

UNICORNN: UNIMODAL CALIBRATED ORDINAL REGRESSION NEURAL NETWORK

Anonymous authors

Paper under double-blind review

ABSTRACT

Ordinal regression is a supervised machine learning technique aimed at predicting the value of a discrete dependent variable with an ordered set of possible outcomes. Many of the algorithms that have been developed to address this issue rely on maximum likelihood for training. However, the standard maximum likelihood approach often fails to adequately capture the inherent order of classes, even though it tends to produce well-calibrated probabilities. Alternatively, some methods use Optimal Transport (OT) divergence as their training objective. Unlike maximum likelihood, OT accounts for the ordering of classes; however, in this manuscript, we show that it doesn't always yield well-calibrated probabilities. To overcome these limitations, we introduce UNICORNN, an approach inspired by the well-known Proportional Odds Model, which offers three key guarantees: (i) it ensures unimodal output probabilities, a valuable feature for many real-world applications; (ii) it employs OT loss during training to accurately capture the natural order of classes; (iii) it provides well-calibrated probability estimates through a post-training accuracy-preserving calibration step. Experimental results on six real-world datasets demonstrate that UNICORNN consistently either outperforms or performs as well as recently proposed deep learning approaches for ordinal regression. It excels in both accuracy and probability calibration, while also guaranteeing output unimodality. The code will be publicly available upon acceptance.

1 INTRODUCTION

Ordinal regression is an area of supervised machine learning, where the goal is to predict the value of a discrete dependent variable, whose set of (symbolic) possible values is ordered. Despite often being overshadowed by more common tasks like classification and regression, ordinal regression covers a wide range of important applications, such as medical severity grading, credit rating, age estimation, and many more (De Vente et al., 2020; Wienholt et al., 2024; Niu et al., 2016; Kim and Ahn, 2012).

Many practitioners often treat ordinal regression problems as classification or regression problems (for example, this was the case with many submissions to Kaggle's Diabetic Retinopathy competition¹ in 2015). While having common characteristics with both classification and regression, ordinal regression can arguably be viewed as a mid-point between the two. An ordinal model is of course similar to a classification model, in that both predict a discrete value ("label") out of a finite set of possible ones. However, the existence of an order on the set of labels, when available, can potentially lead to improved performance compared to a standard classifier, which does not assume such order exists. This typically occurs via distinguishing between the severity of prediction mistakes: while in classification typically "all mistakes are created equal", in ordinal regression different mistakes may be associated with different severity (for example, in the context of tumor grade prediction, predicting "3" when the ground truth value is "4" may be less severe than a "1" prediction). In regression problems, the dependent variable naturally does take values from an ordered set, however, this set is typically a continuum and is treated numerically, while in ordinal regression it lacks any numerical relation beyond merely order. Therefore, regression performance may be sensitive to monotonic transformations of the dependent variable, while such sensitivity does not take place in ordinal regression problems, as the order is invariant to monotonic transformations. Hence one may

¹<https://www.kaggle.com/c/diabetic-retinopathy-detection/>

054 expect that typical ordinal regression algorithms have the potential to outperform classification or
055 regression approaches when the range of the dependent variable is finite and ordered.

056 A fundamental ordinal regression model is the Proportional Odds Model (POM) (McCullagh, 1980),
057 a generalized linear model similar in spirit to logistic regression, however the logits are defined
058 for their cumulative probabilities. One potential source of sub-optimality of POM and of several
059 recently-proposed approaches for deep ordinal regression, is the often-reasonable requirement that a
060 probabilistic model for ordinal regression will output unimodal probabilities. A k -level multinomial
061 distribution is called *unimodal* if there exists $j \in \{1, \dots, k\}$ such that $\Pr(Y = 1) \leq \dots \leq \Pr(Y =$
062 $j) \geq \dots \geq \Pr(Y = k)$. Although there are domains in which unimodality is not necessarily a
063 desirable property, such as tasks where the most common targets tend to fall at the extremes, in many
064 other real-world domains it is a natural requirement, for example, when predicting a grade of a tumor,
065 it may be counter-intuitive to trust a model prediction which says that a predicted tumor’s grade
066 is either "1" or "4", but not "2" or "3". However, unimodality is unfortunately not always fulfilled
067 despite often being a desired characteristic. While this was identified by several recent works for deep
068 ordinal regression (Gao et al., 2017; Diaz and Marathe, 2019; Liu et al., 2019a; 2020), unimodality is
069 often encouraged (but not enforced) via soft targets. However, as we show in Section 2, using soft
070 targets is suboptimal for achieving unimodality.

071 Another essential feature of an ordinal regression model is its ability to effectively capture the ordered
072 relationship among classes within its training objective, while still reflecting the certainty of the
073 model in its predictions. POM is typically trained via maximum likelihood, similar to several recently
074 proposed deep ordinal regression approaches (Belharbi et al., 2019; Vargas et al., 2020; Fu et al., 2018;
075 Beckham and Pal, 2017; Berg et al., 2020; Vishnu et al., 2019). We argue that maximum likelihood
076 is a sub-optimal measure of quality for ordinal regression setup, as it only considers the probability
077 mass the model assigns to the true class, ignoring the remaining mass. This implicitly assumes that
078 “all mistakes are equal”, which, as discussed above, is not the case for ordinal regression. However, a
079 benefit of maximum likelihood is that it tends to yield well-calibrated probabilities. A well-calibrated
080 model ensures that, for example, a 90% predicted probability corresponds to events that actually
081 occur 90% of the time. However, many models, including recent deep ordinal regression approaches,
082 struggle with this. These models often exhibit overconfidence or underconfidence, meaning their
083 probability outputs are unreliable indicators of their true prediction certainty. Another commonly
084 used loss function for ordinal regression tasks is Optimal Transport (OT) divergence. OT excels at
085 capturing the inherent order between labels, potentially making it a better fit. However, as we explain
in Section 3.3, OT might lead to peaked output distribution that lacks calibration.

086 In this manuscript, we therefore focus on two main contributions. First, We present UNICORN, a
087 novel approach for ordinal regression, based on deep learning machinery, which tackles the three
088 issues pointed out above (i) it contains a mechanism to enforce *unimodality of the output distribution*,
089 implemented via architectural design, (ii) it *effectively captures the ordered relationship among*
090 *classes* using OT as a training objective (iii) it undergoes a post-training calibration process to output
091 *well-calibrated probability estimates* that reflect the model’s confidence in its predictions while still
092 preserving the model’s accuracy. Second, as discussed in Section 3.3, we identify a trade-off between
093 certain requirements, noting that OT may prioritize peaked distributions over calibrated ones, which,
094 to the best of our knowledge, was not pointed out in the literature in the context of deep ordinal
095 regression. Importantly, this bonds the requirements together, as other methods that utilize OT as
096 a training objective, end up being uncalibrated. We present experimental results on six real-world
097 image benchmark datasets which demonstrate that UNICORN consistently performs on par with
098 and often better than several recently proposed approaches for deep ordinal regression in terms of
both prediction accuracy and probability calibration while having an unimodality guarantee.

