Unimodal, Calibrated, and Order-Aware: A Deep Architecture for Reliable Ordinal Regression

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. It is often hindered by a trade-off between respecting the inherent order of classes and producing calibrated probability estimates. While methods based on maximum likelihood generally perform well in terms of calibration, they disregard the ordinal nature of the labels. Conversely, Optimal Transport explicitly models this order but generates poorly calibrated outputs. To resolve this conflict, we propose UnicornNet, the first approach to ensure all three critical properties: it leverages an **optimal transport (OT) loss** to model the ordinal structure, employs a novel accuracy-preserving procedure to guarantee **well-calibrated probabilities**, and incorporates an architectural design that enforces **unimodality** of the output distribution - a key inductive bias for many real-world ordinal regression tasks. On six real-world image benchmarks and a tabular dataset, UnicornNet matches or improves upon recent deep ordinal regression models on both accuracy and calibration metrics, while enforcing unimodality.

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 & Ahn, 2012).

Ordinal regression assumes a label space $\mathcal{Y} = \{1, 2, \ldots, k\}$ endowed with a total order relation $\preceq$, such that $1 \preceq \ldots \preceq k$. Unlike standard classification, ordinal regression must account for this inherent order among classes. A natural approach when dealing with ordinal evaluation metrics like Mean Absolute Error (MAE) is to employ cost-sensitive loss functions where the penalty for misclassification increases strictly with the ordinal distance between predicted and true labels. For instance, predicting class "1" when the true class is "4" incurs a higher penalty than predicting "3", reflecting the severity of the prediction error.

One desirable property for ordinal regression models is that their predicted probability distributions will be unimodal. A k-level multinomial distribution is unimodal if there exists $j \in \{1, \ldots, k\}$ such that $\mathbb{P}(Y = 1) \leq \ldots \leq \mathbb{P}(Y = j) \geq \ldots \geq \mathbb{P}(Y = k)$, where Y is a random variable. In many real-world domains, unimodality is a natural requirement – for example, when predicting tumor grades, a bimodal prediction suggesting either grade "1" or "4" but not "2" or "3" would be counterintuitive and unreliable. While several recent works have identified this issue in deep ordinal regression (Gao et al., 2017; Diaz & Marathe, 2019; Liu et al., 2019a; 2020), existing approaches often only encourage unimodality through soft targets rather than enforcing it, which proves suboptimal.

Notably, the choice of training objective significantly impacts both the model's ability to capture ordered relationships and its calibration quality. Many ordinal regression methods, including the Proportional Odds Model (POM) and recent deep learning approaches (Belharbi et al., 2019; Vargas et al., 2020; Fu et al., 2018), employ maximum likelihood training. While some of these encode order through their parameterization, the likelihood objective itself depends only on the probability assigned to the true class and is insensitive to

how the remaining mass is distributed over the ordered incorrect classes, making it non-cost-sensitive for order-aware metrics such as MAE. While maximum likelihood often yields well-calibrated probabilities – where for example, a 90% predicted probability corresponds to events occurring 90% of the time, alternative losses like Optimal Transport (OT) (Hou et al., 2016; Beckham & Pal, 2017; Liu et al., 2019a) better capture label order but tend to produce overly peaked distributions that compromise calibration, as we will describe next.

In this manuscript, we introduce UnicornNet, a novel deep learning approach for ordinal regression that is the first to resolve the three fundamental challenges of the task. Specifically, (i) it enforces unimodality of the output distribution through architectural design, (ii) it effectively captures the ordered relationships among classes using OT as a training objective, and (iii) it introduces a post-training calibration step that produces well-calibrated probability estimates while preserving accuracy, thereby mitigating the calibration limitations of OT. Our analysis further reveals that OT objectives inherently favor sharply peaked distributions, providing a principled explanation for the observed calibration issues in OT-trained models. To the best of our knowledge, UnicornNet is the first framework to guarantee unimodality, ordering consistency, and calibration jointly. Experimental results, summarized in Table 1, on six real-world image benchmark datasets demonstrate that UnicornNet performs on par with or better than several state-of-the-art approaches in both prediction accuracy and probability calibration.

2 Related Work

Ordinal regression has a long history in machine learning and statistics, with recent work focusing on deep learning-based methods. A common approach reformulates the task as multi-label classification (Fu et al., 2018; Liu et al., 2017; 2018b; Vishnu et al., 2019; Berg et al., 2020; Cheng et al., 2008; Li et al., 2021), where the task is framed as predicting a sequence of binary decisions indicating whether the target label exceeds each possible threshold. However, this approach has two key issues: (1) the predicted cumulative probabilities are not always consistent (i.e., non-decreasing), and (2) even when consistency is enforced (Liu et al., 2018a; Shi et al., 2023; Cao et al., 2020), the resulting class probabilities are often not unimodal (Liu et al., 2019b; Vargas et al., 2020; Pan et al., 2018; Kook et al., 2020). At the other extreme, ordinal labels can be treated as continuous-valued targets, as in continuous representation learning (Zha et al., 2023) and imbalanced regression (Yang et al., 2021), which operate over a real-valued continuum rather than a finite set of ordered categories. Ordinal regression sits between these two views: it predicts discrete labels, as in classification, while accounting for their order, as in regression.

One elegant way to ensure unimodal output probabilities is to model the conditional distribution using a Poisson or Binomial distribution (Beckham & Pal, 2017), both inherently unimodal. The model predicts a scalar parameter (λ for Poisson, p for Binomial), which is mapped to a normalized probability mass function. A shared parameter τ controls the shape of the distribution. While architecturally convenient, this approach cannot flexibly express uncertainty, as a single parameter governs both the mode and the scale.

Another approach promotes unimodality by training with soft targets-smoothed label distributions that assign nonzero probabilities to nearby classes to reflect ordinal structure (Gao et al., 2017; Diaz & Marathe, 2019; Liu et al., 2019a; 2020). However, this has two key drawbacks. First, unimodality is encouraged but not guaranteed; as shown in our experiments, such models can still produce non-unimodal outputs. Second, soft targets use a fixed decay pattern that encodes no uncertainty, making them equivalent to one-hot labels in terms of information content. Other methods include using additional loss terms (Li et al., 2022; Cardoso et al., 2023) or constrained optimization (Belharbi et al., 2019), but these offer no guarantees at test time.

Several works train with cross-entropy and one-hot or binary targets (Belharbi et al., 2019; Vargas et al., 2020; Fu et al., 2018; Beckham & Pal, 2017; Berg et al., 2020; Vishnu et al., 2019; Cardoso et al., 2023; Yamasaki, 2022), and accuracy-preserving calibration has been proposed to improve these models (Esaki et al., 2024). However, with one-hot targets, cross-entropy reduces to the negative log-probability of the true class, making it insensitive to the remaining mass. This ignores class order and limits effectiveness in ordinal settings where metrics like MAE matter. While cross-entropy encourages calibrated probabilities (Lakshminarayanan et al., 2017), it is not cost-sensitive. To incorporate order, some works use OT loss (Hou et al., 2016; Beckham &

Table 1: Summary Table Reporting the Average Performance Across Six Image Datasets. The Full Per-Dataset Results Are Provided in Table 2. Lower MAE/ECE is Better; Higher Unimodality is Better.

Method	MAE ↓	Unimodality (%) ↑	ECE ↓
Beckham and Pal	0.597	100.0	0.170
Liu et al.	0.515	46.7	0.227
DLDL	0.582	75.2	0.317
SORD	0.532	97.5	0.102
POM	0.548	55.0	0.293
UnimodalNet	0.553	100.0	0.305
CORN	0.565	–	–
ORCU	0.508	100.0	0.095
UnicornNet	**0.483**	**100.0**	**0.073**

Pal, 2017; Liu et al., 2019a), which captures ordinal structure as a cost-sensitive loss. However, as will be discussed, OT loss tends to produce overconfident predictions.

Closely related to our work, ORCU (Kim et al., 2025) jointly promotes unimodality and calibration through a soft ordinal encoding and a regularization term. However, its soft encoding is fixed and data-independent, and unimodality is encouraged rather than guaranteed; in contrast, UnicornNet learns the distribution location and scale from data and guarantees unimodality by construction.

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

3 Preliminaries

3.1 The Proportional Odds Model

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

$$\begin{aligned}
&\mathbb{P}(Y \preceq j | \boldsymbol{X} = \boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \\
&= \mathbb{P}(\boldsymbol{\beta}^T \boldsymbol{x} + \boldsymbol{\epsilon} \leq \boldsymbol{\alpha}_j | \boldsymbol{X} = \boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \\
&= F(\boldsymbol{\alpha}_j - \boldsymbol{\beta}^T \boldsymbol{x})
\end{aligned} \tag{1}$$

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

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

It is convenient to interpret Eq. (1) by viewing $\boldsymbol{\beta}^T \boldsymbol{x}$ 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 Fig. 1.

