (4–12, $n_1 = 1519$), teenager (13–19, $n_2 = 1180$), young adult (20–30 $n_3 = 8068$), adult (31–45, $n_4 = 5433$), middle aged (46–61, $n_5 = 3216$) and senior (>61, $n_6 = 2395$) [dataset 25]. The images are labeled with the people's age (0 to 116) from which the age-class is determined. In addition, the data set provides the tabular feature sex (female, male). As our main goal is not on performance improvement but on the evaluation of our proposed method, we use the already aligned and cropped versions of the images. For some example images see Fig. 4.

We simulate tabular predictors $\boldsymbol{x}$ with predefined effects on the ordinal outcome of the UTKFace data set, where we assume a proportional odds model $F_Y(y_k|\boldsymbol{x}) = F_Z(\vartheta_k - \boldsymbol{x}^\top \boldsymbol{\beta})$ (see Section 2.2.1). Ten predictors are simulated, four of which are noise predictors that have no effect on the outcome. The six informative predictors are simulated to have an effect of $\pm\log 1.5$, $\pm\log 2$ and $\pm\log 3$ on the log-odds scale, to reflect small to large effect sizes commonly seen in medical and epidemiological applications (Fig. 5). All predictors are mutually independent of each other and the image data. A more detailed description of the simulation procedure is given in Appendix B.2.

Note that in the more complex ontrams involving a CNN, the effect estimates are expected to experience shrinkage towards 0 due to implicit regularization by training via stochastic gradient descent [26] in the presence of the high-dimensional CNN.

*Wine quality* The Wine quality data set consists of 4898 observations [dataset 27]. The ordinal outcome describes the wine quality measured on a scale with 10 levels of which only 6 consecutive classes (3 to 8, $n_3 = 10$, $n_4 = 53$, $n_5 = 681$, $n_6 = 638$, $n_7 = 199$, $n_8 = 18$) are observed. The data set contains 11 predictors, such as acidity, citric acid and sugar content. As in [28], we consider a subset of the data (red wine, $n = 1599$).

### 4.2. Models

The models we use for evaluating and benchmarking the proposed ontrams are summarized in Table 2. The explicit CNN architecture are described in Appendix B.1. These models feature different flexibility and interpretability and are trained with the different loss functions described in Sections 2 and 3 and Appendix A. For UTKFace, we analyse the data set using deep ensembling [29], a state of the art approach in probabilistic deep learning methods leading to more reliable probabilistic predictions [30]. Specifically, models are trained five times with a different weight initialization in each iteration. The resulting predicted conditional outcome distribution is averaged over the five runs and this averaged conditional outcome distribution is then used for model evaluation. This procedure is supposed to prevent double descent and improve test performance [30]. The exact training and validation setup used in the experiments is described in Appendix B.3.

### 4.3. Software

We implement MCC models and ontrams in the two programming languages R 3.6-3 and Python 3.7. The models are written in `Keras` based on a `TensorFlow` backend using `TensorFlow` version >2.0 [31,32] and trained on a GPU. Both polr and generalized additive proportional odds models are fitted in R using `tram::polr()` [33] and `mgcv::gam()` [34], respectively. Further analysis and visualization is performed in R. For reproducibility, all code is made available on GitHub.[2]

### 4.4. Model evaluation

*Evaluation metrics:* The main focus of ontrams is to be able to interpret their individual components and the most flexible ontram is equivalent to the MCC model. In turn, prediction performance of ontrams can only ever be as good as in MCC. Therefore, we assess prediction performance mainly to illustrate trading off model flexibility against ease of interpretation. We evaluate the prediction performance of ontrams and MCCs with proper scoring rules, namely the negative log-likelihood (NLL) and the ranked probability score (RPS). Roughly speaking, proper scoring rules encourage honest probabilistic predictions because they take their optimal value when the predicted conditional outcome distribution

[2] https://www.github.com/LucasKookUZH/ontram-paper.

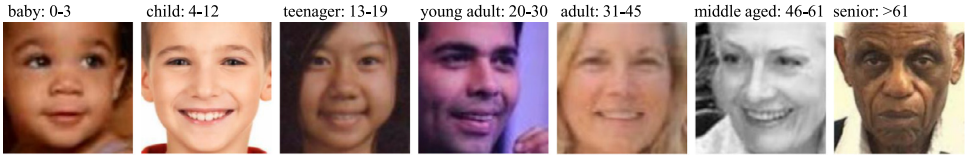


**Fig. 4.** Example images for UTKFace. Example images of the seven ordinal age-classes (baby, child, teenager, young adult, adult, middle aged and senior) of the cropped and aligned UTKFace data set are presented.