100 2 RELATED WORK

101 Being a traditional area of machine learning and statistics, there exists a large corpus of literature on
102 ordinal regression. In this section, we focus on approaches based on recent deep-learning architectures.
103 Several such approaches have been proposed in recent years. One common approach is to turn the
104 ordinal regression problem into a multi-label classification problem (Fu et al., 2018; Liu et al., 2017;
105 2018b; Vishnu et al., 2019; Berg et al., 2020; Cheng et al., 2008; Li et al., 2021). We argue that the
106 multi-label approach has two major problematic aspects: first, the output probabilities are not always
107

108 guaranteed to be consistent, in the sense of increasing cumulative distribution (i.e., we would like
109 to predict $\Pr(y \leq 1) \leq \Pr(y \leq 2) \leq \dots \leq \Pr(y \leq k)$). Second, even if the output probabilities
110 are consistent, as is the case in Liu et al. (2018a); Shi et al. (2023); Cao et al. (2020), the predicted
111 class probabilities are not necessarily unimodal. This is the case in several recent works (Liu et al.,
112 2019b; Vargas et al., 2020; Pan et al., 2018; Kook et al., 2020). **Another line of research focuses on**
113 **addressing ordinal regression by developing unbiased estimator-based approaches that are robust to**
114 **label noise, as proposed by Garg and Manwani (2020).**

115 One elegant mechanism is to obtain unimodal output probabilities, based on either the Poisson or
116 the Binomial distributions (Beckham and Pal, 2017), which are both unimodal. In both cases, the
117 model outputs a scalar (λ in the case of the Poisson, p in the case of the binomial) for each prediction,
118 which is then mapped to a probability mass function that is used (after normalization) as the model
119 output probabilities. Moreover, this method also learns a dataset-wide τ parameter which controls the
120 shape of the output distribution. While being a convenient architectural-based solution for handling
121 unimodality, this approach is inherently limited in its ability to express the level of uncertainty of the
122 model’s prediction. To see why, note that since a single parameter determines both the location of
123 the mode and the decay of the probabilities, the model cannot output a highly flat or highly peaked
124 probability vector, for example.

125 A different approach to unimodality has been to train the model with soft targets (Gao et al., 2017;
126 Diaz and Marathe, 2019; Liu et al., 2019a; 2020). However, we argue that the utilization of soft targets
127 suffers from two important disadvantages. First, unimodality is only encouraged, but not enforced.
128 In Section 5 we will show cases where models trained with soft targets yield large amounts of
129 non-unimodal predictions. Second, soft targets have a pre-defined decay pattern, which is determined
130 a-priori and hence does not reflect any level of uncertainty with respect to the prediction. Therefore,
131 they are equivalent (in the sense of a 1:1 map) to Dirac predictions (i.e., “one-hot”), and are devoid
132 of any probabilistic insight whatsoever. As we show in this paper, our approach attends to both
133 issues: we guarantee unimodal outputs, by design, and yield well-calibrated probabilities outputs that
134 reflect the model’s uncertainty. Other approaches for handling unimodality include Li et al. (2022);
135 Cardoso et al. (2023), where unimodality is encouraged through a dedicated loss term (although
136 not guaranteed), and Belharbi et al. (2019), which employs constrained optimization to achieve
unimodality on the training data, but with no guarantees on the predictions on test data.

137 Several works use cross entropy as a training objective while using one-hot (or binary) targets (Bel-
138 harbi et al., 2019; Vargas et al., 2020; Fu et al., 2018; Beckham and Pal, 2017; Berg et al., 2020;
139 Vishnu et al., 2019; Cardoso et al., 2023). Also, an accuracy-preserving calibration on a cross-entropy
140 trained model was proposed in (Esaki et al., 2024). However, in the case of one-hot targets, the
141 cross entropy term equals the negative logarithm of the probability assigned by the model to the true
142 class, making it invariant to the distribution of the remaining probability mass. While reasonable
143 in a standard classification setting, this ignores the order of the classes, making it sub-optimal for
144 an ordinal regression setting. Nonetheless, as cross-entropy is a proper-scoring rule, it tends to
145 yield calibrated probability estimates (Lakshminarayanan et al., 2017). To address cross-entropy’s
146 limitation in handling the order of classes, some approaches use OT loss (Hou et al., 2016; Beckham
147 and Pal, 2017; Liu et al., 2019a), which is a natural way to incorporate the order of the classes into
148 the loss term. However, OT tends to favor peaked output distributions instead of calibrated ones
149 (see Section 3.3). UNICORNN aims to benefit from both the ability to capture the order of classes
150 using OT as a training objective, and the generation of calibrated probability estimates through
accuracy-preserving calibration.

151 In summary, to the best of our knowledge, no existing work has successfully met all three fundamental
152 requirements for an effective ordinal regression model: (i) Ensuring unimodality in the output
153 distribution, ideally through architectural design; (ii) Aligning the training objective function with
154 the ordinal nature of the label space; (iii) Reflecting the model’s uncertainty in the decay of the
155 output probabilities, preferably with well-calibrated probabilities. These requirements led to the
156 development of UNICORNN, presented in this paper.

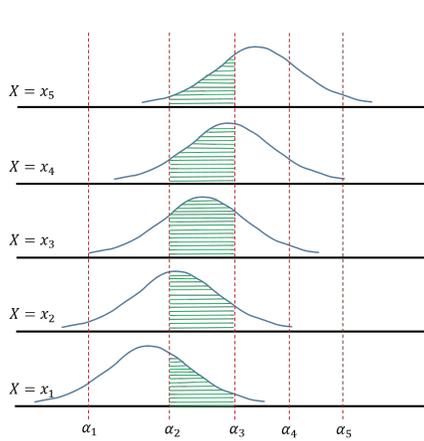


Figure 1: The proportional odds model. x_i is a realization of X . The standard logistic density is shifted by $\beta^T x_i$. The thresholds α_j define the bins which determine the probability predicted by the model to each class. For example, the green area defines the probability $\Pr(Y = 3)$ for given $\alpha_1, \dots, \alpha_5$ and β .

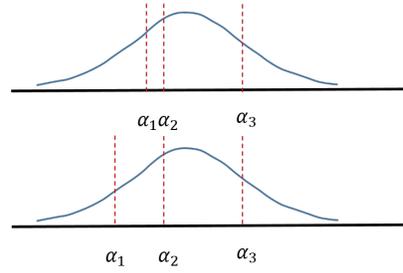


Figure 2: POM deficiencies. The plots illustrate two instances of POM, which highlight its two deficiencies: (1) POM does not always output unimodal probabilities: the first plot shows an example where the output probabilities are such that $\Pr(Y = 1) > \Pr(Y = 2) < \Pr(Y = 3) > \Pr(Y = 4)$. (2) The likelihood function of POM is invariant to the way the predicted probability mass of the incorrect classes is assigned: if the correct class is 3, both instances have the same likelihood, even though in the second instance, the probability mass assigned to neighboring class 2 is larger.

3 PRELIMINARIES

3.1 THE PROPORTIONAL ODDS MODEL

Let $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ be random variables, having joint probability \mathcal{P}_{XY} , where $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{1, \dots, k\}$, and $1, \dots, k$ are considered as symbols. Let \preceq be an order relation defined on \mathcal{Y} such that $1 \preceq \dots \preceq k$. The proportional odds model is parametrized by $\alpha \in \mathbb{R}^{|\mathcal{Y}|-1}$, $\beta \in \mathbb{R}^d$ and applies to data $\{(x_i, y_i)\}_{i=1}^n$, sampled i.i.d from \mathcal{P}_{XY} . Let ϵ be a logistic random variable (thus having a sigmoid cumulative distribution function $F(x) = \frac{1}{1+\exp(-x)}$), and let Z be a random variable defined as $Z = \beta^T X + \epsilon$. The entries of α are used to define the cumulative conditional probabilities via

$$\mathbb{P}(Y \preceq j | X = x) = \Pr(Z \leq \alpha_j) = F(\alpha_j - \beta^T x) \quad (1)$$

Similarly to logistic regression, this yields linear log-odds (logits), however, defined with respect to cumulative terms

$$\gamma_j \equiv \log \frac{\mathbb{P}(Y \preceq j | X = x)}{\mathbb{P}(Y \succ j | X = x)} = \alpha_j - \beta^T x$$

It is convenient to interpret equation 1 by viewing $\beta^T x$ as a factor that shifts the standard logistic density function, while the α_j terms are thresholds, with respect to which the cumulative probabilities are defined. This is depicted in Figure 1.

Let (x, y) be a realization of (X, Y) . The likelihood assigned by the model to (x, y) is

$$L(\alpha, \beta; (x, y)) = \mathbb{P}(Y = y | X = x; \alpha, \beta) = F(\alpha_y - \beta^T x) - F(\alpha_{y-1} - \beta^T x), \quad (2)$$

considering $\alpha_0 = -\infty$ and $\alpha_k = \infty$. The model is typically trained in a standard fashion by maximizing the log-likelihood function on the training data.