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

$$\begin{aligned}
&\mathbb{P}(Y = y | \boldsymbol{X} = \boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \\
&= F(\alpha_y - \boldsymbol{\beta}^T \boldsymbol{x}) - F(\boldsymbol{\alpha}_{y-1} - \boldsymbol{\beta}^T \boldsymbol{x}),
\end{aligned} \tag{2}$$

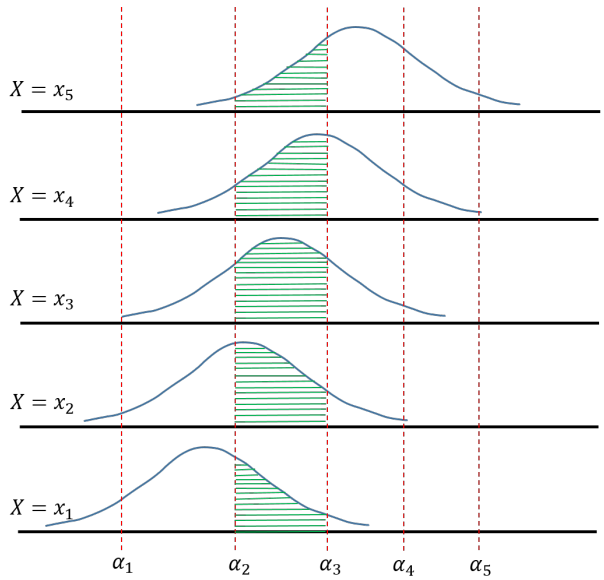

Figure 1: The Proportional Odds Model. $\boldsymbol{x_i}$ is a realization of $\boldsymbol{X}$. The standard logistic density is shifted by $\boldsymbol{\beta}^T\boldsymbol{x_i}$. The thresholds $\boldsymbol{\alpha}_j$ define the bins which determine the probability predicted by the model to each class. For example, the green area defines the probability $\mathbb{P}(Y = 3|\boldsymbol{X} = \boldsymbol{x}; \boldsymbol{\alpha}, \boldsymbol{\beta})$.

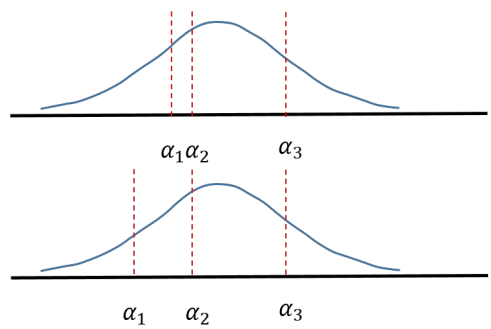

Figure 2: Limitations of POM. The plots show two examples illustrating key limitations of POM: (1) it can produce non-unimodal probability outputs—for instance, $\mathbb{P}(Y{=}2|\boldsymbol{X} = \boldsymbol{x})$ can be lower than both $\mathbb{P}(Y{=}1|\boldsymbol{X} = \boldsymbol{x})$ and $\mathbb{P}(Y{=}3|\boldsymbol{X} = \boldsymbol{x})$; (2) its likelihood is insensitive to how incorrect probabilities are distributed—as long as the correct class (e.g., 3) gets the same mass, the likelihood remains unchanged, even if more mass is assigned to nearby classes in one case.

considering $\boldsymbol{\alpha}_0 = -\infty$ and $\boldsymbol{\alpha}_k = \infty$. 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 main limitations. First, its output probabilities are not guaranteed to be unimodal (Fig. 1). Second, the likelihood in Eq. (2) depends only on the predicted probability of the true class, ignoring how the remaining mass is distributed. This neglects label order and may degrade performance (Fig. 2). The same issue arises with ordinal likelihood (Chu et al., 2005) and cross-entropy loss under one-hot targets. UnicornNet will address these shortcomings.

3.2 Probability Calibration

Probability Calibration refers to the alignment between the predicted probabilities generated by a classification model and the true, empirical probabilities observed in the data. For instance, it is anticipated that when a classification model assigns a probability of 0.8 to class i for a certain sample, approximately 80% of those samples would indeed belong to class i based on their ground truth labels. Formally, the concept of calibration can be defined as (Wang, 2023):

$$\mathbb{P}(Y = i|p_i(\boldsymbol{X}) = q) = q, \quad \forall q \in [0, 1], i \in [k] \tag{3}$$

Here, k denotes the number of classes, $\boldsymbol{X}$ and Y are random variables that represent the model input and the ground truth class, respectively, and $p(\boldsymbol{X}) \in \mathbb{R}^k$ where $p_i(\boldsymbol{X})$ is a random variable that signifies the model's output probability for class i given the random variable $\boldsymbol{X}$.

Denote $\boldsymbol{e_Y} \in \mathbb{R}^k$ as the one-hot encoding random vector of Y, so that the true conditional distribution of Y given $\boldsymbol{X} = \boldsymbol{x}$ is $\mathbb{E}_{Y|\boldsymbol{X}}[\boldsymbol{e_Y} \mid \boldsymbol{x}]$, whose i-th entry is $\mathbb{P}(Y = i \mid \boldsymbol{X} = \boldsymbol{x})$. A model whose outputs match this distribution, $p(\boldsymbol{x}) = \mathbb{E}_{Y|\boldsymbol{X}}[\boldsymbol{e_Y} \mid \boldsymbol{x}]$, satisfies Eq. (3) and is therefore calibrated. This motivates fitting the model outputs to the true conditional distribution. For each $\boldsymbol{x}$, the squared L_2 objective $\mathbb{E}_{Y|\boldsymbol{X}}\left[\|\boldsymbol{e_Y} - p(\boldsymbol{x})\|_2^2 \mid \boldsymbol{x}\right]$ is minimized exactly when $p(\boldsymbol{x}) = \mathbb{E}_{Y|\boldsymbol{X}}[\boldsymbol{e_Y} \mid \boldsymbol{x}]$, since the conditional mean is the unique minimizer of the

expected squared error. As this conditional distribution is unknown, we approximate the expectation using a single Monte Carlo sample, the observed one-hot label $\boldsymbol{e_y}$, i.e., minimize the following:

$$BS(\boldsymbol{e_y}, p(\boldsymbol{x})) = \|\boldsymbol{e_y} - p(\boldsymbol{x})\|_2^2 \tag{4}$$

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

Accuracy-Preserving Calibration This type of calibration is performed by adjusting a model's predicted probabilities to improve calibration without altering accuracy. Temperature Scaling (Guo et al., 2017) learns a single temperature to rescale logits before softmax, while Adaptive Temperature Scaling (Balanya et al., 2024) generalizes this by learning an input-dependent mapping $T(\mathbf{x})$. As part of UnicornNet, we introduce a calibration approach trained via BS that preserves accuracy. In contrast, more expressive post-hoc methods such as Dirichlet calibration (Kull et al., 2019) are not accuracy-preserving: they re-fit the full class distribution against one-hot targets and can therefore alter the predicted class, potentially degrading the MAE attained during OT training.

3.3 Optimal Transport

OT distance provides a principled way to measure the discrepancy between two probability distributions by quantifying the minimal effort required to transform one distribution into another under a given cost function. Specifically, let $\boldsymbol{p}$ and $\boldsymbol{q}$ be probability distributions over a finite metric space $M = \{1, \ldots, k\}$, equipped with a cost function $c(i, j)$ between elements $i, j \in M$. When the target distribution $\boldsymbol{q}$ is a Dirac distribution concentrated on a single element $j \in M$, the OT distance reduces to

$$\mathrm{OT}(\boldsymbol{p}, \boldsymbol{q}) = \sum_i p_i \, c(i, j). \tag{5}$$

When $\boldsymbol{p}$ represents model outputs and $\boldsymbol{q}$ is a one-hot target, this objective is differentiable with respect to $\boldsymbol{p}$. The cost function c can encode domain knowledge; a natural choice is $c(i, j) = |i - j|^m$ with $m \geq 1$.

Another theoretical motivation for incorporating OT in ordinal regression arises from Proposition 5.2 in Frogner et al. (2015). Let $\mathcal{H}$ be a hypothesis class of functions $h_\theta : \mathcal{X} \to \Delta^k$, where Δ^k denotes the probability simplex over k ordinal labels. Let $h_{\hat{\theta}}$ be the OT empirical risk minimizer. For an input $\boldsymbol{x}$, let $h_\theta(\boldsymbol{x}) \in \Delta^k$ be the predicted distribution, and $\boldsymbol{e_y}$ the one-hot vector corresponding to the true label $y \in \{1, \ldots, k\}$. We define $\hat{y}_{\hat{\theta}}(\boldsymbol{x}) := \arg\max_j h_{\hat{\theta}}(\boldsymbol{x})_j$ to be the predicted class. Then:

Proposition 1. *(Frogner et al., 2015) For the cost function $c(i, j) = |i - j|$ and any $\delta > 0$, with probability at least $1 - \delta$ it holds that*

$$\mathbb{E}_{\boldsymbol{x}, y}\left[|\hat{y}_{\hat{\theta}}(\boldsymbol{x}) - y|\right] \leq \inf_{h_\theta \in \mathcal{H}} k \cdot \mathbb{E}\left[OT(h_\theta(\boldsymbol{x}), \boldsymbol{e_y})\right] + C$$

Here, C refers to a constant term that depends on k, δ, the number of samples, the metric, and properties of the hypothesis space, such as its Rademacher complexity. The proposition implies that minimizing the OT loss leads to lower MAE by encouraging predictions that respect the ordinal structure of the labels, making it a well-suited loss function for ordinal regression tasks. A related, more general bound on ordinal misclassification error has been established via reductions to weighted binary classification (Li & Lin, 2006).