**Table 2**
Summary of the models used for evaluating the ontram methods. In the upper part we list models used for the Wine data, which contain only tabular predictors ($\boldsymbol{x}$). In the lower part, we show models for the UTKFace data, which consist of image data and tabular predictors ($\boldsymbol{x}, \mathrm{B}$). Above the thin lines we list the baseline models; below the ontrams. For each model, which can be framed as a transformation model, the transformation function is given. Parameters in the shift terms of a transformation function can be interpreted as log odds-ratios if $F_Z$ is chosen to be the standard-logistic distribution. Then, any model involving a simple intercept is an instance of a proportional odds model.

| Data set | Model name | Abbreviation | Trafo $h(y_k\|\boldsymbol{x}, \mathrm{B})$ |
|---|---|---|---|
| UTKFace | Multi-class classification | MCC | |
| | Multi-class classification + tabular | MCC-$\boldsymbol{x}$ | |
| | Complex intercept | $\mathrm{CI}_\mathrm{B}$ | $\vartheta_k(\mathrm{B})$ |
| | Complex intercept + tabular | $\mathrm{CI}_\mathrm{B}$-$\mathrm{LS}_{\boldsymbol{x}}$ | $\vartheta_k(\mathrm{B}) - \boldsymbol{x}^\top\boldsymbol{\beta}$ |
| | Simple intercept + complex shift | SI-$\mathrm{CS}_\mathrm{B}$ | $\vartheta_k - \eta(\mathrm{B})$ |
| | Simple intercept + complex shift + tabular | SI-$\mathrm{CS}_\mathrm{B}$-$\mathrm{LS}_{\boldsymbol{x}}$ | $\vartheta_k - \eta(\mathrm{B}) - \boldsymbol{x}^\top\boldsymbol{\beta}$ |
| | Simple intercept + tabular | SI-$\mathrm{LS}_{\boldsymbol{x}}$ | $\vartheta_k - \boldsymbol{x}^\top\boldsymbol{\beta}$ |
| Wine | Multi-class classification | MCC | |
| | Generalized additive proportional odds model | GAM | $\vartheta_k - \sum_{j=1}^p \beta_j(x_j)$ |
| | Proportional odds logistic regression | polr | $\vartheta_k - \boldsymbol{x}^\top\boldsymbol{\beta}$ |
| | Complex intercept | $\mathrm{CI}_{\boldsymbol{x}}$ | $\vartheta_k(\boldsymbol{x})$ |
| | Simple intercept + GAM complex shift | SI-$\mathrm{CS}^\star_{\boldsymbol{x}}$ | $\vartheta_k - \sum_{j=1}^p \beta_j(x_j)$ |
| | Simple intercept + linear shift | SI-$\mathrm{LS}_{\boldsymbol{x}}$ | $\vartheta_k - \boldsymbol{x}^\top\boldsymbol{\beta}$ |

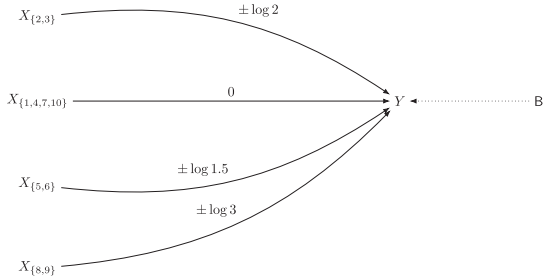


**Fig. 5.** Simulation of predictors for UTKFace data. $X_j \overset{\text{i.i.d.}}{\sim} \mathrm{N}(0, 1.55^2)$, $j = 1, \ldots, 10$. The predictors $X_j$ are simulated such that their effects adhere to the proportional odds assumption. That is, the effect of $\boldsymbol{\beta}$ is common to all class boundaries. Note that the arrows indicate effects on the log-odds scale of the outcome $Y$, *i.e.*, $F_Y(y_k|\boldsymbol{x}) = F_\mathrm{L}(\vartheta_k - \boldsymbol{x}^\top\boldsymbol{\beta})$. The dotted arrow from B to $Y$ indicates that the image is not entering the simulation directly but is assumed to have an effect on the outcome.

corresponds to the data generating distribution (for details see Appendix F). In Appendix H we compute additional evaluation metrics which are commonly used for ordinal classification models, *i.e.*, accuracy and QWK which is discussed in Appendix E.

*Estimation and interpretability* To evaluate whether ontrams yield reliably interpretable effect estimates of shift components we make use of the simulated tabular predictors and compare the known true effects of the individual predictors to the estimates. For other predictors we discuss the plausibility of the estimated effects or, if applicable, compare them to results of other benchmark experiments.

## 5. Results

Results for the MCC models and ontrams for the UTKFace and wine data are given in the following Sections 5.1 and 5.2, respectively.

### 5.1. UTKFace

We first evaluate ontrams on the UTKFace data set, which contains images and tabular predictors that allow to illustrate the interpretation of the shift terms. As in other applications, age is discretized and treated as an ordinal outcome (see *e.g.*, [35]).

We first train a SI-$\mathrm{CS}_\mathrm{B}$-$\mathrm{LS}_\mathrm{sex}$ ontram with transformation function $h(y_k|\boldsymbol{x}) = \vartheta_k - \eta(\mathrm{B}) - \beta_\mathrm{sex} \cdot \mathbb{1}(\mathrm{sex} = \mathrm{female})$ that includes the tabular predictor sex in addition to the images. We assume that the prediction of the age class depends on the appearance of a person and therefore on the image but not on a person's sex. On the other hand, a person's sex can often be deduced from an image, which renders the tabular feature and image data collinear and makes estimation and interpretability of the individual effects more difficult. However, collinear data is representative for most practical applications. We thus expect the estimated coefficient $\beta_\mathrm{sex}$ to be small in comparison to the effect of the image $\eta(\mathrm{B})$, which we expect to be a better predictor of a person's age.