Despite its popularity, POM has two key limitations: First, the model's output probabilities are not necessarily unimodal (see Figure 2). Second, the likelihood function in equation 2 depends only on the probability the model assigns to the correct class y and is invariant to the way the remaining probability mass is assigned by the model. This ignores the order on the label set, and hence does not use important information that might be used to improve prediction quality, as depicted in Figure 2. This is also true of ordinal likelihood proposed in Chu et al. (2005). In addition, it is important to

216 mention that as the cross-entropy term is essentially equivalent to the model’s negative log-likelihood
 217 function, this invariance to the partition of the remaining mass over the incorrect classes is common
 218 to all models trained via cross-entropy minimization, as long as the target labels are one-hot. In
 219 Section 4 we will show how UNICORN overcomes these two limitations of POM.

221 3.2 PROBABILITY CALIBRATION

222 Probability Calibration refers to the alignment between the predicted probabilities generated by a
 223 classification model and the true, empirical probabilities observed in the data. For instance, it is
 224 anticipated that when a classification model assigns a probability of 0.8 to class i for a certain sample,
 225 approximately 80% of those samples would indeed belong to class i based on their ground truth
 226 labels. Formally, the concept of calibration can be defined as:

$$228 \mathbb{P}[y = i | p_i(x) = q] = q, \quad \forall q \in [0, 1], i \in [k] \quad (3)$$

229 Here, k denotes the number of classes, x and y represent the model input and the ground truth
 230 class, respectively, and $p_i(x)$ signifies the model’s output probability for class i given input x . The
 231 probability is taken over the joint distribution of x, y .

232 Alternatively, if we denote y as a one-hot vector indicating the ground truth label, we can express an
 233 equivalent definition to equation 3: A classification model is calibrated when $\mathbb{E}_{x,y}[y|p(x)] = p(x)$. In
 234 practice, $p(x)$ is a one-to-one function, hence we can refine the definition further: $\mathbb{E}_{x,y}[y|x] = p(x)$.

235 To achieve calibration, minimizing the squared L_2 norm on the difference between the two terms
 236 is desirable. However, given the unknown *real* joint distribution x, y , we can only approximate
 237 $\mathbb{E}_{x,y}[y|x]$. One approach is to use a Monte Carlo approximation with one sample, yielding the
 238 approximation y , i.e., minimize the following:

$$240 BS(y, p(x)) = \|y - p(x)\|_2^2 \quad (4)$$

241 This term is known as the Brier Score (BS) (Brier, 1950). BS is a well-known Proper Scoring Rule
 242 (Dawid and Musio, 2014), which evaluates the accuracy of probabilistic predictions. Proper scoring
 243 rules are maximized when the probabilistic forecast matches the true probability distribution of
 244 outcomes. Therefore, we will use BS as a training objective to calibrate the model predictions.

245 **Accuracy-Preserving Calibration** An accuracy-preserving calibration method is a technique that
 246 adjusts the probability outputs of a pre-trained model to improve their calibration, without affecting
 247 the model’s accuracy. A popular method is Temperature Scaling (TS) (Guo et al., 2017), which learns
 248 a single temperature parameter that is used to rescale the model’s logits before applying the softmax
 249 activation, adjusting the confidence predictions. Adaptive Temperature Scaling (ATS) (Balanya
 250 et al., 2024) extends TS such that instead of a single temperature parameter, it learns a mapping
 251 $x \mapsto T(x)$ that adaptively scales the logits based on the input x . As part of UNICORN we define
 252 an accuracy-preserving calibration method with similarity to TS and ATS which is trained via Brier
 253 Score (Section 4.3).

256 3.3 OPTIMAL TRANSPORT

257 Let M be a finite metric space with moving cost metric $c(x, y)$ between elements $x, y \in M$, and
 258 let p, q be probability mass functions on M . The optimal transport (OT) or 1-Wasserstein distance
 259 between q and p is defined as:

$$262 OT(p, q) = \inf_{\gamma \in \Gamma} \int_{M \times M} c(x, y) d\gamma(x, y), \quad (5)$$

263 Where Γ is the set of joint probabilities on $M \times M$ with marginals q and p , and c specifies moving
 264 costs between elements of M . This computes the optimal way to transport q into p . When q is a
 265 Dirac (one-hot), OT simplifies to:

$$268 OT(p, q) = \sum_{i=1}^k p_i c(i, j), \quad (6)$$

Where j is the correct class and k is the number of classes. With model outputs p and one-hot target q , this loss is differentiable w.r.t p . The cost metric c can encode domain knowledge. For ordered classes $M = 1, \dots, k$, a natural cost is $c(i, j) = |i - j|^m$ for some $m \geq 1$.

Importantly, one drawback of OT as a loss function, is its tendency to prioritize peaked distributions over the actual probabilities. For instance, Let x represent a sample data point, y denote the class label associated with x , and q be the one-hot encoded vector representation of y . Consider the conditional probabilities $\mathbb{P}[y = 1|x] = 0.25, \mathbb{P}[y = 2|x] = 0.5, \mathbb{P}[y = 3|x] = 0.25$. Now, suppose there are two model outputs: $p_1 = [0.25, 0.5, 0.25]$ and $p_2 = [0, 1, 0]$. To achieve calibration, the model should ideally output p_1 as it equals the conditional probability distribution $y|x$. However, it is notable that for the cost metric $c(i, j) = |i - j|$, we find that $OT(p_1, q) = 1$ with a combined probability of $0.25 + 0.25 = 0.5$, and $OT(p_2, q) = 0.5$ with a probability of 0.5 . Consequently, $\mathbb{E}_{y|x}[OT(p_1, q)] = 0.75$, and similarly, $\mathbb{E}_{y|x}[OT(p_2, q)] = 0.5$. This observation indicates that OT tends to favor peaked distributions over calibrated ones. UNICORNN addresses this challenge, as described in Section 4.3.

4 UNICORNN

In this section we describe our novel mechanism for UNImodal Calibrated Ordinal Regression Neural Network, UNICORNN an approach for architectural-based generation of unimodal output probability distributions, as well as accuracy-preserving probability calibration.

4.1 RATIONAL

Achieving unimodality directly via architectural design has a major advantage since the output probabilities are guaranteed to be unimodal for every input instance, as is also the case for the mechanism proposed in Beckham and Pal (2017). However, UNICORNN employs the truncated normal distribution, depending on two parameters (μ, σ) , which influence the location of the mode and the decay of the probability mass. This adds flexibility to the shape of the output probability vector, compared to the mechanism in Beckham and Pal (2017) where a single parameter determines both the mode and the decay. For determining the mode of the distribution, a map $x \mapsto \mu$ is learned via OT using equation 6. However, as mentioned in Section 3.3, OT tends to favor peaked distributions over calibrated ones. To address this issue, we adopt an accuracy-preserving calibration strategy where, subsequent to learning a map $x \mapsto \mu$, the model further learns a map $x \mapsto \sigma$ by optimizing the Brier Score using equation 4, while maintaining the previously learned map $x \mapsto \mu$ fixed.

4.2 UNIMODAL OUTPUT PROBABILITIES GENERATION

Inspired by POM, we utilize thresholds to define bins, so that the total mass inside each bin is the output probability of the corresponding class. However, we observe that the lack of unimodality of POM can be fixed by letting the bins be of equal length and remain fixed during training.

Therefore, instead of learning the thresholds, during training two maps $x \mapsto \mu, x \mapsto \sigma$ are learned, where μ is a location parameter, and σ is a scale parameter. Both define a truncated normal distribution $\mathcal{N}_{\text{trunc}}(\mu, \sigma^2, -1, 1)$ (a normal distribution which having the same density as the normal density on $[-1, 1]$, normalized to have a unit integral, and is zero outside this interval), from which the output probabilities are derived.