However, one drawback of OT as a loss function is its tendency to prioritize peaked distributions over the actual probabilities. For instance, consider the conditional probabilities $\mathbb{P}(Y = 1 \mid \boldsymbol{x}) = 0.25$, $\mathbb{P}(Y = 2 \mid \boldsymbol{x}) = 0.5$, $\mathbb{P}(Y = 3 \mid \boldsymbol{x}) = 0.25$, and the cost function $c(i, j) = |i - j|$. Suppose two candidate model outputs: the calibrated $\boldsymbol{p} = [0.25, 0.5, 0.25]$, which equals the true conditional distribution $\boldsymbol{\pi}(\boldsymbol{x})$ of Y given $\boldsymbol{X} = \boldsymbol{x}$, and the peaked $\boldsymbol{e_2} = [0, 1, 0]$. For a one-hot label $\boldsymbol{e_y}$, Eq. (5) gives $\mathrm{OT}(\boldsymbol{p}, \boldsymbol{e_y}) = \sum_i p_i \, |i - y|$, so $\mathrm{OT}(\boldsymbol{p}, \boldsymbol{e_1}) = 1$, $\mathrm{OT}(\boldsymbol{p}, \boldsymbol{e_2}) = 0.5$, and $\mathrm{OT}(\boldsymbol{p}, \boldsymbol{e_3}) = 1$, while $\mathrm{OT}(\boldsymbol{e_2}, \boldsymbol{e_y}) = |2 - y|$ equals 0 at $y = 2$ and 1 at $y \in \{1, 3\}$.

Taking the expectation over Y then gives $\mathbb{E}_{Y|\boldsymbol{X}}[\mathrm{OT}(\boldsymbol{p}, \boldsymbol{e_Y}) \mid \boldsymbol{x}] = 0.25 \cdot 1 + 0.5 \cdot 0.5 + 0.25 \cdot 1 = 0.75$, whereas $\mathbb{E}_{Y|\boldsymbol{X}}[\mathrm{OT}(\boldsymbol{e_2}, \boldsymbol{e_Y}) \mid \boldsymbol{x}] = 0.25 \cdot 1 + 0.5 \cdot 0 + 0.25 \cdot 1 = 0.5$. Although $\boldsymbol{p}$ is the calibrated output, the peaked $\boldsymbol{e_2}$ attains the lower expected OT loss, indicating that OT tends to favor peaked distributions over calibrated ones.

The following proposition formalizes this phenomenon for an arbitrary cost function and true conditional distribution; the proof is given in Appendix A.3.

Proposition 2. *Fix an input $\boldsymbol{x}$ with true conditional distribution $\boldsymbol{\pi}(\boldsymbol{x}) \in \Delta^k$, $\pi_i(\boldsymbol{x}) = \mathbb{P}(Y = i \mid \boldsymbol{X} = \boldsymbol{x})$, and let*

$$R(\boldsymbol{p}) := \mathbb{E}_{Y|\boldsymbol{X}}[OT(\boldsymbol{p}, \boldsymbol{e_Y}) \mid \boldsymbol{x}], \qquad \boldsymbol{p} \in \Delta^k,$$

denote the expected OT loss of output $\boldsymbol{p}$ against one-hot label $\boldsymbol{e_Y}$. For any cost function c, R is linear in $\boldsymbol{p}$ and therefore minimized at a vertex of the simplex: an optimal output is a one-hot vector $\boldsymbol{e_{i^\star}}$, where $i^\star$ is the class of least expected cost (a median of $\boldsymbol{\pi}(\boldsymbol{x})$ when $c(i,j) = |i-j|$). Consequently, unless $\boldsymbol{\pi}(\boldsymbol{x})$ is already concentrated on a single class, the calibrated output $\boldsymbol{p} = \boldsymbol{\pi}(\boldsymbol{x})$ does not minimize the expected OT loss.

Together, the two propositions expose a tension in OT training. Proposition 1 is reassuring: the *class* an OT-trained model predicts (its arg max) comes with a bound on the MAE, so OT is well suited to ordinal regression. Proposition 2 sounds the warning: the *full distribution* that minimizes the OT loss is not the calibrated $\boldsymbol{\pi}(\boldsymbol{x})$ but a one-hot spike at the median. The peak thus sits in the right place, exactly where it helps the MAE, yet all probability mass piles onto it, leaving the model over-confident and miscalibrated. Put differently, OT is not a proper scoring rule (Gneiting & Raftery, 2007): unlike the Brier score, its risk is not minimized by reporting the true conditional probabilities.

The remedy follows directly: since OT already places the peak where we want it, we keep the loss and correct only the spread of probability mass around that peak. A calibration step that repairs this spread while leaving the mode untouched preserves accuracy and recalibrates only the probabilities. The next section introduces UnicornNet, which realizes this idea.

4 UnicornNet

4.1 Rationale

Achieving unimodality by design provides a decisive advantage: every input is guaranteed to yield a unimodal probability distribution. While this principle underlies the mechanism of Beckham & Pal (2017), our approach UnicornNet advances it by decoupling the distribution's *mode* and *scale* into two parameters (μ, σ). This separation offers finer control – μ dictates where the peak lies, while σ tunes how sharply or diffusely probability mass spreads, allowing the model to adapt its confidence structure with greater precision than a single-parameter formulation.

For training, we move away from maximum likelihood estimation, whose objective is not cost-sensitive to the ordinal distance between classes. Instead, we optimize an *OT loss*, which naturally respects the ordered structure of classes. To counter the aforementioned issue of OT producing poorly calibrated probabilities, we introduce a lightweight yet principled calibration step: by refining only the scale parameter σ with BS as a surrogate loss, we preserve accuracy while aligning predicted probabilities with empirical frequencies.

Taken together, these design choices form a tightly integrated mechanism that guarantees *unimodality by construction*, *ordinal sensitivity through OT*, and *reliability via post-hoc calibration*. The result is probability estimates that combine theoretical rigor with practical reliability.

4.2 Unimodal Output Distribution 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 modifying the CDF, letting the bins be of equal length and remain fixed during training.

Therefore, instead of learning the thresholds, during training two input-dependent maps $\boldsymbol{x} \mapsto \mu, \boldsymbol{x} \mapsto \sigma$ are learned, where μ, σ are the location and scale parameters, respectively. Both define a truncated normal distribution $\mathcal{N}_{\text{trunc}}(\mu, \sigma^2, -1, 1)$ (a normal distribution that has 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 $-1 = \alpha_0, \alpha_1, ..., \alpha_k = 1$, so that $\alpha_i - \alpha_{i-1} = \frac{2}{k}$. The probabilities are given by

$$p_i(\boldsymbol{x}) = \mathbb{P}(Y = i | \boldsymbol{X} = \boldsymbol{x}; \mu, \sigma) = \\ F_{\mu(\boldsymbol{x}),\sigma(\boldsymbol{x})}(\alpha_i) - F_{\mu(\boldsymbol{x}),\sigma(\boldsymbol{x})}(\alpha_{i-1}), \tag{6}$$

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 $\boldsymbol{x}$.

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 $\boldsymbol{x} \mapsto \mu, \boldsymbol{x} \mapsto \sigma$ are expressed via two deep neural networks (DNNs) which share a common backbone model, hence can represent a complex nonlinear relation. Our proposed mechanism for generating unimodal output probabilities is depicted in Fig. 3. The following lemma, proved in Appendix A.1, establishes that the model output probabilities are indeed unimodal.

Lemma 1. *Let $\boldsymbol{x} \in \mathbb{R}^d$ be an input to the model, which is mapped to $\mu = \mu(\boldsymbol{x}), \sigma = \sigma(\boldsymbol{x})$. Let $p_1, \dots p_k$ be the model output probabilities, generated via Eq. (6). Then $p_1, \dots p_k$ define a unimodal multinomial random variable.*

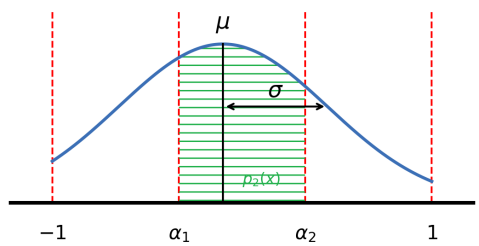

Figure 3: Generation of Unimodal Output Probabilities for $k = 3$ Classes. An input $\boldsymbol{x}$ is mapped to a location μ and a scale σ, which together define a truncated normal distribution $\mathcal{N}_{\text{trunc}}(\mu, \sigma^2, -1, 1)$ on $[-1, 1]$. The location μ sets the peak of the distribution (and hence the predicted class), while the scale σ controls how widely the probability mass spreads around the peak. The interval $[-1, 1]$ is divided into k equal-length bins, one per class, and each output probability is the mass the distribution assigns to the corresponding bin; the green area is $p_2(\boldsymbol{x}) = \mathbb{P}(Y = 2 \mid \boldsymbol{X} = \boldsymbol{x})$.