For evaluation, we use publicly available data of the actress Meryl Streep, *i.e.*, female sex and two images showing her at the age of 41 ($\mathrm{B}$, age group $[31, 46)$) and 67 ($\mathrm{B}'$, age group $[61, 117)$) to depict the predicted PDF and estimated log odds-ratio in the SI-$\mathrm{CS}_\mathrm{B}$-$\mathrm{LS}_\mathrm{sex}$ model (see Fig. 6). The model yields the image-effect estimates $\eta(\mathrm{B}) = 5.1$ and $\eta(\mathrm{B}') = 10.1$, while the effect of sex stays constant ($\beta_\mathrm{sex} = 0.3$). As expected $\eta(\mathrm{B}') > \eta(\mathrm{B})$, indicating that $\mathrm{B}'$ is more likely to belong to a higher age group than B. In particular, the difference between the two estimates yields a log odds-ratio $\eta(\mathrm{B}') - \eta(\mathrm{B}) = 5$, which is interpretable as an exp(5)-fold increase in the odds of belonging to a higher age class compared to all classes below, when changing from B to $\mathrm{B}'$ and keeping sex constant.

For a more systematic and empirical evaluation of the flexibility and interpretability of ontrams, we fit seven models with the image data, the 10 simulated tabular predictors with known true effect sizes $\boldsymbol{\beta}$ and a combination of both (see Table 2). The models differ in their flexibility due to different transformation functions and the parametrization of the loss. In Appendix G, we com-

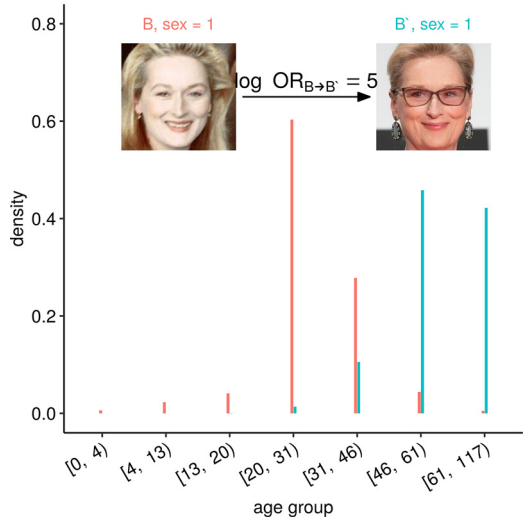


**Fig. 6.** Predicted densities in a SI-CS$_B$-LS$_{sex}$ ONTRAM once using the image of a 41 year-old and a 67 year-old Meryl Streep. What sets ONTRAMS apart from other ordinal DL classifiers, is the directly interpretable effect of changing an image in terms of a log odds-ratio. Namely, the odds of belonging to a higher age change by a factor of exp(5) when changing image B to B′, keeping sex constant. In turn, a change in the odds results in a change of the corresponding conditional outcome distribution, which puts higher probability mass on larger age groups when changing B to B′. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

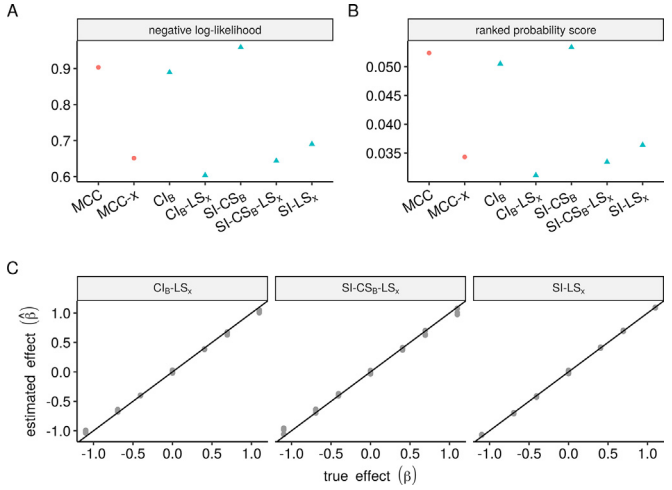


**Fig. 7.** Test prediction performance for deep ensembles based on the UTKFace data. The figure summarizes the results for the models MCC, MCC-$\boldsymbol{x}$, CI$_B$, CI$_B$-LS$_{\boldsymbol{x}}$, SI-CS$_B$, SI-CI$_B$-LS$_{\boldsymbol{x}}$, SI-LS$_{\boldsymbol{x}}$ (x-axes) in terms of negative log-likelihood (A) and ranked probability score (B). Lower values in NLL and RPS indicate better predictive performance. Baseline models are depticted as red dots, ONTRAMS as blue triangles. C: True versus estimated predictor effects. The figure summarizes the true versus estimated effects of the simulated tabular predictors of the UTKFace data set. The effect estimates result from the linear shift terms, LS$_{\boldsymbol{x}}$, in the models CI$_B$-LS$_{\boldsymbol{x}}$, SI-CS$_B$-LS$_{\boldsymbol{x}}$, SI-LS$_{\boldsymbol{x}}$. In case of correct estimation, the parameters lie on the main diagonal. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

pare the MCC model and the CI$_{\boldsymbol{x}}$ ONTRAM to another ordinal classification model trained with a loss based on Cohen's quadratic weighted kappa [QWK, 17].