Formally, we divide the range $[-1, 1]$ to k equal bins similarly to da Costa et al. (2008), where k is the number of classes, defined by $-1 = \alpha_0, \alpha_1, \dots, \alpha_k = 1$, so that $\alpha_i - \alpha_{i-1} = \frac{2}{k}$. The probabilities are given by

$$p_i(x) = \mathbb{P}(Y = i|X = x) = F_{\mu(x), \sigma(x)}(\alpha_i) - F_{\mu(x), \sigma(x)}(\alpha_{i-1}), \quad (7)$$

where $F_{\mu, \sigma}(\cdot)$ is the $\mathcal{N}_{\text{trunc}}(\mu, \sigma^2, -1, 1)$ cumulative distribution function, and note that μ, σ are in fact functions of the input instance x .

To compensate for the fact that the probability-generating mechanism depends on fewer parameters than POM (2 for the former, $d + k - 1$ for the latter), the maps $x \mapsto \mu, x \mapsto \sigma$ are expressed via two deep neural networks (DNNs) which share a common backbone model, hence can represent a complex

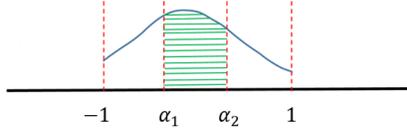


Figure 3: Generation of unimodal output probabilities for $k = 3$ classes. An input x is mapped to a (μ, σ) pair, which defines a truncated normal distribution $\mathcal{N}_{\text{trunc}}(\mu, \sigma^2, -1, 1)$ over the real line. The output probabilities are proportional to the mass in the bins, which are of equal length. The green area equals to probability $p_2(x)$, corresponding to $\mathbb{P}(Y = 2|X = x)$.

nonlinear relation. Our proposed mechanism for the generation of unimodal output probabilities is depicted in Figure 3.

The following lemma, proved in Appendix A, establishes that the model output probabilities are indeed unimodal.

Lemma 1. *Let $x \in \mathbb{R}^d$ be an input to the model, which is mapped to $\mu = \mu(x), \sigma = \sigma(x)$. Let p_1, \dots, p_k be the model output probabilities, generated via equation 7. Then p_1, \dots, p_k define a unimodal multinomial random variable.*

4.3 ACCURACY-PRESERVING CALIBRATION IN UNICORNN

UNICORNN introduces a re-training procedure for $x \mapsto \sigma$ mapping while preserving the already trained mapping $x \mapsto \mu$. The learning of $x \mapsto \sigma$ is done by minimizing the Brier Score loss in equation 4, which ensures well-calibrated probabilities and preserves the model accuracy. Since the mapping $x \mapsto \sigma$ controls the decay of the generated probabilities and the mapping $x \mapsto \mu$ which controls the mode remains unchanged, the accuracy is not affected by the calibration, resulting in an accuracy-preserving calibration. We state it in the following lemma (proved in Appendix B).

Lemma 2. *Let $x \in \mathbb{R}^d$ be an input to the model, mapped to $\mu = \mu(x), \sigma_1 = \sigma_1(x)$. Let $\sigma_2 = \sigma_2(x)$ be a re-trained mapping via minimization of loss using equation 4. Let $p_1^{\sigma_1}, \dots, p_k^{\sigma_1}$ and $p_1^{\sigma_2}, \dots, p_k^{\sigma_2}$ be the model output probabilities generated using (μ, σ_1) and (μ, σ_2) , respectively, via equation 7. Then:*

$$\operatorname{argmax}_{1 \leq i \leq k} p_i^{\sigma_1} = \operatorname{argmax}_{1 \leq i \leq k} p_i^{\sigma_2}.$$

4.4 TRAINING PROCEDURE

To summarize, UNICORNN’s training process consists of two distinct phases, as detailed in Algorithms 1, 2. In the first phase, described in Algorithm 1, the parameters θ_μ and θ_σ of $\mu(\cdot)$ and $\sigma(\cdot)$, respectively, along with the backbone parameters ϕ , are jointly optimized using Gradient Descent to minimize the OT loss defined in Eq. 6.

In the second phase, outlined in Algorithm 2, the parameters θ_μ and ϕ are kept fixed, while θ_σ is further optimized via Gradient Descent to minimize the Brier Score (BS) defined in Eq. 4.

Algorithm 1 Optimal Transport-based Training

Input: Training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, number of classes k , batch size b , number of epochs T

Output: Parameters $\theta_\mu, \theta_\sigma$ of $\mu(\cdot)$ and $\sigma(\cdot)$, respectively, and backbone parameters ϕ

- 1: Initialize parameters $\theta_\mu, \theta_\sigma, \phi$
 - 2: **for** epoch $t = 1$ to T **do**
 - 3: **for** each batch $\mathcal{B} \subseteq \mathcal{D}$ of size b **do**
 - 4: Compute $\mu_{\phi, \theta_\mu}(x_i)$ and $\sigma_{\phi, \theta_\sigma}(x_i)$ for all $x_i \in \mathcal{B}$ and calculate $p(x_i)$ using Eq. 7
 - 5: Compute the loss $\mathcal{L}_{\text{OT}}(\mathcal{B}) = \frac{1}{b} \sum_{(x_i, y_i) \in \mathcal{B}} \text{OT}(p(x_i), y_i)$ using Eq. 6
 - 6: Update $\theta_\mu, \theta_\sigma, \phi$ via Gradient Descent
 - 7: **end for**
 - 8: **end for**
 - 9: **return** $\theta_\mu, \theta_\sigma, \phi$
-

Algorithm 2 Accuracy Preserving Calibration

Input: Training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, number of classes k , batch size b , number of epochs T , parameters θ_μ of $\mu(\cdot)$ and backbone parameters ϕ

Output: Parameters θ_σ of $\sigma(\cdot)$

- 1: **for** epoch $t = 1$ to T **do**
- 2: **for** each batch $\mathcal{B} \subseteq \mathcal{D}$ of size b **do**
- 3: Compute $\mu_{\phi, \theta_\mu}(x_i)$ and $\sigma_{\phi, \theta_\sigma}(x_i)$ for all $x_i \in \mathcal{B}$ and calculate $p(x_i)$ using Eq. 7
- 4: Compute the loss $\mathcal{L}_{BS}(\mathcal{B}) = \frac{1}{b} \sum_{(x_i, y_i) \in \mathcal{B}} BS(p(x_i), y_i)$ using Eq. 4
- 5: Update **only** θ_σ via Gradient Descent
- 6: **end for**
- 7: **end for**
- 8: **return** θ_σ

5 EXPERIMENTAL RESULTS

5.1 DATASETS

We evaluate UNICORNN on six real-world benchmark image datasets, involving various ordinal regression tasks: age-detection (Adience Eidinger et al. (2014), FG-Net Fu et al. (2014), AAF Cheng et al. (2019)), facial beauty prediction (SCUT-FBP5500 Liang et al. (2018)), bio-medical image classification (Retina-MNIST Yang et al. (2021)) and image aesthetics estimation (EVA Kang et al. (2020)). A more detailed description of the datasets appears in Appendix D. Some examples from the Adience and Retina-MNIST datasets are shown in Figure 4.

5.2 BENCHMARK

We compare UNICORNN to five recently presented approaches for deep ordinal regression, with unimodal output probabilities and to a deep learning approach of POM:

DLDL (Gao et al., 2017), an approach utilizing soft labels, generated using squared exponentially decaying distributions, trained using Kullback-Leibler divergence minimization (equivalent to cross-entropy minimization).

SORD (Diaz and Marathe, 2019), an approach utilizing soft labels, generated using linear exponentially decaying distributions, trained using Kullback-Leibler divergence minimization.

Beckham and Pal (2017), an architectural-based approach in which unimodal output probabilities are generated using the binomial distribution (single-learned parameter), trained using optimal transport loss.

Liu et al. (2019a), an approach utilizing soft labels, created as a mixture of Dirac, uniform, and linear exponentially decaying distributions, trained using optimal transport loss.

POM (McCullagh, 1980), a variant of the POM, incorporating a deep learning model with a POM layer integrated on top, trained using cross-entropy loss.