4.3 Accuracy-Preserving Calibration

UnicornNet introduces a re-training procedure for $\boldsymbol{x} \mapsto \sigma$ mapping while preserving the already trained mapping $\boldsymbol{x} \mapsto \mu$. The learning of $\boldsymbol{x} \mapsto \sigma$ is done by minimizing the Brier Score loss in Eq. (4), which ensures well-calibrated probabilities and preserves the model accuracy. Since the mapping $\boldsymbol{x} \mapsto \sigma$ controls the decay of the generated probabilities and the mapping $\boldsymbol{x} \mapsto \mu$ which controls the mode, remains unchanged, the accuracy and MAE are not affected, although the probability output distribution is indeed modified, resulting in an accuracy-preserving calibration. We state it in the following lemma (proved in Appendix A.2).

Lemma 2. *Let $\boldsymbol{x} \in \mathbb{R}^d$ be an input to the model, mapped to $\mu = \mu(\boldsymbol{x})$, $\sigma_1 = \boldsymbol{\sigma_1}(\boldsymbol{x})$. Let $\sigma_2 = \boldsymbol{\sigma_2}(\boldsymbol{x})$ be a re-trained mapping via minimization of loss using Eq. (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 Eq. (6). Then:*

$$\text{argmax}_{1 \le i \le k}\, p_i^{\sigma_1} = \text{argmax}_{1 \le i \le k}\, p_i^{\sigma_2}.$$

We adopt a two-stage scheme rather than a joint objective $\mathcal{L}_{\mathrm{OT}} + \lambda\,\mathcal{L}_{\mathrm{BS}}$, as the two losses act on σ in opposing directions: the OT loss favors small σ, producing sharply peaked distributions, whereas the Brier Score favors larger σ, producing smoother and better-calibrated ones. Joint optimization is consequently sensitive to the weighting λ and, in our experiments, underperformed the two-stage procedure.

4.4 Training Procedure

To summarize, UnicornNet's training process consists of two distinct phases, as detailed in Algorithms 1, 2. In the first phase (Alg. 1), the parameters $\boldsymbol{\theta}_\mu$ and $\boldsymbol{\theta}_\sigma$ of $\mu(\cdot)$ and $\sigma(\cdot)$, respectively, along with the backbone parameters $\boldsymbol{\phi}$, are jointly optimized using Gradient Descent to minimize the OT loss defined in Eq. (5).

In the second phase (Alg. 2), the parameters $\boldsymbol{\theta}_\mu$ and $\boldsymbol{\phi}$ are kept fixed, while $\boldsymbol{\theta}_\sigma$ is further optimized via Gradient Descent to minimize the BS defined in Eq. (4).

Algorithm 1 Optimal Transport-based Training

Input: Training data $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{N}$, number of classes k, batch size b, number of epochs T
Output: Parameters $\boldsymbol{\theta}_\mu$, $\boldsymbol{\theta}_\sigma$ of $\mu(\cdot)$ and $\sigma(\cdot)$, respectively, and backbone parameters $\boldsymbol{\phi}$

```
Initialize parameters θ_μ, θ_σ, φ
for epoch t = 1 to T do
    for each batch B ⊆ D of size b do
        Compute μ_{φ,θ_μ}(x_i) and σ_{φ,θ_σ}(x_i) for all x_i ∈ B and calculate p(x_i) using Eq. (6)
        Compute the loss
        L_OT(B) = (1/b) Σ_{(x_i,y_i)∈B} OT(p(x_i), y_i) using Eq. (5)
        Update θ_μ, θ_σ, φ via Gradient Descent
    end for
end for
Return θ_μ, θ_σ, φ
```

Algorithm 2 Accuracy Preserving Calibration

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

```
for epoch t = 1 to T do
    for each batch B ⊆ D of size b do
        Compute μ_{φ,θ_μ}(x_i) and σ_{φ,θ_σ}(x_i) for all x_i ∈ B and calculate p(x_i) using Eq. (6)
        Compute the loss
        L_BS(B) = (1/b) Σ_{(x_i,y_i)∈B} BS(p(x_i), y_i) using Eq. (4)
        Update only θ_σ via Gradient Descent
    end for
end for
Return θ_σ
```

5 Experimental Results

5.1 Datasets

We evaluate UnicornNet 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., 2023)), image aesthetics estimation (EVA (Kang et al., 2020)). In addition to these image benchmarks, we evaluate UnicornNet on the Fireman tabular dataset (16 ordinal classes) to assess its performance beyond the image domain. A more detailed description, with sampled images, of the datasets appears in Appendix D.

Table 2: Performance of Various Methods on Real-World Datasets (Mean ± Std).

Dataset	Method	MAE ↓	OOA ↑	Spearman ↑	QWK ↑	% Unimodal ↑	ECE ($b = 10$) ↓
Adience	Beckham and Pal	.50 ± .06	.93 ± .01	**.90 ± .01**	**.91 ± .02**	**1 ± 0**	.23 ± .05
	Liu et al.	.47 ± .05	.94 ± .01	.89 ± .02	.90 ± .02	.47 ± .04	.21 ± .03
	DLDL	.49 ± .06	.93 ± .01	.88 ± .02	.89 ± .03	.62 ± .08	.42 ± .04
	SORD	.47 ± .06	.94 ± .01	.89 ± .02	.90 ± .02	.99 ± .003	.14 ± .03
	POM	.48 ± .06	.94 ± .01	.89 ± .03	**.91 ± .03**	.81 ± .04	.32 ± .04
	UnimodalNet	.49 ± .06	.93 ± .01	.88 ± .02	.90 ± .03	**1 ± 0**	.34 ± .04
	CORN	.52 ± .07	–	–	–	–	–
	ORCU	**.45 ± .06**	**.95 ± .01**	**.90 ± .02**	**.91 ± .03**	**1 ± 0**	**.06 ± .01**
	UnicornNet	.46 ± .05	**.95 ± .01**	**.90 ± .02**	**.91 ± .02**	**1 ± 0**	.07 ± .03
Retina MNIST	Beckham and Pal	.80 ± .02	.79 ± .01	.58 ± .02	.55 ± .02	**1 ± 0**	.17 ± .01
	Liu et al.	.68 ± .02	**.82 ± .01**	**.61 ± .02**	.58 ± .02	.72 ± .03	.27 ± .01
	DLDL	.72 ± .02	.81 ± .01	.59 ± .02	.58 ± .01	.98 ± .01	.29 ± .01
	SORD	.75 ± .02	.78 ± .01	.58 ± .01	.57 ± .01	.87 ± .03	.11 ± .01
	POM	.83 ± .02	.76 ± .01	.53 ± .02	.53 ± .02	.43 ± .03	.17 ± .01
	UnimodalNet	.74 ± .02	.80 ± .01	.59 ± .01	.58 ± .01	**1 ± 0**	.14 ± .01
	CORN	.76 ± .02	–	–	–	–	–
	ORCU	.69 ± .02	**.83 ± .02**	**.62 ± .02**	**.59 ± .02**	.99 ± .01	.22 ± .01
	UnicornNet	**.67 ± .01**	.82 ± .01	.61 ± .01	**.59 ± .01**	**1 ± 0**	**.06 ± .01**
FG-NET	Beckham and Pal	.44 ± .06	.95 ± .03	.80 ± .04	.79 ± .03	**1 ± 0**	.16 ± .02
	Liu et al.	.32 ± .08	.96 ± .02	.86 ± .03	.84 ± .04	.01 ± .01	.08 ± .02
	DLDL	.40 ± .09	.96 ± .02	.81 ± .03	.81 ± .03	.12 ± .04	.49 ± .06
	SORD	.36 ± .08	.97 ± .02	.84 ± .05	.82 ± .04	.99 ± .01	.26 ± .05
	POM	.33 ± .06	.97 ± .02	.84 ± .04	.85 ± .04	.41 ± .06	.21 ± .03
	UnimodalNet	.36 ± .07	.97 ± .01	.84 ± .02	.84 ± .03	**1 ± 0**	.25 ± .04
	CORN	.34 ± .07	–	–	–	–	–
	ORCU	.33 ± .07	.98 ± .03	.85 ± .03	.86 ± .04	**1 ± 0**	.11 ± .02
	UnicornNet	**.31 ± .08**	**.99 ± .02**	**.87 ± .03**	**.88 ± .01**	**1 ± 0**	**.07 ± .02**
AAF	Beckham and Pal	.61 ± .13	.91 ± .05	.81 ± .03	.80 ± .05	**1 ± 0**	.16 ± .04
	Liu et al.	.43 ± .01	**.97 ± .005**	.82 ± .01	**.85 ± .01**	.7 ± .17	.22 ± .01
	DLDL	.54 ± .02	.95 ± .01	.77 ± .01	.77 ± .02	.95 ± .02	.32 ± .02
	SORD	.44 ± .02	.96 ± .01	.82 ± .01	.84 ± .01	**1 ± 0**	.14 ± .02
	POM	.45 ± .01	.96 ± .01	.80 ± .01	.83 ± .01	.48 ± .07	.28 ± .02
	UnimodalNet	.45 ± .01	.96 ± .01	.80 ± .01	.83 ± .01	**1 ± 0**	.30 ± .01
	CORN	.46 ± .01	–	–	–	–	–
	ORCU	.44 ± .02	.96 ± .005	.82 ± .01	.84 ± .01	**1 ± 0**	.04 ± .02
	UnicornNet	**.42 ± .01**	**.97 ± .005**	**.83 ± .01**	**.85 ± .005**	**1 ± 0**	**.03 ± .01**
SCUT-FBP5500	Beckham and Pal	.62 ± .12	.92 ± .04	.84 ± .02	.81 ± .05	**1 ± 0**	.19 ± .08
	Liu et al.	.54 ± .03	.95 ± .01	.83 ± .01	.82 ± .01	.18 ± .14	.27 ± .02
	DLDL	.68 ± .05	.90 ± .01	.79 ± .02	.73 ± .02	.86 ± .08	.28 ± .04
	SORD	.57 ± .03	.94 ± .01	.83 ± .01	.81 ± .01	.99 ± .005	.09 ± .02
	POM	.53 ± .03	.95 ± .01	.83 ± .01	.83 ± .01	.28 ± .04	.33 ± .03
	UnimodalNet	.52 ± .02	.95 ± .01	.82 ± .01	.83 ± .01	**1 ± 0**	.35 ± .01
	CORN	.58 ± .03	–	–	–	–	–
	ORCU	.52 ± .04	.95 ± .01	.83 ± .01	.83 ± .02	**1 ± 0**	**.05 ± .01**
	UnicornNet	**.47 ± .02**	**.96 ± .01**	**.85 ± .01**	**.85 ± .01**	**1 ± 0**	.06 ± .02
EVA	Beckham and Pal	.61 ± .03	.93 ± .01	**.60 ± .01**	**.60 ± .02**	**1 ± 0**	.11 ± .01
	Liu et al.	.65 ± .02	.91 ± .02	.53 ± .01	.52 ± .02	.72 ± .05	.31 ± .01
	DLDL	.66 ± .03	.91 ± .01	.53 ± .02	.52 ± .02	.98 ± .01	.20 ± .01
	SORD	.60 ± .02	.93 ± .01	.58 ± .02	.57 ± .02	**1 ± 0**	**.07 ± .02**
	POM	.65 ± .03	.91 ± .01	.53 ± .02	.53 ± .03	.89 ± .02	.45 ± .02
	UnimodalNet	.66 ± .03	.91 ± .01	.52 ± .02	.52 ± .02	**1 ± 0**	.45 ± .02
	CORN	.67 ± .02	–	–	–	–	–
	ORCU	.62 ± .02	.93 ± .01	.55 ± .02	.55 ± .02	**1 ± 0**	.09 ± .02
	UnicornNet	**.57 ± .01**	**.94 ± .005**	**.6 ± .01**	.58 ± .01	**1 ± 0**	.08 ± .02

