
Neural Collapse in Cumulative Link Models for Ordinal Regression: An Analysis with Unconstrained Feature Model

Chuang Ma¹ Tomoyuki Obuchi^{1,2} Toshiyuki Tanaka¹

¹Kyoto University, ²RIKEN AIP,
{ma.chuang.52h@st, obuchi@i, tt@i}.kyoto-u.ac.jp

Abstract

A phenomenon known as “Neural Collapse (NC)” in deep classification tasks, in which the penultimate-layer features and the final classifiers exhibit an extremely simple geometric structure, has recently attracted considerable attention, with the expectation that it can deepen our understanding of how deep neural networks behave. The Unconstrained Feature Model (UFM) has been proposed to explain NC theoretically, and there emerges a growing body of work that extends NC to tasks other than classification and leverages it for practical applications. In this study, we investigate whether a similar phenomenon arises in deep Ordinal Regression (OR) tasks, via combining the cumulative link model for OR and UFM. We show that a phenomenon we call Ordinal Neural Collapse (ONC) indeed emerges and is characterized by the following three properties: (ONC1) all optimal features in the same class collapse to their within-class mean when regularization is applied; (ONC2) these class means align with the classifier, meaning that they collapse onto a one-dimensional subspace; (ONC3) the optimal latent variables (corresponding to logits or preactivations in classification tasks) are aligned according to the class order, and in particular, in the zero-regularization limit, a highly local and simple geometric relationship emerges between the latent variables and the threshold values. We prove these properties analytically within the UFM framework with fixed threshold values and corroborate them empirically across a variety of datasets. We also discuss how these insights can be leveraged in OR, highlighting the use of fixed thresholds.

1 Introduction

In classification tasks on balanced datasets, it has been observed that, after sufficient training, the outputs (or features) of the penultimate layer and the final classifier weights in sufficiently expressive Deep Neural Networks (DNNs) exhibit a remarkably simple symmetric structure. Papayan et al. [2020] conducted thorough experiments across architectures and datasets to corroborate this phenomenon, and identified its four intertwined signatures, which are summarized as **Neural Collapse (NC)**: (NC1) all feature vectors from the same class converge exactly onto their shared class mean, extinguishing within-class variance; (NC2) once these class means are recentered at the global mean, they occupy the vertices of a maximally symmetric Simplex Equiangular Tight Frame (Simplex ETF); (NC3) each classifier weight vector becomes parallel to its corresponding class mean vector, reflecting a self-dual alignment; (NC4) the network simply classifies by nearest class mean. NC, being considered to offer a valuable clue toward understanding DNNs, has inspired a number of theoretical studies [Mixon et al., 2022, Fang et al., 2021, Zhu et al., 2021, E and Wojtowysch, 2020, Lu and Steinerberger, 2022] which introduced **Unconstrained Feature Model (UFM)** being a central theoretical tool in this context.

UFM has allowed extending NC to broader problem settings and facilitated the analytical investigation of its properties [Zhou et al., 2022, Thrampoulidis et al., 2022, Dang et al., 2023, 2024, Li et al., 2024a]. For example, Andriopoulos et al. [2024] applied this framework to multivariate regression, finding a phenomenon called Neural Regression Collapse (NRC) in which features collapse to a target subspace and weight vectors align accordingly. The proliferation of these extensions suggests that NC is a universal phenomenon in DNNs.

Beyond classification and regression, there is a task called **Ordinal Regression (OR)** which aims to solve prediction tasks where labels are discrete categories with a natural order. Unlike classification, not all misclassifications are equally wrong in OR tasks; unlike regression, label values in OR do not bear quantitative information and only their ordering relationship is essential. A common approach to OR problems is to use threshold models [Verwaeren et al., 2012] which assume that an unobserved continuous latent variable generates the observed ordinal response: the map to latent space is traditionally assumed to be linear with respect to (w.r.t.) input datapoints [Herbrich et al., 2000]. Even within this framework, more challenging scenarios can be addressed by applying an appropriate transformation (feature extractor) to the input datapoints, and the effectiveness of DNNs as feature extractors has been reported [Dorado-Moreno et al., 2012, Vargas et al., 2020]. The latent variable in the threshold models occupies the same role as the logit in classification networks, prompting the question of whether phenomena similar to NC may also emerge in the feature space of OR.