UnimodalNet (Cardoso et al., 2023), a non-parametric architectural-based approach for unimodality, trained using cross-entropy loss.

To perform a fair comparison, we implemented all methods, using the same image transformations, backbone CNN and training procedures, so that the methods differ only in their output layer architectures and loss functions. We performed 5 independent trials, using the same train-validation-test splits for all methods. Additional technical details can be found in Appendix E. For reproducibility, the supplementary material contains code reproducing the results reported in this section.

5.3 EVALUATION METRICS

We report several commonly-used evaluation metrics for ordinal regression tasks: Mean Absolute Error (MAE), One-Off Accuracy (OOA), Spearman correlation, Quadratic Weighted Kappa (QWK), as well as the percentage of test examples with unimodal predicted output probabilities.

432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485

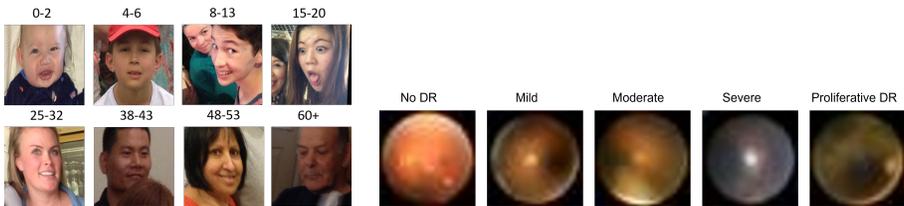


Figure 4: Left: Examples from the Adience dataset. The age category is indicated above each image. Right: Examples from the Retina mnist dataset. Diabetic Retinopathy classes are indicated above.

In addition, following the discussion in section 3.2, we evaluate the model’s probability calibration using the well known Expected Calibration Error (ECE). ECE quantifies the discrepancy between predicted probabilities and actual outcomes, measuring how well the predicted probabilities of a model align with the true likelihood of the predicted events. To compute ECE, the $[0, 1]$ interval is first divided into a set of b equal-length bins. ECE is defined as:

$$ECE = \sum_{i=1}^b \frac{|B_i|}{|X|} |\text{Acc}(B_i) - \text{Conf}(B_i)|,$$

where b is the number of probability bins (in all of our experiments $b = 10$), $\text{Acc}(B_i)$ is the empirical accuracy in bin B_i and $\text{Conf}(B_i)$ is the average predicted probability in bin B_i . An ECE of 0 indicates perfect calibration, while higher values signify miscalibration, with the model being over-confident or under-confident.

5.4 RESULTS ON REAL WORLD DATASETS

Table 1 shows the test results of each method on the six benchmark datasets. As can be seen, UNICORNN performs at least on par and often better than the compared baselines, in a fairly consistent manner, across the various datasets and evaluation metrics, specifically with respect to the MAE, where UNICORNN outperforms on all the six datasets. In addition, observe that only UNICORNN, UnimodalNet and Beckham and Pal (2017) output unimodal probabilities, via architectural design, while the other baselines, trained using soft targets, do not always output unimodal probabilities. Yet, unlike UnimodalNet and Beckham and Pal (2017) which tend to produce poorly calibrated probability estimates, UNICORNN outperforms all baselines in terms of probability calibration across five datasets, while on the EVA dataset, the difference from SORD is negligible.

5.5 ABLATION STUDY

In this section, we analyzed the impact of the calibration phase introduced in Section 4.3 on the ECE values of UNICORNN. Table 2 shows the ECE with and without the calibration phase on the Adience, EVA, AAF, and Retina MNIST datasets. Without calibration, the ECE is higher, indicating that the model’s outputs are not well-calibrated due to the OT properties discussed in Section 3.3. This highlights the importance of incorporating the calibration step to accurately reflect the model’s confidence in its predictions.

6 CONCLUSION

In this manuscript, we identify several issues with current deep ordinal regression methods, including the potential for OT to result in poor probability calibration. We therefore presented UNICORNN, an approach for deep ordinal regression, inspired by the proportional odds model. UNICORNN utilizes an architectural mechanism for the generation of unimodal output probabilities, trained using OT objective and calibrated using an accuracy-preserving calibration process which encourages uncertainty awareness. We demonstrated that while performing on par with and often better than other recently proposed approaches for ordinal regression, the presented method enjoys the benefits of *guaranteed* unimodal and well-calibrated output probabilities.

Table 1: Performance of various methods on real world datasets, in a mean \pm std format

Dataset	Method	MAE \downarrow	OOA \uparrow	Spearman \uparrow	QWK \uparrow	% Unimodal \uparrow	ECE ($b = 10$) \downarrow
Adience	Beckham and Pal	.5 \pm .06	.93 \pm .01	.9 \pm .01	.91 \pm .02	1 \pm 0	.23 \pm .05
	Liu et al.	.47 \pm .05	.94 \pm .01	.89 \pm .02	.9 \pm .02	.47 \pm .04	.21 \pm .03
	DLDL	.49 \pm .06	.93 \pm .01	.88 \pm .02	.89 \pm .03	.62 \pm .08	.42 \pm .04
	SORD	.47 \pm .06	.94 \pm .01	.89 \pm .02	.9 \pm .02	.99 \pm .003	.14 \pm .03
	POM	.48 \pm .06	.94 \pm .01	.89 \pm .03	.91 \pm .03	.81 \pm .04	.32 \pm .04
	UnimodalNet	.49 \pm .06	.93 \pm .01	.88 \pm .02	.90 \pm .03	1 \pm 0	.34 \pm .04
	UNICORN	.46 \pm .05	.95 \pm .01	.9 \pm .02	.91 \pm .02	1 \pm 0	.07 \pm .03
Retina MNIST	Beckham and Pal	.8 \pm .02	.79 \pm .01	.58 \pm .02	.55 \pm .02	1 \pm 0	.17 \pm .01
	Liu et al.	.68 \pm .02	.82 \pm .01	.61 \pm .02	.58 \pm .02	.72 \pm .03	.27 \pm .01
	DLDL	.72 \pm .02	.81 \pm .01	.59 \pm .02	.58 \pm .01	.98 \pm .01	.29 \pm .01
	SORD	.75 \pm .02	.78 \pm .01	.58 \pm .007	.57 \pm .007	.87 \pm .03	.11 \pm .01
	POM	.83 \pm .02	.76 \pm .01	.53 \pm .02	.53 \pm .02	.43 \pm .03	.17 \pm .01
	UnimodalNet	.74 \pm .02	.8 \pm .01	.59 \pm .01	.58 \pm .01	1 \pm 0	.14 \pm .01
	UNICORN	.67 \pm .009	.82 \pm .007	.61 \pm .01	.59 \pm .01	1 \pm 0	.06 \pm .01
FG-NET	Beckham and Pal	.44 \pm .06	.95 \pm .03	.80 \pm .04	.79 \pm .03	1 \pm 0	.16 \pm .02
	Liu et al.	.32 \pm .08	.96 \pm .02	.86 \pm .03	.84 \pm .04	.01 \pm .01	.08 \pm .02
	DLDL	.4 \pm .09	.96 \pm .02	.81 \pm .03	.81 \pm .03	.12 \pm .04	.49 \pm .06
	SORD	.36 \pm .08	.97 \pm .02	.84 \pm .05	.82 \pm .04	.99 \pm .01	.26 \pm .05
	POM	.33 \pm .06	.97 \pm .02	.84 \pm .04	.85 \pm .04	.41 \pm .06	.21 \pm .03
	UnimodalNet	.36 \pm .07	.97 \pm .01	.84 \pm .02	.84 \pm .03	1 \pm 0	.25 \pm .04
	UNICORN	.31 \pm .08	.99 \pm .02	.87 \pm .03	.88 \pm .01	1 \pm 0	.07 \pm .02
AAF	Beckham and Pal	.61 \pm .13	.91 \pm .05	.81 \pm .03	.8 \pm .05	1 \pm 0	.16 \pm .04
	Liu et al.	.43 \pm .01	.97 \pm .005	.82 \pm .01	.85 \pm .01	.7 \pm .17	.22 \pm .01
	DLDL	.54 \pm .02	.95 \pm .01	.77 \pm .01	.77 \pm .02	.95 \pm .02	.32 \pm .02
	SORD	.44 \pm .02	.96 \pm .01	.82 \pm .01	.84 \pm .01	1 \pm 0	.14 \pm .02
	POM	.45 \pm .01	.96 \pm .01	.80 \pm .01	.83 \pm .01	.48 \pm .07	.28 \pm .02
	UnimodalNet	.45 \pm .01	.96 \pm .01	.80 \pm .01	.83 \pm .01	1 \pm 0	.30 \pm .01
	UNICORN	.42 \pm .01	.97 \pm .005	.83 \pm .01	.85 \pm .005	1 \pm 0	.03 \pm .01
SCUT-FBP5500	Beckham and Pal	.62 \pm .12	.92 \pm .04	.84 \pm .02	.81 \pm .05	1 \pm 0	.19 \pm .08
	Liu et al.	.54 \pm .03	.95 \pm .01	.83 \pm .01	.82 \pm .01	.18 \pm .14	.27 \pm .02
	DLDL	.68 \pm .05	.9 \pm .01	.79 \pm .02	.73 \pm .02	.86 \pm .08	.28 \pm .04
	SORD	.57 \pm .03	.94 \pm .01	.83 \pm .01	.81 \pm .01	.99 \pm .005	.09 \pm .02
	POM	.53 \pm .03	.95 \pm .01	.83 \pm .01	.83 \pm .01	.28 \pm .04	.33 \pm .03
	UnimodalNet	.52 \pm .02	.95 \pm .01	.82 \pm .01	.83 \pm .01	1 \pm 0	.35 \pm .01
	UNICORN	.47 \pm .02	.96 \pm .01	.85 \pm .01	.85 \pm .01	1 \pm 0	.06 \pm .02
EVA	Beckham and Pal	.61 \pm .03	.93 \pm .01	.6 \pm .01	.6 \pm .02	1 \pm 0	.11 \pm .01
	Liu et al.	.65 \pm .02	.91 \pm .02	.53 \pm .01	.52 \pm .02	.72 \pm .05	.31 \pm .01
	DLDL	.66 \pm .03	.91 \pm .01	.53 \pm .02	.52 \pm .02	.98 \pm .01	.2 \pm .01
	SORD	.6 \pm .02	.93 \pm .01	.58 \pm .02	.57 \pm .02	1 \pm 0	.07 \pm .02
	POM	.65 \pm .03	.91 \pm .01	.53 \pm .02	.53 \pm .03	.89 \pm .02	.45 \pm .02
	UnimodalNet	.66 \pm .03	.91 \pm .01	.52 \pm .02	.52 \pm .02	1 \pm 0	.45 \pm .02
	UNICORN	.57 \pm .01	.94 \pm .005	.6 \pm .01	.58 \pm .01	1 \pm 0	.08 \pm .02