5.2 Benchmarks

We compare UnicornNet to eight 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 & Marathe, 2019), an approach utilizing soft labels, generated using linear exponentially decaying distributions, trained using Kullback-Leibler divergence minimization.
- Beckham & Pal (2017), an architectural-based approach in which unimodal output probabilities are generated using the binomial distribution (single-learned parameter), trained using OT loss.
- Liu et al. (2019a), an approach utilizing soft labels, created as a mixture of Dirac, uniform, and linear exponentially decaying distributions, trained using OT 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.
- CORN (Shi et al., 2023), a method ensuring rank-consistent ordinal regression by modeling conditional probabilities over binary classification tasks, trained on conditional data subsets and using the probability chain rule, while avoiding restrictive weight-sharing constraints. CORN outputs only the predicted class per instance, without a full probability distribution; following its original evaluation protocol, we report only the MAE.
- ORCU (Kim et al., 2025), a loss function that jointly promotes unimodality and well-calibrated confidence estimates via soft ordinal encoding and an ordinal-aware regularization term.

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 C.

5.3 Evaluation Metrics

We report several common evaluation metrics for ordinal regression: MAE (Baccianella et al., 2009), One-Off Accuracy (OOA) (ELKarazle et al., 2022), Spearman correlation (Spearman, 1961), Quadratic Weighted Kappa (QWK) (Cohen, 1968), and the percentage of test examples with unimodal output probabilities.

In addition, we evaluate the model's probability calibration using the well known Expected Calibration Error (ECE) (Guo et al., 2017). 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: $ECE = \sum_{i=1}^{b} \frac{|B_i|}{|X|} |\mathrm{Acc}(B_i) - \mathrm{Conf}(B_i)|$, where b is the number of probability bins (in all experiments $b = 10$), $\mathrm{Acc}(B_i)$ is the empirical accuracy in bin B_i and $\mathrm{Conf}(B_i)$ 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

Table 2 shows the test results of each method on the six image benchmark datasets. As can be seen, UnicornNet performs at least on par and often better than the compared baselines, in a fairly consistent manner, across the various datasets and evaluation metrics. In terms of MAE, UnicornNet achieves the lowest or near-lowest error on all datasets; its key advantage is jointly delivering well-calibrated and guaranteed-unimodal

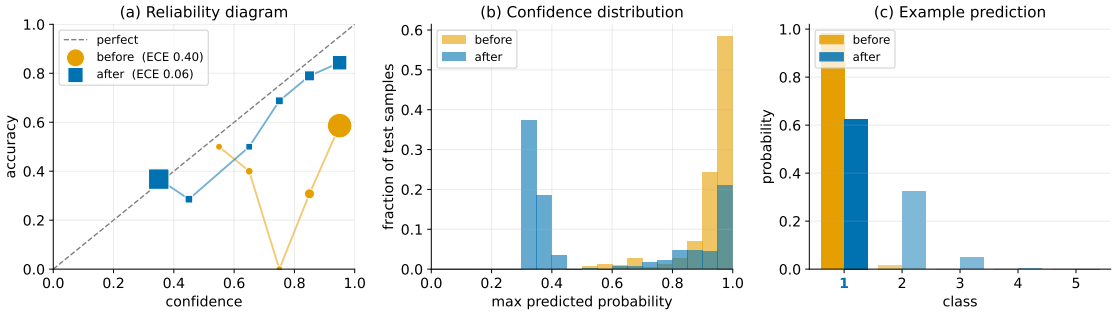

Figure 4: Accuracy-preserving σ-scaling on Retina MNIST, which preserves the predicted mode and thus MAE. **(a)** Reliability diagram (marker size $\propto$ bin count): OT is overconfident (below the diagonal), while σ-scaling tracks the diagonal. **(b)** Distribution of the maximum predicted probability: OT concentrates mass near 1; σ-scaling spreads it to reflect uncertainty. **(c)** Example prediction (true class opaque): peaked under OT, smoothed by σ-scaling while preserving the mode.

probabilities, which no baseline provides simultaneously. In addition, observe that only UnicornNet, UnimodalNet, ORCU and Beckham & Pal (2017) output consistently unimodal probabilities, while the other baselines, trained using soft targets, do not always output unimodal probabilities. Yet, unlike UnimodalNet and Beckham & Pal (2017), which tend to produce poorly calibrated probability estimates, UnicornNet consistently produce well calibrated probability estimates, where on most of the datasets, UnicornNet outperforms all baselines, while on the other datasets, the difference from ORCU is negligible. To sum up the results, UnicornNet produces unimodal and well-calibrated probabilities without compromising accuracy. Additional experiments are presented in Appendix B.

Tabular Data Beyond image data, Table 3 reports results on the Fireman tabular dataset (16 ordinal classes). UnicornNet attains MAE and calibration on par with the best baseline while guaranteeing unimodality. UnimodalNet is competitive on this dataset, but its calibration degrades considerably on the image benchmarks (Table 2), whereas UnicornNet remains consistently strong across domains.

Table 3: Performance on the Fireman Tabular Dataset (Mean $\pm$ Std).

Method	**MAE** $\downarrow$	**OOA** $\uparrow$	**Spearman** $\uparrow$	**QWK** $\uparrow$	**% Unimodal** $\uparrow$	**ECE** ($b = 10$) $\downarrow$
Beckham and Pal	.84 $\pm$ .02	.81 $\pm$ .01	**.97 $\pm$ .003**	**.97 $\pm$ .002**	**1 $\pm$ 0**	.09 $\pm$ .01
Liu et al.	1.36 $\pm$ .05	.62 $\pm$ .04	.92 $\pm$ .001	.92 $\pm$ .001	.01 $\pm$ .01	.45 $\pm$ .03
SORD	1.01 $\pm$ .01	.76 $\pm$ .001	.95 $\pm$ .003	.95 $\pm$ .004	.93 $\pm$ .03	.17 $\pm$ .00
POM	.77 $\pm$ .002	.85 $\pm$ .001	**.97 $\pm$ .001**	**.97 $\pm$ .001**	.01 $\pm$ .005	.08 $\pm$ .001
UnimodalNet	**.74 $\pm$ .01**	**.86 $\pm$ .003**	**.97 $\pm$ .002**	**.97 $\pm$ .00**	**1 $\pm$ 0**	**.06 $\pm$ .005**
CORN	.76 $\pm$ .01	–	–	–	–	–
ORCU	.84 $\pm$ .01	.81 $\pm$ .01	**.97 $\pm$ .002**	**.97 $\pm$ .003**	**1 $\pm$ 0**	.22 $\pm$ .003
UnicornNet	.75 $\pm$ .04	.84 $\pm$ .02	**.97 $\pm$ .003**	**.97 $\pm$ .002**	**1 $\pm$ 0**	**.06 $\pm$ .01**

5.5 Ablation Study

Calibration We evaluated the effect of the calibration phase on ECE. As shown in Table 4, removing the calibration phase increases ECE, indicating that the model's outputs are not well-calibrated due to the OT properties discussed earlier. This highlights the importance of incorporating the calibration step to accurately reflect the model's confidence in its predictions. Figure 4 illustrates this on Retina MNIST: OT yields overconfident, over-peaked predictions, whereas σ-scaling aligns confidence with accuracy and spreads the distribution while leaving the mode (and thus MAE) unchanged.