To address this question, in this paper we explore phenomena analogous to NC within the context of OR. To that end, we adopt the Cumulative Link Model (CLM) [Agresti, 2010]—a classical type of threshold models—and analyze it in combination with UFM. As a result, we find that such a phenomenon indeed occurs when the ℓ_2 regularization is applied, and we name it Ordinal Neural Collapse (ONC). ONC is characterized by the following three properties:

- (ONC1) Within-class Mean Collapse:** *all optimal features in the same class collapse to their within-class mean.*
- (ONC2) Collapse to One-Dimensional Subspace:** *these class means align with the classifier, meaning that they collapse onto a one-dimensional subspace.*
- (ONC3) Collapse to Ordinal Structure:** *the optimal latent variables are aligned according to the class order, and in particular, in the zero-regularization limit, a highly local and simple relationship emerges between the latent variables and the threshold values.*

We note that this result is obtained under the assumption that all the threshold values of CLM are fixed. Although it is not necessarily a standard assumption in recent studies, we argue that it is necessary for the emergence of ONC and, moreover, can be a practically meaningful assumption. A discussion about this point will be given later in Section 6.

We also validated ONC through experiments using five imbalanced ordinal datasets and a DNN architecture. The result provides clear empirical evidence of ONC under fixed threshold values. Furthermore, our experiments with learnable thresholds still exhibit ONC, implying its robustness.

2 Related work

We cite here only recent results that are particularly relevant to the present work.

UFM and related issues. UFM and the related models were proposed almost concurrently in a number of pieces of work [Mixon et al., 2022, Fang et al., 2021, Zhu et al., 2021, E and Wojtowysch, 2020, Lu and Steinerberger, 2022]. The core idea of UFM lies in decoupling the features from the data by treating the output of a specific layer, typically the penultimate layer, as free learnable variables, while explicitly modeling the nonlinear functions and weight vectors in the subsequent layers. This simplifying assumption enables us to analytically derive nontrivial results. Even in such a simplified model, the analysis can remain nontrivial. For example, UFM generally admits multiple local minima in its loss landscape. Among them, it was shown by Zhu et al. [2021] that only the global minimum exhibits the NC structure in the case of balanced classification. Departing from the typical analysis of the single-layer UFM, the first investigation of UFM with multiple layers was carried out by Tirer and Bruna [2022], finding that NC still emerges as the unique global optimum. Furthermore, to understand the phenomenon called Deep NC, in which NC propagates not only to the

final layer but also to intermediate layers [Hui et al., 2022, He and Su, 2023, Rangamani et al., 2023, Parker et al., 2023, Masarczyk et al., 2023], Súkeník et al. [2023] extended UFM to multiple nonlinear layers and proved that, in the case of binary classification, Deep NC emerges as the unique global optimum. Their follow-up study [Súkeník et al., 2024] demonstrated that, in multi-layer architectures, increasing the number of classes causes Deep NC to cease being the optimal solution.

NC extensions under varied settings on classification. In classification, the NC framework has been adapted to different settings from the balanced case. Zhou et al. [2022] showed that cross-entropy, focal loss, label smoothing, and even mean squared error lead to the same NC geometry using UFM. Thrampoulidis et al. [2022] and Hong and Ling [2024] found that with cross-entropy loss and ℓ_2 regularization, class imbalance does not prevent NC1 but the global geometry generalizes from a Simplex ETF to a more general structure. In extremely imbalanced cases, Fang et al. [2021] found a “minority collapse” phenomenon where minority-class features collapse onto a single point. Dang et al. [2023] proved in deep linear UFM that every global minimizer forms orthogonal directions whose magnitudes scale proportionally to class sample sizes, and Dang et al. [2024] proved essentially the same statement for rectified linear unit (ReLU) UFM.