Table 2: The effect of UNICORN’s calibration phase on the ECE for the Adience and Retina MNIST datasets.

Dataset	Post-hoc Calibration	ECE ($b = 10$) \downarrow
Retina MNIST	yes	.06 \pm .01
	no	.38 \pm .007
Adience	yes	.07 \pm .03
	no	.18 \pm .03
AAF	yes	.03 \pm .01
	no	.28 \pm .01
EVA	yes	.08 \pm .02
	no	.32 \pm .02

Reproducibility Statement All technical details of the experiments are provided in Section E, also the code will be made publicly available upon acceptance of this work.

REFERENCES

Balanya, S. A., Maroñas, J., and Ramos, D. (2024). Adaptive temperature scaling for robust calibration of deep neural networks. *Neural Computing and Applications*, pages 1–23.

- 540 Beckham, C. and Pal, C. (2017). Unimodal probability distributions for deep ordinal classification.
541 In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages
542 411–419.
- 543 Belharbi, S., Ben Ayed, I., McCaffrey, L., and Granger, E. (2019). Non-parametric uni-modality
544 constraints for deep ordinal classification. *arXiv*, pages arXiv–1911.
- 545 Berg, A., Oskarsson, M., and O’Connor, M. (2020). Deep ordinal regression with label diversity.
546 *arXiv preprint arXiv:2006.15864*.
- 547 Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly weather*
548 *review*, 78(1):1–3.
- 549 Cao, W., Mirjalili, V., and Raschka, S. (2020). Rank consistent ordinal regression for neural networks
550 with application to age estimation. *Pattern Recognition Letters*, 140:325–331.
- 551 Cardoso, J. S., Cruz, R., and Albuquerque, T. (2023). Unimodal distributions for ordinal regression.
552 *arXiv preprint arXiv:2303.04547*.
- 553 Cheng, J., Li, Y., Wang, J., Yu, L., and Wang, S. (2019). Exploiting effective facial patches for robust
554 gender recognition. *Tsinghua Science and Technology*, 24(3):333–345.
- 555 Cheng, J., Wang, Z., and Pollastri, G. (2008). A neural network approach to ordinal regression.
556 In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on*
557 *Computational Intelligence)*, pages 1279–1284. IEEE.
- 558 Chu, W., Ghahramani, Z., and Williams, C. K. (2005). Gaussian processes for ordinal regression.
559 *Journal of machine learning research*, 6(7).
- 560 da Costa, J. F. P., Alonso, H., and Cardoso, J. S. (2008). The unimodal model for the classification of
561 ordinal data. *Neural Networks*, 21(1):78–91.
- 562 Dawid, A. P. and Musio, M. (2014). Theory and applications of proper scoring rules. *Metron*,
563 72(2):169–183.
- 564 De Vente, C., Vos, P., Hosseinzadeh, M., Pluim, J., and Veta, M. (2020). Deep learning regression
565 for prostate cancer detection and grading in bi-parametric mri. *IEEE Transactions on Biomedical*
566 *Engineering*, 68(2):374–383.
- 567 Diaz, R. and Marathe, A. (2019). Soft labels for ordinal regression. In *Proceedings of the IEEE*
568 *Conference on Computer Vision and Pattern Recognition*, pages 4738–4747.
- 569 Eidinger, E., Enbar, R., and Hassner, T. (2014). Age and gender estimation of unfiltered faces. *IEEE*
570 *Transactions on Information Forensics and Security*, 9(12):2170–2179.
- 571 Esaki, Y., Nakamura, A., Kawano, K., Tokuhisa, R., and Kutsuna, T. (2024). Accuracy-preserving
572 calibration via statistical modeling on probability simplex. *arXiv preprint arXiv:2402.13765*.
- 573 Fu, H., Gong, M., Wang, C., Batmanghelich, K., and Tao, D. (2018). Deep ordinal regression network
574 for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and*
575 *Pattern Recognition*, pages 2002–2011.
- 576 Fu, Y., Hospedales, T. M., Xiang, T., Gong, S., and Yao, Y. (2014). Interestingness prediction by
577 robust learning to rank. In *European conference on computer vision*, pages 488–503. Springer.
- 578 Gao, B.-B., Xing, C., Xie, C.-W., Wu, J., and Geng, X. (2017). Deep label distribution learning with
579 label ambiguity. *IEEE Transactions on Image Processing*, 26(6):2825–2838.
- 580 Garg, B. and Manwani, N. (2020). Robust deep ordinal regression under label noise. In *Asian*
581 *conference on machine learning*, pages 782–796. PMLR.
- 582 Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks.
583 In *International conference on machine learning*, pages 1321–1330. PMLR.
- 584
- 585
- 586
- 587
- 588
- 589
- 590
- 591
- 592
- 593