Temperature Scaling A widely used method for post-hoc calibration is global Temperature Scaling (TS) (Guo et al., 2017), which rescales the logits of the trained model with a single learned temperature. Table 5 compares TS with our σ-scaling (Sec. 4.3) across six image datasets. Although TS can improve

calibration, our σ-scaling attains lower or competitive ECE, as it adapts a per-instance scale of the unimodal truncated-normal output rather than applying a single global temperature.

Table 4: Effect of Calibration on ECE

Dataset	Calibration	ECE ($b = 10$) ↓
Retina MNIST	✓	**.06 ± .01**
	✗	.38 ± .01
Adience	✓	**.07 ± .03**
	✗	.18 ± .03
AAF	✓	**.03 ± .01**
	✗	.28 ± .01
EVA	✓	**.08 ± .02**
	✗	.32 ± .02

Table 5: ECE of our model under the accuracy-preserving σ-scaling vs. global Temperature Scaling (TS), across all six image datasets. Lower is better; best per dataset in bold.

Dataset	σ-scaling (ours) ECE ↓	Temperature Scaling ECE ↓
Adience	**.07 ± .03**	.17 ± .05
Retina MNIST	**.06 ± .01**	.18 ± .01
FG-NET	**.07 ± .02**	.10 ± .04
AAF	**.03 ± .01**	.04 ± .01
SCUT-FBP5500	.06 ± .02	**.03 ± .01**
EVA	**.08 ± .02**	.10 ± .02

Optimal Transport We compared UnicornNet trained with cross-entropy vs. OT loss across five datasets. Table 6 shows that OT achieves lower or comparable MAE, validating its suitability for ordinal tasks where class order matters.

Table 6: Comparison of OT and Cross-Entropy Loss

Dataset	OT MAE ↓	CE MAE ↓
Adience	**.46 ± .05**	.47 ± .06
AAF	.42 ± .01	.42 ± .01
EVA	**.57 ± .02**	.60 ± .05
FG-NET	.31 ± .08	.31 ± .06
RetinaMNIST	**.67 ± .01**	.87 ± .10

6 Conclusion

In this manuscript, we identify several issues with current deep ordinal regression methods, notably OT's tendency toward poor calibration. We therefore presented UnicornNet, an approach for deep ordinal regression, inspired by the proportional odds model. UnicornNet utilizes an architectural mechanism for the generation of unimodal output probabilities, trained using OT objective and calibrated by an accuracy-preserving calibration process which encourages uncertainty awareness. While performing on par with and often better than other recent approaches, the presented method enjoys the benefits of *guaranteed* unimodal and well-calibrated output probabilities.

Limitations UnicornNet targets ordinal tasks where unimodality is a meaningful inductive bias. In domains where the true label distribution is genuinely multi-modal or concentrated at the extremes (e.g., certain rating tasks), enforcing unimodality may be inappropriate. Moreover, the truncated-normal parameterization defines a restricted family of unimodal distributions: it captures symmetric and moderately skewed shapes, but may not represent strong skewness, sharp cut-offs, or heavy-tailed asymmetric patterns. Extending the approach to richer families, such as mixtures or asymmetric parameterizations, is a natural direction for future work.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Evaluation measures for ordinal regression. In *2009 Ninth international conference on intelligent systems design and applications*, pp. 283–287. IEEE, 2009.

Sergio A Balanya, Juan Maroñas, and Daniel Ramos. Adaptive temperature scaling for robust calibration of deep neural networks. *Neural Computing and Applications*, 36(14):8073–8095, 2024.

Christopher Beckham and Christopher Pal. Unimodal probability distributions for deep ordinal classification. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 411–419, 2017.

Soufiane Belharbi, Ismail Ben Ayed, Luke McCaffrey, and Eric Granger. Non-parametric uni-modality constraints for deep ordinal classification. *arXiv preprint arXiv:1911.10720*, 2019.

Axel Berg, Magnus Oskarsson, and Mark O'Connor. Deep ordinal regression with label diversity. *arXiv preprint arXiv:2006.15864*, 2020.

Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1): 1–3, 1950.

Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140:325–331, 2020.

Jaime S Cardoso, Ricardo Cruz, and Tomé Albuquerque. Unimodal distributions for ordinal regression. *arXiv preprint arXiv:2303.04547*, 2023.

Jianlin Cheng, Zheng Wang, and Gianluca Pollastri. A neural network approach to ordinal regression. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1279–1284. IEEE, 2008.

Jingchun Cheng, Yali Li, Jilong Wang, Le Yu, and Shengjin Wang. Exploiting effective facial patches for robust gender recognition. *Tsinghua Science and Technology*, 24(3):333–345, 2019.

Wei Chu, Zoubin Ghahramani, and Christopher KI Williams. Gaussian processes for ordinal regression. *Journal of machine learning research*, 6(7), 2005.

Jacob Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213, 1968.

Joaquim F Pinto da Costa, Hugo Alonso, and Jaime S Cardoso. The unimodal model for the classification of ordinal data. *Neural Networks*, 21(1):78–91, 2008.

Alexander Philip Dawid and Monica Musio. Theory and applications of proper scoring rules. *Metron*, 72(2): 169–183, 2014.

Coen De Vente, Pieter Vos, Matin Hosseinzadeh, Josien Pluim, and Mitko Veta. Deep learning regression for prostate cancer detection and grading in bi-parametric MRI. *IEEE Transactions on Biomedical Engineering*, 68(2):374–383, 2020.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

Raul Diaz and Amit Marathe. Soft labels for ordinal regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4738–4747, 2019.

Eran Eidinger, Roee Enbar, and Tal Hassner. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, 9(12):2170–2179, 2014.

Khaled ELKarazle, Valliappan Raman, and Patrick Then. Facial age estimation using machine learning techniques: An overview. *Big Data and Cognitive Computing*, 6(4):128, 2022.

Yasushi Esaki, Akihiro Nakamura, Keisuke Kawano, Ryoko Tokuhisa, and Takuro Kutsuna. Accuracy-preserving calibration via statistical modeling on probability simplex. *arXiv preprint arXiv:2402.13765*, 2024.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. *Advances in neural information processing systems*, 28, 2015.

Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2002–2011, 2018.

Yanwei Fu, Timothy M Hospedales, Tao Xiang, Shaogang Gong, and Yuan Yao. Interestingness prediction by robust learning to rank. In *European conference on computer vision*, pp. 488–503. Springer, 2014.

Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng. Deep label distribution learning with label ambiguity. *IEEE Transactions on Image Processing*, 26(6):2825–2838, 2017.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Le Hou, Chen-Ping Yu, and Dimitris Samaras. Squared earth mover's distance-based loss for training deep neural networks. *arXiv preprint arXiv:1611.05916*, 2016.

Chen Kang, Giuseppe Valenzise, and Frédéric Dufaux. Eva: An explainable visual aesthetics dataset. In *Joint Workshop on Aesthetic and Technical Quality Assessment of Multimedia and Media Analytics for Societal Trends*, pp. 5–13, 2020.

Daehwan Kim, Haejun Chung, and Ikbeom Jang. Calibration of ordinal regression networks, 2025. URL `https://arxiv.org/abs/2410.15658`.

Kyoung-jae Kim and Hyunchul Ahn. A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach. *Computers & Operations Research*, 39(8):1800–1811, 2012.

Lucas Kook, Lisa Herzog, Torsten Hothorn, Oliver Dürr, and Beate Sick. Deep and interpretable regression models for ordinal outcomes. *arXiv preprint arXiv:2010.08376*, 2020.

Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32, 2019.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Ling Li and Hsuan-Tien Lin. Ordinal regression by extended binary classification. *Advances in neural information processing systems*, 19, 2006.

Qiang Li, Jingjing Wang, Zhaoliang Yao, Yachun Li, Pengju Yang, Jingwei Yan, Chunmao Wang, and Shiliang Pu. Unimodal-concentrated loss: Fully adaptive label distribution learning for ordinal regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20513–20522, 2022.

Wanhua Li, Xiaoke Huang, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Learning probabilistic ordinal embeddings for uncertainty-aware regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13896–13905, 2021.

Lingyu Liang, Luojun Lin, Lianwen Jin, Duorui Xie, and Mengru Li. Scut-fbp5500: A diverse benchmark dataset for multi-paradigm facial beauty prediction. In *2018 24th International conference on pattern recognition (ICPR)*, pp. 1598–1603. IEEE, 2018.

Xiaofeng Liu, Yang Zou, Yuhang Song, Chao Yang, Jane You, and BV K Vijaya Kumar. Ordinal regression with neuron stick-breaking for medical diagnosis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 0–0, 2018a.

Xiaofeng Liu, Xu Han, Yukai Qiao, Yi Ge, Site Li, and Jun Lu. Unimodal-uniform constrained wasserstein training for medical diagnosis. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0, 2019a.