We first consider the most flexible models, MCC and CI$_B$, which are based on the UTKFace image data and only differ in the parametrization of the loss function (see Appendix A for the MCC and Eq. (8) for the ONTRAM loss). As expected, the CI$_B$ ONTRAM and MCC model achieve comparable prediction performances in terms of NLL and RPS (see Fig. 7 A and B). After including the simulated tabular predictors, the performance in both models increases notably (see MCC-$\boldsymbol{x}$ and CI$_B$-LS$_{\boldsymbol{x}}$ in Fig. 7 A and B). In case of the MCC-$\boldsymbol{x}$ model, the tabular predictors are attached to the feature vector resulting from the convolutional part of the CNN, which allows interactions between image and tabular predictors and therefore makes the model slightly more flexible than the CI$_B$-LS$_{\boldsymbol{x}}$. However, in contrast to the CI$_B$-LS$_{\boldsymbol{x}}$, the MCC-$\boldsymbol{x}$ allows no interpretation of the effect of the tabular predictors on the outcome.

Less flexible but more interpretable ONTRAMS are obtained by including the image data as complex shift rather than as complex intercept term (SI-CS$_B$). Although the SI-CS$_B$ model is less flexible than the CI$_B$ model, prediction performance is comparable (see Fig. 7 A and B). Again, adding the simulated tabular data as a linear shift term (SI-CS$_B$-LS$_{\boldsymbol{x}}$) results in improved prediction performance.

Using a model with simulated tabular data only (SI-LS$_{\boldsymbol{x}}$) yields a better performance than models that include image data only (see SI-LS$_{\boldsymbol{x}}$ *vs.* MCC, CI$_B$, SI-CS$_B$ in Fig. 7 A and B). However, when comparing the models with image data and tabular predictors to the model with tabular predictors only, an increase in prediction performance is observed (see SI-LS$_{\boldsymbol{x}}$ *vs.* MCC-$\boldsymbol{x}$, CI$_B$-LS$_{\boldsymbol{x}}$ and SI-CS$_B$-LS$_{\boldsymbol{x}}$). This indicates that the images contain additional information for age prediction.

In practice, the ONTRAMS CI$_B$-LS$_{\boldsymbol{x}}$ and SI-CS$_B$-LS$_{\boldsymbol{x}}$ are most attractive because they provide interpretatable estimates for the effects of the tabular predictors with an acceptably low decrease in prediction performance.

To assess whether effect estimates for the tabular predictors are reliable in models with and without additional image data, we compare the true effects $\beta$ to the estimated effects $\hat{\beta}$ for the ONTRAMS with linear shift terms (CI$_B$-LS$_{\boldsymbol{x}}$, SI-CS$_B$-LS$_{\boldsymbol{x}}$, SI-LS$_{\boldsymbol{x}}$). As summarized in Fig. 7 C, all models recover the correct estimates up to minor shrinkage effects in the presence of high-dimensional CNNs.

### 5.2. Wine quality

The experiments with the UTKFace data have shown that we get reliable and interpretable model components when including simulated, mutually independent tabular predictors besides image data. In the following, we summarize a couple of experiments with the smaller wine data set containing solely tabular predictors to demonstrate how we can estimate reliable linear and non-linear effect estimates for potentially dependent tabular predictors. In addition, we evaluate how the ONTRAM parametrization of the loss (see Eq. (8)) yields a gain in training speed and how this gain depends on the size of the training data. Note that all those models can simply be extended to additionally include image data, e.g., by attaching a complex shift term CS$_B$.

The wine dataset is a benchmark data set for a proportional odds model that allows to interpret the fitted effect estimates as log odds-ratios (see Section 2.2.1). To illustrate the high flexibility of ONTRAMS and that we correctly estimate linear, non-simulated tabular predictors, we fit a proportional odds model with linear effects via a SI-LS$_{\boldsymbol{x}}$ model and compare the model to the same model using the R function `tram::Polr()`. As expected, Fig. 8 shows that both models yield the same prediction performance in terms of NLL (A) and RPS (B) and estimated predictor effects (C).