- 594 Hou, L., Yu, C.-P., and Samaras, D. (2016). Squared earth mover’s distance-based loss for training
595 deep neural networks. *arXiv preprint arXiv:1611.05916*.
596
- 597 Kang, C., Valenzise, G., and Dufaux, F. (2020). Eva: An explainable visual aesthetics dataset. In
598 *Joint Workshop on Aesthetic and Technical Quality Assessment of Multimedia and Media Analytics*
599 *for Societal Trends*, pages 5–13.
- 600 Kim, K.-j. and Ahn, H. (2012). A corporate credit rating model using multi-class support vector
601 machines with an ordinal pairwise partitioning approach. *Computers & Operations Research*,
602 39(8):1800–1811.
603
- 604 Kook, L., Herzog, L., Hothorn, T., Dürr, O., and Sick, B. (2020). Deep and interpretable regression
605 models for ordinal outcomes. *arXiv preprint arXiv:2010.08376*.
- 606 Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty
607 estimation using deep ensembles. *Advances in neural information processing systems*, 30.
608
- 609 Li, Q., Wang, J., Yao, Z., Li, Y., Yang, P., Yan, J., Wang, C., and Pu, S. (2022). Unimodal-
610 concentrated loss: Fully adaptive label distribution learning for ordinal regression. In *Proceedings*
611 *of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20513–20522.
- 612 Li, W., Huang, X., Lu, J., Feng, J., and Zhou, J. (2021). Learning probabilistic ordinal embeddings
613 for uncertainty-aware regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision*
614 *and Pattern Recognition*, pages 13896–13905.
615
- 616 Liang, L., Lin, L., Jin, L., Xie, D., and Li, M. (2018). Scut-fbp5500: A diverse benchmark dataset for
617 multi-paradigm facial beauty prediction.
- 618 Liu, X., Fan, F., Kong, L., Diao, Z., Xie, W., Lu, J., and You, J. (2020). Unimodal regularized neuron
619 stick-breaking for ordinal classification. *Neurocomputing*.
- 620
- 621 Liu, X., Han, X., Qiao, Y., Ge, Y., Li, S., and Lu, J. (2019a). Unimodal-uniform constrained
622 wasserstein training for medical diagnosis. In *Proceedings of the IEEE International Conference*
623 *on Computer Vision Workshops*, pages 0–0.
- 624
- 625 Liu, X., Zou, Y., Song, Y., Yang, C., You, J., and K Vijaya Kumar, B. (2018a). Ordinal regression
626 with neuron stick-breaking for medical diagnosis. In *Proceedings of the European Conference on*
627 *Computer Vision (ECCV)*, pages 0–0.
- 628
- 629 Liu, Y., Kong, A. W.-K., and Goh, C. K. (2017). Deep ordinal regression based on data relationship
630 for small datasets. In *IJCAI*, pages 2372–2378.
- 631
- 632 Liu, Y., Wai Kin Kong, A., and Keong Goh, C. (2018b). A constrained deep neural network for
633 ordinal regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern*
634 *Recognition*, pages 831–839.
- 635
- 636 Liu, Y., Wang, F., and Kong, A. W. K. (2019b). Probabilistic deep ordinal regression based on
637 gaussian processes. In *Proceedings of the IEEE International Conference on Computer Vision*,
638 pages 5301–5309.
- 639
- 640 McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society:*
641 *Series B (Methodological)*, 42(2):109–127.
- 642
- 643 Niu, Z., Zhou, M., Wang, L., Gao, X., and Hua, G. (2016). Ordinal regression with multiple output
644 cnn for age estimation. In *Proceedings of the IEEE conference on computer vision and pattern*
645 *recognition*, pages 4920–4928.
- 646
- 647 Pan, H., Han, H., Shan, S., and Chen, X. (2018). Mean-variance loss for deep age estimation from a
648 face. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages
649 5285–5294.
- 650
- 651 Shi, X., Cao, W., and Raschka, S. (2021). Deep neural networks for rank-consistent ordinal regression
652 based on conditional probabilities. *arXiv preprint arXiv:2111.08851*.

648 Shi, X., Cao, W., and Raschka, S. (2023). Deep neural networks for rank-consistent ordinal regression
649 based on conditional probabilities. *Pattern Analysis and Applications*, 26(3):941–955.
650

651 Vargas, V. M., Gutiérrez, P. A., and Hervás-Martínez, C. (2020). Cumulative link models for deep
652 ordinal classification. *Neurocomputing*.

653 Vishnu, T., Malhotra, P., Vig, L., Shroff, G., et al. (2019). Data-driven prognostics with predictive
654 uncertainty estimation using ensemble of deep ordinal regression models. *International Journal of*
655 *Prognostics and Health Management*, 10(4).
656

657 Wienholt, P., Hermans, A., Khader, F., Puladi, B., Leibe, B., Kuhl, C., Nebelung, S., and Truhn, D.
658 (2024). An ordinal regression framework for a deep learning based severity assessment for chest
659 radiographs. *arXiv preprint arXiv:2402.05685*.

660 Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., and Ni, B. (2021). Medmnist v2: A
661 large-scale lightweight benchmark for 2d and 3d biomedical image classification. *arXiv preprint*
662 *arXiv:2110.14795*.
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701

A PROOF OF LEMMA 1

Proof. Let p_i, p_{i+1} be the output probabilities of two adjacent classes, and let $-1 = \alpha_0, \dots, \alpha_k = 1$ be the thresholds. We will show that (i) if $\mu \leq \alpha_{i-1}$ then $p_i \geq p_{i+1}$. Symmetrical argument will then imply that if $\mu \geq \alpha_{i+1}$ then $p_{i+1} \geq p_i$ (ii) if $\mu \in (\alpha_{i-1}, \alpha_i)$ then $p_i > p_{i+1}$, whenever the latter exists. Similarly, this would imply that $p_i > p_{i-1}$. Together, (i) and (ii) will prove the statement of the lemma.

Denote by f the density of the $\mathcal{N}_{\text{trunc}}(\mu, \sigma^2, -1, 1)$ distribution. To prove (i) observe that $p_i > \frac{2f(\alpha_i)}{k} > p_{i+1}$.

To prove (ii), divide the i 'th bin to two sub-bins $B_{i,1}, B_{i,2}$, of lengths $a = \mu - \alpha_{i-1}$ and $b = \alpha_i - \mu$, respectively. Similarly, divide the $i + 1$ 'th bin to two bins $B_{i+1,1}, B_{i+1,2}$ length b and a , respectively. Then from (i)

$$\int_{B_{i,2}} f(x)dx > \int_{B_{i+1,1}} f(x)dx. \quad (8)$$

In addition, observe that

$$\begin{aligned} \int_{B_{i,1}} f(x)dx &= \int_0^a f(\mu + x)dx \\ &> \int_0^a f(\mu + 2b + x)dx \\ &= \int_{B_{i+1,2}} f(x)dx. \end{aligned} \quad (9)$$

Adding up equation 8 and equation 9, we obtain $p_i > p_{i+1}$. Apart from just being unimodal, we also proved that if $\mu \in (\alpha_{i-1}, \alpha_i)$ then class i is the predicted class by the model.

□

B PROOF OF LEMMA 2

Proof. Let $-1 = \alpha_0 < \alpha_1 < \dots < \alpha_k = 1$ be the thresholds. In the proof of Lemma 1 (Appendix A), we demonstrated that for any $i \in \{1, \dots, k\}$, if $\mu \in (\alpha_{i-1}, \alpha_i)$, then class i is the predicted class by the model.

Given that the two sets of parameters of the truncated normal distribution, (μ, σ_1^2) and (μ, σ_2^2) share the same mean μ and the thresholds $\alpha_0, \dots, \alpha_k$ are constant, the predicted class i remains the same for both sets of parameters regardless of the variances σ_1^2 and σ_2^2 .

□

C ADDITIONAL EXPERIMENTS

C.1 COMPARING UNICORNN AGAINST CORN

Table C.1 describes the results of two experiments of UNICORNN against CORN² (Shi et al., 2023) on the Adience and EVA datasets. The technical details of the experiments are the same as the ones that appear in Appendix E. As can be seen, UNICORNN outperforms CORN on both datasets with lower MAE scores..