Xiaofeng Liu, Fangfang Fan, Lingsheng Kong, Zhihui Diao, Wanqing Xie, Jun Lu, and Jane You. Unimodal regularized neuron stick-breaking for ordinal classification. *Neurocomputing*, 388:34–44, 2020.

Yanzhu Liu, Adams Wai-Kin Kong, and Chi Keong Goh. Deep ordinal regression based on data relationship for small datasets. In *IJCAI*, pp. 2372–2378, 2017.

Yanzhu Liu, Adams Wai Kin Kong, and Chi Keong Goh. A constrained deep neural network for ordinal regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 831–839, 2018b.

Yanzhu Liu, Fan Wang, and Adams Wai Kin Kong. Probabilistic deep ordinal regression based on gaussian processes. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5301–5309, 2019b.

Peter McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 42(2):109–127, 1980.

Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output CNN for age estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4920–4928, 2016.

Hongyu Pan, Hu Han, Shiguang Shan, and Xilin Chen. Mean-variance loss for deep age estimation from a face. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5285–5294, 2018.

Xintong Shi, Wenzhi Cao, and Sebastian Raschka. Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. *Pattern Analysis and Applications*, 26(3):941–955, 2023.

Charles Spearman. The proof and measurement of association between two things. 1961.

Victor Manuel Vargas, Pedro Antonio Gutiérrez, and Cesar Hervas-Martinez. Cumulative link models for deep ordinal classification. *Neurocomputing*, 401:48–58, 2020.

TV Vishnu, Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, et al. Data-driven prognostics with predictive uncertainty estimation using ensemble of deep ordinal regression models. *International Journal of Prognostics and Health Management*, 10(4), 2019.

Cheng Wang. Calibration in deep learning: A survey of the state-of-the-art. *arXiv preprint arXiv:2308.01222*, 2023.

Patrick Wienholt, Alexander Hermans, Firas Khader, Behrus Puladi, Bastian Leibe, Christiane Kuhl, Sven Nebelung, and Daniel Truhn. An ordinal regression framework for a deep learning based severity assessment for chest radiographs. *arXiv preprint arXiv:2402.05685*, 2024.

Ryoya Yamasaki. Unimodal likelihood models for ordinal data. 2022.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Yuzhe Yang, Kaiwen Zha, Yingcong Chen, Hao Wang, and Dina Katabi. Delving into deep imbalanced regression. In *International conference on machine learning*, pp. 11842–11851. PMLR, 2021.

Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. Rank-n-contrast: Learning continuous representations for regression. *Advances in Neural Information Processing Systems*, 36:17882–17903, 2023.

A Proofs of Lemmas

A.1 Proof of Lemma 1

Proof. Let p_i, p_{i+1} be the output probabilities of two adjacent classes, and let $-1 = \alpha_0, \ldots, \alpha_k = 1$ 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)

$$\int_{B_{i,2}} f(x)dx > \int_{B_{i+1,1}} f(x)dx. \tag{7}$$

In addition, observe that

$$\begin{aligned}\int_{B_{i,1}} f(x)dx &= \int_0^a f(\mu + x)dx \\ &> \int_0^a f(\mu + 2b + x)dx \\ &= \int_{B_{i+1,2}} f(x)dx. \end{aligned} \tag{8}$$

Adding up Equations. (7) and (8), 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.

□

A.2 Proof of Lemma 2

Proof. Let $-1 = \alpha_0 < \alpha_1 < \ldots < \alpha_k = 1$ be the thresholds. In the proof of Lemma 1, we demonstrated that for any $i \in \{1, \ldots, k\}$, 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 $\alpha_0, \ldots, \alpha_k$ 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. □

A.3 Proof of Proposition 2

Proof. Fix an input $\boldsymbol{x}$ and write $\boldsymbol{\pi} = \boldsymbol{\pi}(\boldsymbol{x})$. We prove the claim for an arbitrary cost function c.

Step 1 (the expected OT loss is linear). Since $\boldsymbol{e_Y}$ is a Dirac on Y, Eq. (5) gives $\text{OT}(\boldsymbol{p}, \boldsymbol{e_Y}) = \sum_i p_i\, c(i, Y)$ for every realization of Y. Taking the expectation over $Y \sim \boldsymbol{\pi}$ and exchanging the finite sum with the expectation,

$$R(\boldsymbol{p}) = \sum_i p_i\, \bar{c}_i, \qquad \bar{c}_i := \mathbb{E}_{Y|\boldsymbol{X}}[c(i, Y) \mid \boldsymbol{x}]\,,$$

so R is linear in $\boldsymbol{p}$, where $\bar{c}_i$ is the expected cost of predicting class i.

Step 2 (the minimizer is a one-hot vector). A linear function on the simplex Δ^k attains its minimum at a vertex. The vertices are the one-hot vectors $\boldsymbol{e_1}, \ldots, \boldsymbol{e_k}$, with $R(\boldsymbol{e_i}) = \bar{c}_i$. Hence $\min_{\boldsymbol{p} \in \Delta^k} R(\boldsymbol{p}) = \min_i \bar{c}_i$, attained at $\boldsymbol{e_{i^\star}}$ for any $i^\star \in \arg\min_i \bar{c}_i$. The minimizer is unique exactly when $\arg\min_i \bar{c}_i$ is a single class; otherwise the minimizers are the convex combinations of the tied one-hot vectors.

Step 3 (identifying $i^\star$). The minimizing class is the familiar one: for $c(i, j) = |i - j|$, $\bar{c}_i = \mathbb{E}_Y|i - Y|$ is minimized at a median of $\boldsymbol{\pi}$, and for $c(i, j) = (i - j)^2$, $\bar{c}_i = \mathbb{E}_Y(i - Y)^2$ is minimized at the class nearest the mean $\mathbb{E}_Y[Y]$.

Step 4 (the calibrated output is not optimal). Comparing the calibrated output against the optimum,

$$R(\boldsymbol{\pi}) = \sum_i \pi_i \, \bar{c}_i \;\geq\; \min_i \bar{c}_i = R(\boldsymbol{e}_{i^\star}),$$

with equality iff $\boldsymbol{\pi}$ places all its mass on $\arg\min_i \bar{c}_i$. When that set is a single class, equality forces $\boldsymbol{\pi} = \boldsymbol{e}_{i^\star}$, i.e. $\boldsymbol{\pi}$ is itself a one-hot vector. Thus, unless $\boldsymbol{\pi}$ is already concentrated on the optimal class, OT drives the output to the peaked $\boldsymbol{e}_{i^\star}$ rather than the calibrated $\boldsymbol{\pi}(\boldsymbol{x})$, exactly the behavior the example anticipated. □

B Additional Experiments

B.1 Comparing UnicornNet Against CORN Baselines on the CORN Datasets

In Table 7 we provide the MAE results of our method trained on AFAD and Fireman dataset additionally to the results from Shi et al. (2023) for CE-NN, OR-NN, CORAL and CORN, where we borrowed also their results. We train our model with the same backbone, 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. (2023). We train our method five times with different random initialization seeds. Our method outperforms the baselines on both datasets.

Table 7: 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
UnicornNet	**2.645 ± 0.02**	**0.755 ± 0.003**

B.2 Comparing UnicornNet 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 8, indicate that the baseline achieves near-unimodal distributions (80% unimodality) on the Fireman dataset but does not guarantee perfect unimodality. Additionally, the MAE of this baseline is higher compared to UnicornNet. 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 8: Performance Comparison on Balanced Datasets (AFAD and Fireman)

Method	**AFAD (balanced)**		**Fireman (balanced)**	
	MAE ↓	**Unimodality ↑**	**MAE ↓**	**Unimodality ↑**
UnicornNet	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

B.3 Computational Cost

In this section, the computational cost associated with both the backbone training and the calibration phase is analyzed. The calibration stage introduces minimal overhead in comparison to the primary training phase. Specifically, it involves the optimization of only the scale head parameters $\boldsymbol{\theta}_\sigma$, while the backbone model remains fixed throughout. Since $\boldsymbol{\theta}_\sigma$ constitutes a parameter set that is several orders of magnitude smaller than that of the backbone, and no gradient computation or backpropagation is conducted through the backbone, the additional training cost incurred by the calibration phase is negligible.

To empirically validate the efficiency of the calibration phase, Table 9 presents the wall-clock training durations for both phases across three datasets. The results demonstrate that Phase 2 consistently requires substantially less time than Phase 1.

Table 9: Wall-Clock Training Time Comparison Between Phase 1 (OT-Based Training) and Phase 2 (Accuracy-Preserving Calibration)

Dataset	Phase	Time (HH:mm)
EVA	Phase 1	04:03
	Phase 2	**00:47**
AAF	Phase 1	04:44
	Phase 2	**00:37**
FGNET	Phase 1	02:14
	Phase 2	**00:35**

B.4 Hyperparameter Sensitivity

Overall, no unusual sensitivity was observed during training. Among the hyperparameters, the cost function used in the OT loss plays the most critical role. This cost function encodes prior knowledge about the ordinal relationship between classes and directly influences the magnitude of the gradient updates – larger cost values for distant class pairs amplify the penalization of large errors.

Empirical evidence suggests that selecting a cost function with linear structure (i.e., $m = 1$) yields superior performance across multiple evaluation metrics. To further investigate the effect of this choice, an additional experiment was conducted on the FG-NET dataset, comparing the performance of two models with different cost function hyperparameters: $m = 1$ and $m = 2$. The results, summarized in Table 10, demonstrate that a quadratic cost ($m = 2$) leads to degraded performance, confirming the importance of selecting an appropriate cost structure for the OT loss.