NC extensions beyond classification. Li et al. [2024a] extended NC to multi-label classification by showing that multi-label embeddings lie in the linear span of label-means. Andriopoulos et al. [2024] generalized NC to multivariate regression to find NRC. Wu and Pappan [2024] introduced the concept of “linguistic collapse” in large-scale language models, showing that token embeddings tend toward an approximately uniform distribution on the hypersphere as model scale increases. Furthermore, Nguyen et al. [2024] demonstrated that NC-style embedding collapse also occurs in diffusion models. NC-like phenomena have also been observed in self-supervised learning [Ben-Shaul et al., 2023] and in transfer learning [Galanti et al., 2022, Li et al., 2024b]: the latter studies discussed the relationship between the degree of NC and transfer performance, and also proposed some strategy for leveraging NC insights to improve generalization performance.

3 Formulation

OR. An OR task is formulated as follows. Let $\mathcal{X}$ be an input space and $\mathcal{Y} = \{1, 2, \dots, Q\}$ be an ordered label set with ordering $1 < 2 < \dots < Q$. Given a training set $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ with $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, our goal is to learn an order-respecting mapping $r : \mathcal{X} \rightarrow \mathcal{Y}$. For each label q , we let $D_q = \{(\mathbf{x}_i, y_i) \in D \mid y_i = q\}$ and n_q its size, so that $\sum_{q=1}^Q n_q = N$ holds.

CLM. To express the ordinal structure, threshold models introduce a latent variable $z \in \mathbb{R}$ and also a strictly ordered set of “thresholds” $\mathbf{b} = (b_0, b_1, \dots, b_Q)$ which partitions the z -axis. One typically assumes $(b_0, b_Q) = (-\infty, \infty)$ to partition $\mathbb{R}$ properly, and thus each interval is uniquely associated with one category via the decision rule $y = q \iff z \in (b_{q-1}, b_q]$.

In CLMs, the probability of a specific category is expressed through a cumulative probability $P(y \leq q \mid z)$ conditioned on the latent variable z , which is modeled by using a strictly monotone inverse link function $g : \mathbb{R} \rightarrow (0, 1)$ as

$$P(y \leq q \mid z) = g(b_q - z). \quad (1)$$

There are several typical choices for g , including the logistic function $g(x) = (1 + e^{-x})^{-1}$, the normal cumulative distribution function (CDF) $g(x) = \Phi(x) = \int_{-\infty}^x e^{-\frac{1}{2}z^2} dz / \sqrt{2\pi}$, and the Gumbel CDF $g(x) = 1 - e^{-e^x}$, which correspond to the logit, probit, and clog-log models, respectively.

An input datapoint $\mathbf{x}$ is transformed to a value z in the latent space through a certain map. When using a feature extractor such as DNNs [Vargas et al., 2020], the map is expressed as

$$z = f_{\mathbf{w}, \theta}(\mathbf{x}) = \mathbf{w}^\top \mathbf{h}_\theta(\mathbf{x}), \quad (2)$$

where $\mathbf{w} \in \mathbb{R}^p$ and $\mathbf{h}_\theta : \mathcal{X} \rightarrow \mathbb{R}^p$ are the classifier weight vector and feature extractor, respectively. Here, θ represents the parameters of the feature extractor.

Under the model (1), (2), since the probability that y belongs to class q given z is expressed as $P(y = q \mid z) = P(y \leq q \mid z) - P(y \leq q - 1 \mid z)$, the empirical negative log-likelihood given the

dataset $\{D_q\}_{q=1}^Q$ becomes

$$\mathcal{L}_{\text{NLL}}(\mathbf{w}, \theta, \mathbf{b}) = \frac{1}{N} \sum_{q=1}^Q \sum_{(\mathbf{x}_i, y_i) \in D_q} L(z_i, b_{q-1}, b_q), \quad z_i = f_{\mathbf{w}, \theta}(\mathbf{x}_i), \quad (3)$$

where we let $L(z, a, b) := -\log[g(b - z) - g(a - z)]$. As in recent practices using DNNs, we consider the ℓ_2 regularization on the parameters:

$$\mathcal{R}(\mathbf{w}, \theta) = \frac{\lambda_w}{2} \|\mathbf{w}\|_2^2 + \frac{\lambda_\theta}{2} \|\theta\|_2^2. \quad (4)$$