C.2 COMPARING UNICORNN AGAINST CORN ON NEW DATASETS

In Table C.2 we provide the MAE results of our method trained on AFAD and Fireman dataset additionally to the results from Shi et al. (2021) for CE-NN, OR-NN, CORAL and CORN. The results for these baselines were borrowed from Shi et al. (2021). We train our model with the same

²https://github.com/Raschka-research-group/corn-ordinal-neuralnet/blob/main/model-code/simple-scripts/mlp_corn.py

Table 3: UNICORNN vs CORN.

Method	Dataset	MAE ↓
Adience	CORN	.52 ± .07
	UNICORNN	.46 ± .05
EVA	CORN	.67 ± .02
	UNICORNN	.57 ± .01

backbone as in Shi et al. (2021), Resnet-34 for AFAD dataset, and with two-layer MLP with hidden dimension of size 300 for Fireman dataset. Both datasets are balanced and the train, validation and test splits are the same as in Shi et al. (2021). We train our method 5 times with different random initialization seeds. Our method outperforms the baselines on both datasets.

Table 4: Test MAE of methods trained on AFAD and Fireman Datasets

Method	AFAD	Fireman
CE-NN	3.28 ± 0.04	0.8 ± 0.01
OR-NN	2.85 ± 0.03	0.76 ± 0.01
CORAL	2.99 ± 0.03	0.82 ± 0.01
CORN	2.81 ± 0.02	0.76 ± 0.01
UNICORNN	2.645 ± 0.02	0.755 ± 0.003

C.3 COMPARING UNICORNN AGAINST SIMPLE CONSTRAINT-UNIMODAL BASELINE

We conduct an additional experiment to compare our method with a simple baseline classifier trained using cross-entropy loss and a unimodality constraint. In this baseline, the model predicts a K-dimensional probability vector for each sample, and the unimodality constraint is enforced by minimizing pairwise distances between the predicted probability values. Specifically, the distances are set to be negative for probabilities at indices lower than the target label and positive for indices higher than the target label. Further implementation details are available in the project repository.

The results, shown in Table C.3, indicate that the baseline achieves near-unimodal distributions (80% unimodality) on the Fireman dataset but does not guarantee perfect unimodality. Additionally, the Mean Absolute Error (MAE) of this baseline is higher compared to UNICORNN. Notably, when applied to the AFAD dataset with the same training parameters, the baseline fails to produce unimodal predictions, highlighting its limitations in maintaining unimodality across different datasets.

Table 5: Performance Comparison on Balanced Datasets (AFAD and Fireman)

Method	AFAD (balanced)		Fireman (balanced)	
	MAE ↓	Unimodality ↑	MAE ↓	Unimodality ↑
UNICORNN	2.64 ± 0.02	1.0 ± 0.0	0.755 ± 0.003	1.0 ± 0.0
Classifier with Unimodality	2.846 ± 0.04	0.11 ± 0.04	0.808 ± 0.003	0.865 ± 0.05

D DATASETS

Table 6 contains information on the benchmark datasets used for our experiments.

Adience: During the training the images are resized to (256,256). Additionally, random crop of size 224 and random horizontal flip are applied as augmentations.

FG-Net: We partitioned the dataset to 8 classes, corresponding to decades. Augmentations are the same as in the Adience experiment.

RetinaMNIST dataset has 5 classes, and we apply random affine, horizontal and vertical flips as augmentations during the training. The size of the images is (28,28) as provided by the dataset contributors. The train/test splits are provided by the contributors and were used as-is.

810
811
812
813
814
815
816
817
818
819



820
821

Figure 5: Examples from the FG-Net dataset. Age classes are indicated above each image.

822
823
824
825
826



827
828
829

Figure 6: Examples from the EVA dataset. Aesthetics classes are indicated above each image.

830
831
832

SCUT-FBP5500 dataset contains 5500 face images beautifully ranked from 1 to 5 continuously. we partition the data into 8 classes in accordance with the rank. Augmentations are the same as in the Adience experiment.

833
834
835
836

EVA (Explainable Visual Aesthetics) dataset contains 5101 images aesthetically ranked from 0 to 10 by multiple voters. We calculate the average score for each image and partition the data into 5 classes in accordance with the average score. Augmentations are the same as in the Adience experiment.

837
838
839

AAF (All-Age-Faces) dataset is already pre-processed and contains 13,322 face images (mostly Asian), distributed across all ages (from 2 to 80). We partitioned the dataset into 6 classes. Augmentations are the same as in the Adience experiment.

840
841
842
843

AFAD (The Asian Face data)³ dataset (Niu et al., 2016) contains 165,501 faces in the age range of 15-40 years. No additional preprocessing was applied to this dataset since the faces were already centered. Following Shi et al. (2021), we use a balanced version of the AFAD data⁴ with 13 age labels in the age range of 18-30 years, resulting in total 60K samples.

844
845
846

Fireman⁵ dataset is a tabular dataset that contains 40,768 instances, 10 numeric features, and an ordinal response variable with 16 categories. We use a balanced version of this dataset⁶ consisting of 2,543 instances per class and 40,688 from the 16 ordinal classes in total.

847
848

Figures 5, 6, 7, 8 show examples from the FG-Net, EVA, AAF and SCUT-FBP5500 datasets, respectively.

849

850

E TECHNICAL DETAILS

851

852

Table 7 shows the technical details for the experiments on the real world benchmark datasets reported in this manuscript.

853
854

855
856
857
858

The Adam optimizer was used in all experiments, with the default $\beta = (0.9, 0.999)$. The means and standard deviations reported in table 1 are based on 5 repetitions of each experiment, differing in weights initialization and random train-test splits, except for Adience, for which we repeated the experiment five times, using the same train-test splits as the creators of the dataset⁷. For the

859

³<https://github.com/afad-dataset/tarball>

860

⁴<https://github.com/Raschka-research-group/corn-ordinal-neuralnet/tree/main/datasets/afad>

861

⁵https://github.com/gagolews/ordinal_regression_data

862

⁶<https://github.com/Raschka-research-group/corn-ordinal-neuralnet/tree/main/datasets/firemen>

863

⁷https://github.com/GilLevi/AgeGenderDeepLearning/tree/master/Folds/train_val_txt_files_per_fold

864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917



Figure 7: Examples from the AAF dataset. Age classes are indicated above each image.



Figure 8: Examples from the SCUT-FBP5500 dataset. Beauty classes are indicated above each image.

accuracy-preserving calibration phase (Section 4.3), we used early stopping and the number of epochs and learned parameters were changed for each dataset, as shown in Table 8.

F COMPUTATION DETAILS

We implement our model in Pytorch and run experiments on a Linux server with NVIDIA GeForce GTX 1080 Ti, A100 80GB PCIe GPUs and Intel(R) Core(TM) i7-8700 CPU 3.20GHz CPU.

918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

Table 6: Benchmark datasets characteristics

Dataset	Task	Train Images	Val Images	Test Images	Classes
Adience	age estimation		pre-defined splits		8
FG-Net	age estimation	802	100	100	8
RetinaMNIST	DR classification	1080	120	400	5
AAF	age estimation	9058	1599	2665	6
EVA	aesthetics estimation	3684	651	766	6
SCUT-FBP5500	facial beauty prediction	4250	350	900	8

Table 7: Technical details of the experiments

Dataset	Backbone	Epochs	Batch Size	Initial LR	Decay LR After (epochs)	Weight Decay
Adience	ResNet-101	100	64	10^{-4}	40	10^{-5}
FG-Net	ResNet-18	100	32	10^{-4}	40	10^{-4}
RetinaMNIST	ResNet-18	100	16	10^{-4}	80, 90	10^{-4}
AAF	ResNet-18	100	64	10^{-4}	-	10^{-3}
EVA	ResNet-18	145	64	10^{-4}	-	10^{-5}
SCUT-FBP5500	ResNet-18	100	64	10^{-4}	-	10^{-3}

Table 8: Post-hoc calibration technical details

Dataset	Epochs	Learned Parameters
Adience	100	250
FG-Net	100	250
RetinaMNIST	100	1000
AAF	100	150
EVA	100	50
SCUT-FBP5500	100	100