GAMs (see Table 2, SI-CS$^*_{\boldsymbol{x}}$ with $h(y_k|D) = \vartheta_k - \sum_{j=1}^{p} \beta_j(x_j)$) add another layer of complexity to the model by allowing non-linear effects for each predictor. Because the individual NNs estimating the additive components $\beta_j(x_j)$ do not interact explicitly the estimated log odds-ratio function retains the interpretability of a proportional odds model. Fig. 9 depicts the estimates of an ensemble of ONTRAM GAMs in comparison to a GAM from the R-package **mgcv**. Apart from the constraint-enforced smoothness in **mgcv**'s GAM, both models agree in magnitude and shape of the estimated predictor effects. For instance, the predictor `sulphates` has a

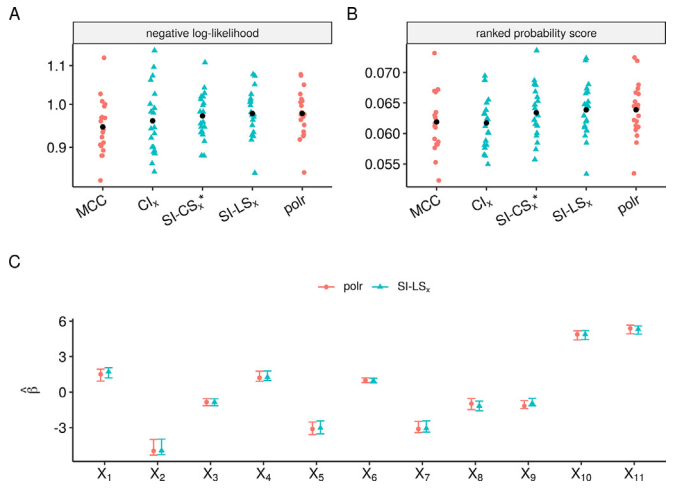


**Fig. 8.** Results for the wine quality data based on the test sets of the cross validation settings. Panels A and B summarize the prediction performance for the models MCC, CI$_{\boldsymbol{x}}$, GAM, SI-LS$_{\boldsymbol{x}}$ and polr (x-axes) based on the wine quality data set in terms of negative log likelihood (A) and ranked probability score (B). Lower values in NLL and RPS indicate improved model performance. Results of ONTRAMS are indicated as blue triangles, others as red dots. The black point gives the mean across the respective metric resulting from the single CV folds. C: Effect estimates with 2.5th and 97.5th percentile for polr and SI-LS$_{\boldsymbol{x}}$ model over the 20 CV folds of the wine quality data set. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

strong positive influence on the rating when increased from 0 to 0.25 (on the transformed scale), in that the odds of the wine being rated higher increase by a factor of 7.4, all other predictors held constant ($\exp(\hat{\beta}_{10}(0.25) - \hat{\beta}_{10}(0)) \approx \exp(2) \approx 7.4$). Afterwards the effect levels off and stays constant for the ONTRAM GAM, due to regularization and few wines with higher sulphate levels being present in the training data. The curve estimated by **mgcv** follows smoothness constraints and instead drops with a large confidence interval, also covering 0. GAMs are a special case of complex shift models, the latter of which allow for higher order interactions between the predictors. Conceptually, ONTRAMS enable to further trade off interpretability and flexibility by modelling some predictor effects linearly while including others in a complex shift or intercept term. If field knowledge suggests non-linearity of effects or interacting predictors, they can be included as a complex shift or, if the proportionality assumption is violated, in a complex intercept term. From Fig. 9 we can see that most of the coefficients could be safely modelled in a linear fashion, which is also evident from the minor loss in predictive power when comparing the GAM against the linear shift ONTRAM (see Fig. 8 A, GAM *vs.* SI-LS$_{\boldsymbol{x}}$).

To assess the effect of respecting the order of the ordinal outcome, we evaluate the most flexible CI$_{\boldsymbol{x}}$ ONTRAM and the MCC model, which solely differ in the parametrization of their loss. As in the UTKFace data, both models show the expected agreement in achieved prediction performance w.r.t. NLL and RPS (see Fig. 8 A and B). However, the CI$_{\boldsymbol{x}}$ model learns much faster in terms of number of epochs until the minimum test loss is achieved, compared to the MCC model. To further investigate this gain in learning speed, we split the wine quality data into $n/n_t$, $n_t \in \{50, 100, 200, 480\}$ folds of size $n_t$ and fit a MCC model and CI$_{\boldsymbol{x}}$ ONTRAM to each fold. The median test loss is computed for each scenario of size $n_t$. The number of epochs needed to achieve minimal median test loss is summarized in Fig. 10 A. The training speed is consistently lower and therefore more efficient for the CI$_{\boldsymbol{x}}$ ONTRAM than for the MCC model (Fig. 10 A). The CI$_{\boldsymbol{x}}$ ONTRAM yields a slightly better prediction performance (median test NLL) for larger sample sizes. This can be explained by the fact that after