Table 10: Sensitivity of the Cost Metric Parameter m on Performance Metrics (FG-NET Dataset)

Method	MAE ↓	OOA ↑	Spearman ↑	QWK ↑	% Unimodal ↑	ECE (b=10) ↓
UnicornNet ($m = 1$)	**.31 ± .08**	**.99 ± .02**	**.87 ± .03**	**.88 ± .01**	**1 ± 0**	**.07 ± .02**
UnicornNet ($m = 2$)	.35 ± .07	.98 ± .01	.85 ± .03	.85 ± .02	1 ± 0	.12 ± .03

B.5 Statistical Significance

We assess the significance of the results in Table 2. Following the protocol of Demšar (2006) for comparisons across multiple datasets, we report, on the per-dataset means of the six image benchmarks, the Wilcoxon signed-rank test and a paired t-test (Table 11), where W/T/L counts datasets in which UnicornNet is better/tied/worse and, with $n = 6$, the smallest attainable Wilcoxon p is 0.0312. We also report per-dataset t-tests (Tables 12 and 13). Lower is better for both metrics; $p < 0.05$ is shown in bold.

Table 11: Across datasets: UnicornNet vs. each baseline.

	MAE			**ECE**		
Baseline	W/T/L	Wilcoxon	t-test	W/T/L	Wilcoxon	t-test
ORCU	5/0/1	.063	.052	4/0/2	.56	.32
SORD	6/0/0	**.031**	**.024**	5/0/1	.063	.056
Liu	6/0/0	**.031**	.076	6/0/0	**.031**	**.004**
Beckham	6/0/0	**.031**	**.007**	6/0/0	**.031**	**.002**
CORN	6/0/0	**.031**	**.004**	– (no ECE)		
POM	6/0/0	**.031**	**.034**	6/0/0	**.031**	**.002**
UnimodalNet	6/0/0	**.031**	**.002**	6/0/0	**.031**	**.002**
DLDL	6/0/0	**.031**	**.014**	6/0/0	**.031**	**.002**

Table 12: Per-dataset MAE significance (p): UnicornNet vs. each baseline.

Dataset	ORCU	SORD	Liu	Beckham	CORN	POM	UnimNet	DLDL
Adience	.52	.86	.85	.35	**.028**	.39	.13	.49
RetinaMNIST	.070	**<.001**	.52	**<.001**	**.004**	**<.001**	**.003**	**.003**
FG-NET	.18	.34	.80	**.026**	.25	.39	.13	.13
AAF	.16	.12	.25	**.031**	**.004**	**.029**	**.044**	**<.001**
SCUT-FBP	.14	**<.001**	**.003**	**.047**	**<.001**	**.026**	**.004**	**<.001**
EVA	**.004**	**.037**	**<.001**	**.042**	**<.001**	**.009**	**<.001**	**.001**

Table 13: Per-dataset ECE significance (p): UnicornNet vs. each baseline (CORN reports no ECE).

Dataset	ORCU	SORD	Liu	Beckham	POM	UnimNet	DLDL
Adience	.32	**.005**	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**
RetinaMNIST	**<.001**	**<.001**	**<.001**	**<.001**	**<.001**	**.002**	**<.001**
FG-NET	.096	**<.001**	.51	**<.001**	**<.001**	**.001**	**<.001**
AAF	.66	**<.001**	**<.001**	**.002**	**<.001**	**<.001**	**<.001**
SCUT-FBP	.29	.10	**<.001**	**.021**	**<.001**	**<.001**	**<.001**
EVA	.79	.31	**<.001**	.055	**<.001**	**<.001**	**<.001**

C Technical Details

Table 14 shows the technical details for the experiments on the real world benchmark datasets reported in this manuscript.

Table 14: 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}

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 of the main paper 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 dataset[1]. For the accuracy-preserving

[1] `https://github.com/GilLevi/AgeGenderDeepLearning/tree/master/Folds/train_val_txt_files_per_fold`

Table 15: 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

calibration phase, we used early stopping and the number of epochs and learned parameters were changed for each dataset, as shown in Table 15.

D Datasets

Table 16: 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
Fireman	tabular	30922	1628	8138	16

Tabel 16 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.

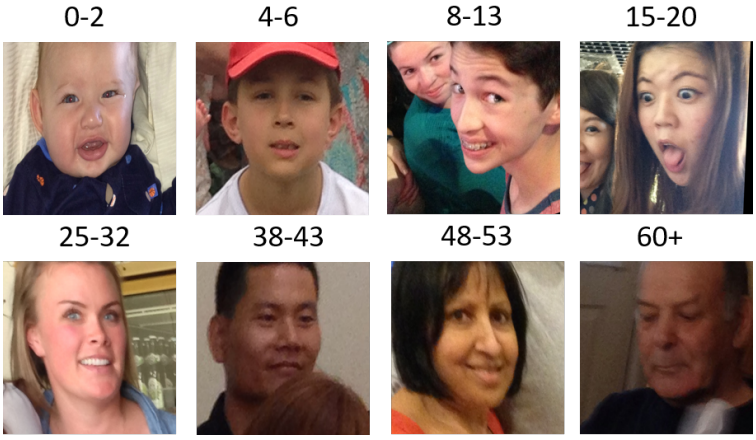

Figure 5: Examples From the Adience Dataset. The age category is indicated above each image.

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

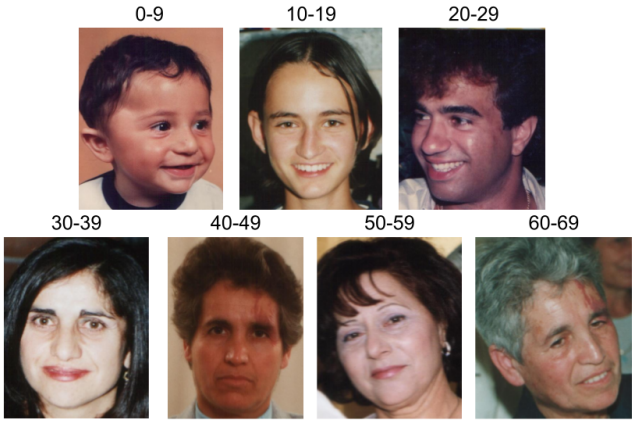

Figure 6: Examples From the FG-Net Dataset. Age classes are indicated above each image.

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 proved by the contributors and were used as-is.

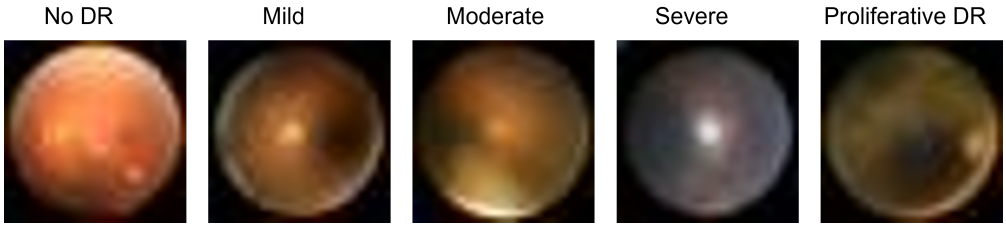

Figure 7: Examples From the Retina MNIST Dataset. Diabetic Retinopathy classes are indicated above.

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.

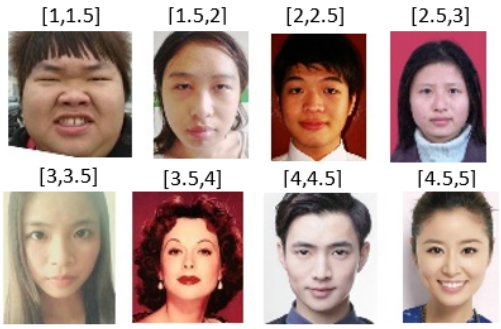

Figure 8: Examples From the SCUT-FBP5500 Dataset. Beauty classes are indicated above each image.

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.

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.

Figure 9: Examples From the EVA Dataset. Aesthetics classes are indicated above each image.

Figure 10: Examples From the AAF Dataset. Age classes are indicated above each image.

AFAD (The Asian Face data)[2] 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. (2023), we use a balanced version of the AFAD dataest[3] with 13 age labels in the age range of 18-30 years, resulting in total 60K samples.

Fireman [4] tabular dataset that contains 40,768 instances, 10 numeric features, and an ordinal response variable with 16 categories. Just like Shi et al. (2023), we use a balanced version of this dataset[5] consisting of 2,543 instances per class and 40,688 from the 16 ordinal classes in total. The train, validation, and test splits are the same as in Shi et al. (2023)

Figures 6, 9, 10, 8 show examples from the FG-Net, EVA, AAF and SCUT-FBP5500 datasets, respectively.

E 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.

[2] https://github.com/afad-dataset/tarball
[3] https://github.com/Raschka-research-group/corn-ordinal-neuralnet/tree/main/datasets/afad
[4] https://github.com/gagolews/ordinal_regression_data
[5] https://github.com/Raschka-research-group/corn-ordinal-neuralnet/tree/main/datasets/firemen