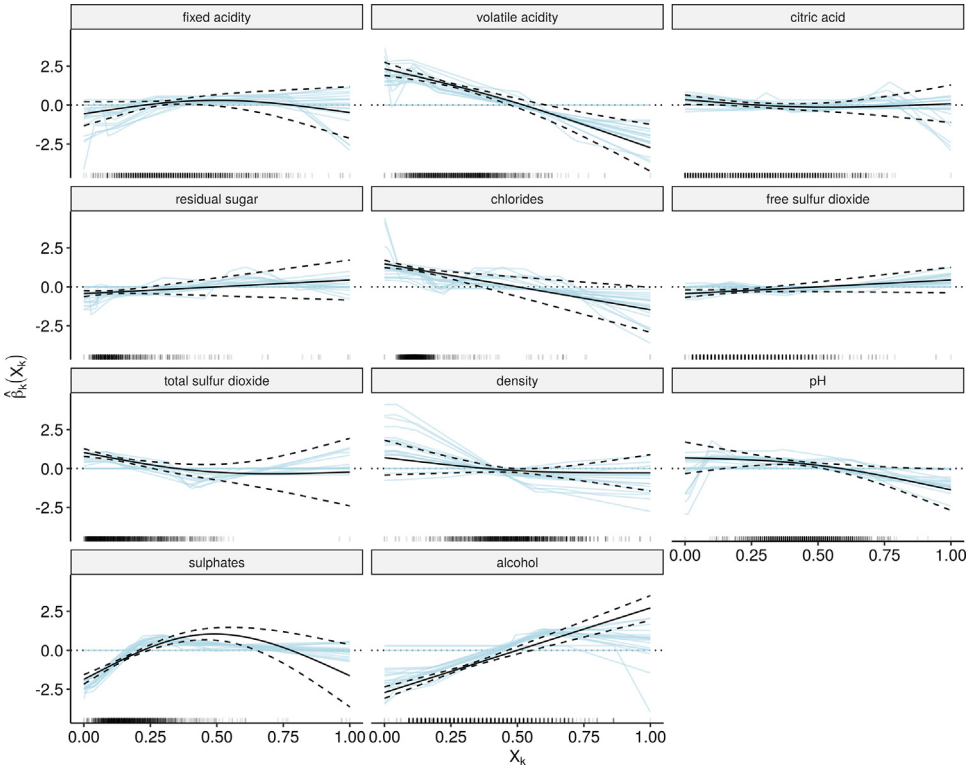


**Fig. 9.** Estimated non-linear effects of 11 tabular predictors on the ordinal quality outcome in the wine data set as achieved by the ONTRAM GAM model. The estimates log odds-ratio functions of an ensemble of 20 runs with different initial weights are shown in blue. The solid black line depicts the estimated log odds-ratio functions estimated by the `mgcv::gam()` function in R together with a 95% confidence interval (dashed black lines). Rugs on the bottom of each plot indicate the observed values for $X_k$, $k = 1, \ldots, 11$, in the training data. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

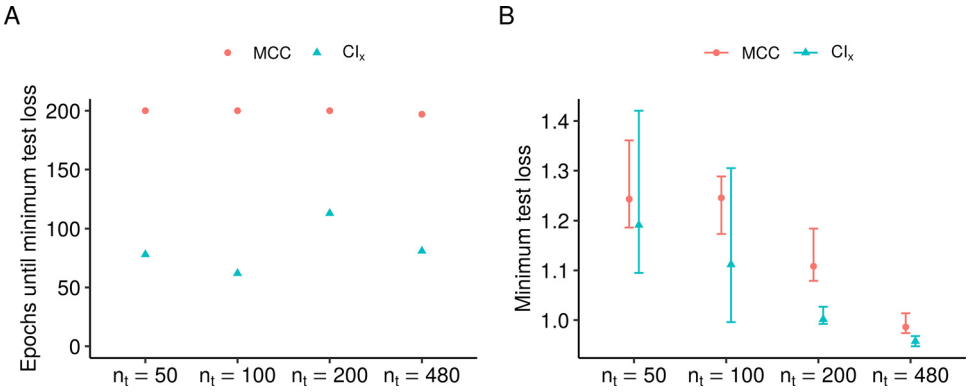


**Fig. 10.** Epochs until minimum test loss for varying sizes of the training data using the wine quality data set. The data are split into $n/n_t$, $n_t \in \{50, 100, 200, 480\}$ folds each of which serves as training data for a multi-class classification and a complex intercept ontram. The median test loss is computed for each scenario $n_t$ and each epoch. Afterwards, the number of epochs until minimum median test loss and the minimum median test loss are recorded. Here the epochs until minimum test loss (A) and the minimum test loss (B) are plotted against the 4 scenarios given by $n_t$.

200 epochs the MCC model still has not reached the minimum test loss (Fig. 10 B). Note that the gain in training speed is only present if the outcome is truly ordered. In Appendix C, we show that the effect vanishes when the ordering of the class labels is permuted.

## 6. Discussion and outlook

In this work we demonstrate how to unite the classical statistical approach to ordinal regression with DL models to achieve interpretability of selected model components. This allows us to estimate effects for the input data. In case of tabular predictors, we prove that the effects are correctly estimated, also in the presence of complex image data. Moreover, we show that the most flexible ontram trained with the reparametrized NLL achieves on-par performance with a MCC DL model using the cross-entropy loss. This may first seem counter-intuitive because the cross-entropy loss ignores the outcome's order. However, the ontram NLL is a reparametrization of the cross-entropy loss and can, therefore, at most achieve the same performance. The advantages of reparametrizing the NLL are (i) a natural scale for the additive and hence interpretable decomposition of tabular and image effects, (ii) a valid probability distribution for the ordinal outcome and (iii) an increase in training speed. In this context, interpretability is the main advantage over other state-of-the-art models because it is of crucial importance in sensitive applications as, for example, in medicine [4].

If the focus lies mainly on classification of an ordinal outcome and less on interpretability and probabilistic predictions, the data analyst may be interested in optimizing a classification metric such as Cohen's kappa. Indeed, Cohen's kappa directly considers the outcome's natural order and misclassifications further away from the observed class are penalized more strongly than misclassifications closer to the observed class. However, this approach results in predictions different from those of regression models such as the MCC and $CI_B$, which is further highlighted in Appendix G. In a regression model, on the other hand, the goal is rather to estimate a valid probability distribution which is achieved with proper loss functions such as the NLL. These fundamental differences between ordinal classification and regression make a fair comparison nearly impossible, as we highlight in Appendix G.

Further, we demonstrate how to select an ontram, which possesses the appropriate amount of flexibility and interpretability for a given application. To achieve a higher degree of interpretability, flexibility has to be restricted, *e.g.,* by moving from a complex intercept to a simple intercept, complex shift model. However, we show that a restriction of flexibility can still yield adequate prediction performance which may even be similar to that of a more flexible model. Interpretability of different model components is further showcased for simple models including only tabular predictors and more complex models with tabular and image data.

The modular nature of ontrams makes them highly versatile and applicable to many other problems with ordinal outcome and complex input, such as text or speech data. Instead of using a CNN for image data, a recurrent neural network can be used to define a more flexible complex intercept or a simpler, but more interpretable complex shift term as in a SI-$CS_B$ ontram. Tabular predictors can then simply be added with linear shift or complex shift terms depending on the degree of interpretability the data analyst aims for.

This work shows the potential of deep transformation models for ordinal outcomes. The predictive power of deep transformation models on regression problems with continuous outcomes has already been demonstrated [6]. However, the approach is easily extendable to the full range of existing interpretable regression models, including models for count and survival outcomes. The extension from ordinal data to count and survival data is hinted at by the parametrization of the ontram NLL, which can be viewed as an interval-censored log-likelihood over the latent variable $Z$ for which the intervals are given by the conditional cut points $h(y_k|D)$. For count data these cut points are given by consecutive integers, *i.e.*, (0,1], (1,2], and so on. In survival data the interval is given by (commonly) right censored outcomes when a patient drops out of a study or experiences a competing event. In case of right-censoring the interval is given by $(t, +\infty)$ for a patient that drops out at time $t$. All benefits in terms of interpretability and modularity will carry over to the deep transformation version of other probabilistic regression models by working with an appropriate likelihood and parametrizing the transformation function via (deep) neural networks.

## 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.

## Acknowledgments

We would like to thank Elvis Murina and Muriel Buri for insightful discussions and Malgorzata Roos for her feedback on the manuscript. We thank all anonymous reviewers for their comments and suggestions, which helped contextualize our proposed

method among other state-of-the-art approaches. The research of LH, LK and BS was supported by Novartis Research Foundation (FreeNovation 2019). TH was supported by the Swiss National Science Foundation (SNF) under the project “A Lego System for Transformation Inference” (grant no. 200021_184603). OD was supported by the Federal Ministry of Education and Research of Germany (BMBF) in the project “DeepDoubt” (grant no. 01IS19083A).

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2021.108263.

## References

[1] A. Oghina, M. Breuss, M. Tsagkias, M. De Rijke, Predicting IMDb movie ratings using social media, in: European Conference on Information Retrieval, Springer, 2012, pp. 503–507.
[2] P. McCullagh, Regression models for ordinal data, J. R. Stat. Soc. Ser. B 42 (2) (1980) 109–127, doi:10.1111/j.2517-6161.1980.tb01109.x.
[3] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT press, 2016, doi:10.1007/s10710-017-9314-z.
[4] 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, doi:10.1038/s42256-019-0048-x.
[5] T. Hothorn, T. Kneib, P. Bühlmann, Conditional transformation models, J. R. Stat. Soc. Ser. B (2014) 3–27, doi:10.1111/rssb.12017.
[6] B. Sick, T. Hothorn, O. Dürr, Deep transformation models: tackling complex regression problems with neural network based transformation models, in: Accepted for Proceedings of the 25th International Conference on Pattern Recognition (ICPR), Milan/Online, 2021, 2021.
[7] G. Tutz, Regression for Categorical Data, vol. 34, Cambridge University Press, 2011, doi:10.1017/CBO9780511842061.
[8] V.M. Vargas, P.A. Gutiérrez, C. Hervás-Martínez, Cumulative link models for deep ordinal classification, Neurocomputing (2020), doi:10.1016/j.neucom.2020.03.034.
[9] W. Chu, Z. Ghahramani, Gaussian processes for ordinal regression, J. Mach. Learn. Res. 6 (Jul) (2005) 1019–1041.
[10] W. Chu, S.S. Keerthi, Support vector ordinal regression, Neural Comput. 19 (3) (2007) 792–815, doi:10.1162/neco.2007.19.3.792.
[11] J.S. Cardoso, J.F. Costa, Learning to classify ordinal data: the data replication method, J. Mach. Learn. Res. 8 (Jul) (2007) 1393–1429.
[12] E. Frank, M. Hall, A simple approach to ordinal classification, in: European Conference on Machine Learning, Springer, 2001, pp. 145–156, doi:10.1007/3-540-44795-4_13.
[13] Z. Niu, M. Zhou, L. Wang, X. Gao, G. Hua, Ordinal regression with multiple output CNN for age estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4920–4928, doi:10.1109/CVPR.2016.532.
[14] J. Cheng, Z. Wang, G. Pollastri, A neural network approach to ordinal regression, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 1279–1284, doi:10.1109/IJCNN.2008.4633963.
[15] W. Cao, V. Mirjalili, S. Raschka, Rank-consistent ordinal regression for neural networks, arXiv preprint arXiv:1901.07884(2019).
[16] J.-C. Xie, C.-M. Pun, Deep and ordinal ensemble learning for human age estimation from facial images, IEEE Trans. Inf. Forensics Secur. 15 (2020) 2361–2374, doi:10.1109/TIFS.2020.2965298.
[17] J. de La Torre, D. Puig, A. Valls, Weighted kappa loss function for multi-class classification of ordinal data in deep learning, Pattern Recognit. Lett. 105 (2018) 144–154, doi:10.1016/j.patrec.2017.05.018.
[18] V.M. Vargas, P.A. Gutiérrez, C. Hervás, Deep ordinal classification based on the proportional odds model, in: International Work-Conference on the Interplay Between Natural and Artificial Computation, Springer, 2019, pp. 441–451, doi:10.1109/BTAS.2016.7791154.
[19] Y. Liu, F. Wang, A.W.K. Kong, Probabilistic deep ordinal regression based on gaussian processes, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 5301–5309, doi:10.1109/ICCV.2019.00540.
[20] M.T. Ribeiro, S. Singh, C. Guestrin, “Why should I trust you?” Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.
[21] J.P. Amorim, I. Domingues, P.H. Abreu, J.A. Santos, Interpreting deep learning models for ordinal problems., in: Proceedings of the European Symposium on Artificial Neural Networks, 2018.
[22] Y. Liu, A.W.-K. Kong, C.K. Goh, Deep ordinal regression based on data relationship for small datasets., in: Proceedings of the International Joint Conference on Artificial Intelligence, 2017, pp. 2372–2378, doi:10.24963/ijcai.2017/330.
[23] Y. Liu, A. Wai Kin Kong, C. Keong Goh, A constrained deep neural network for ordinal regression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 831–839, doi:10.1109/CVPR.2018.00093.
[24] Y. Song, Z. Zhang, Utkface data set, 2020, Accessed: April (https://susanqq.github.io/UTKFace/).
[25] A. Das, A. Dantcheva, F. Bremond, Mitigating bias in gender, age and ethnicity classification: a multi-task convolution neural network approach, in: ECCVW 2018 - European Conference of Computer Vision Workshops, 2018.
[26] A. Ali, E. Dobriban, R. Tibshirani, The implicit regularization of stochastic gradient flow for least squares, in: H.D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 233–244.
[27] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis, Modeling wine preferences by data mining from physicochemical properties, Decis. Support Syst. 47 (4) (2009) 547–553, doi:10.1016/j.dss.2009.05.016.
[28] Y. Gal, Z. Ghahramani, Dropout as a Bayesian approximation: representing model uncertainty in deep learning, in: International Conference on Machine Learning, 2016, pp. 1050–1059.
[29] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in: Advances in Neural Information Processing Systems, 2017, pp. 6402–6413.
[30] A.G. Wilson, P. Izmailov, Bayesian deep learning and a probabilistic perspective of generalization, Advances in Neural Information Processing Systems (NeurIPS), 2020.
[31] F. Chollet, et al., Keras, 2015, (https://keras.io).
[32] M. Abadi, et al., TensorFlow: large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
[33] T. Hothorn, **tram**: Transformation Models, 2020. Rpackage version 0.5-1.
[34] S.N. Wood, Generalized Additive Models: An Introduction with R, second ed., Chapman and Hall/CRC, 2017.
[35] H. Liu, J. Lu, J. Feng, J. Zhou, Ordinal deep learning for facial age estimation, IEEE Trans. Circuits Syst. Video Technol. 29 (2) (2017) 486–501, doi:10.1109/FG.2017.28.

**Lucas Kook** is a PhD candidate at the Department of Biostatistics at the University of Zurich and the Zurich University of Applied Sciences. His research focus lies on distributional regression, causal inference and deep learning.

**Lisa Herzog** is a PhD candidate at the University of Zurich and the Zurich University of Applied Sciences. She has a strong background in biostatistics and machine learning. The goal of her thesis is to develop DL models for medical image analysis to improve functional outcome prediction in stroke patients.

**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 flexible regression models for analyzing biomedical data.

**Oliver Dürr** is a Professor for Data Science at the Konstanz University of Applied Sciences. After his PhD in theoretical physics, he worked in a bioinformatics company. He then was a lecturer for statistics at the ZHAW. Now, he is working on deep learning mainly from a probabilistic perspective.

**Beate Sick** is a Professor for Applied Statistics at ZHAW and co-affiliated 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.