The overall optimization problem is therefore

$$\min_{\mathbf{w}, \theta} (\mathcal{L}_{\text{NLL}}(\mathbf{w}, \theta, \mathbf{b}) + \mathcal{R}(\mathbf{w}, \theta)). \quad (5)$$

UFM for CLM-based OR. In the single-layer UFM, the feature vector $\mathbf{h}_\theta(\mathbf{x}_i)$ itself is treated as a free learnable variable. As a result, for each datapoint $\mathbf{x}_i$, a free variable $\mathbf{h}_i$ is associated. For notational simplicity, we relabel this variable as $\mathbf{h}_{q,i}$, where q indexes the class and $i = 1, \dots, n_q$ indexes the datapoints within D_q . The regularization on the parameter θ is assumed to be converted to that on $H := (\mathbf{h}_{q,i})_{q,i}$. UFM thus allows us to convert (5) into

$$\min_{\mathbf{w}, H} (\mathcal{L}_{\text{NLL,UFM}}(\mathbf{w}, H, \mathbf{b}) + \mathcal{R}_{\text{UFM}}(\mathbf{w}, H)), \quad (6)$$

where

$$\mathcal{L}_{\text{NLL,UFM}}(\mathbf{w}, H, \mathbf{b}) = \frac{1}{N} \sum_{q=1}^Q \sum_{i=1}^{n_q} L(\mathbf{w}^\top \mathbf{h}_{q,i}, b_{q-1}, b_q), \quad (7)$$

$$\mathcal{R}_{\text{UFM}}(\mathbf{w}, H) = \frac{\lambda_w}{2} \|\mathbf{w}\|_2^2 + \frac{\lambda_h}{2N} \sum_{q=1}^Q \sum_{i=1}^{n_q} \|\mathbf{h}_{q,i}\|_2^2. \quad (8)$$

4 Theoretical results based on UFM analysis

Let us state our main theoretical results. Thanks to the structure of our CLM and UFM, (6) can be decomposed into a multi-stage optimization as follows:

$$\min_w \left\{ \frac{\lambda_w}{2} w^2 + \frac{1}{N} \sum_{q=1}^Q \sum_{i=1}^{n_q} \min_{\mathbf{h}_{q,i}} f_q(w, \mathbf{h}_{q,i}) \right\}, \quad (9)$$

where

$$f_q(w, \mathbf{h}) = L(w \mathbf{a}^\top \mathbf{h}, b_{q-1}, b_q) + \frac{\lambda_h}{2} \|\mathbf{h}\|_2^2, \quad (10)$$

and where $w \geq 0$ and $\mathbf{a}$ are the norm of $\mathbf{w}$ and the unit vector representing the direction of $\mathbf{w}$, respectively, so that $\mathbf{w} = w\mathbf{a}$ and $\|\mathbf{a}\|_2 = 1$ hold. Since our objective function to be minimized in (6) is invariant under any orthogonal transformation $\mathbf{w} \rightarrow O\mathbf{w}, \mathbf{h} \rightarrow O\mathbf{h}, \forall O \in O(p)$, we can fix the direction $\mathbf{a}$ of $\mathbf{w}$ without loss of generality. Furthermore, we assume that the derivative g' of the inverse link function g is logarithmically concave (log-concave): some standard choices of g in OR such as the logistic function, the standard normal CDF, and the Gumbel CDF satisfy this assumption.

Under these assumptions, the conditions of ONC can be derived. Before presenting the concrete statements, we first show the following theorem.

Theorem 4.1. *Let $p(x)$ be a log-concave function on $\mathbb{R}$, and let $P(x) = \int_{-\infty}^x p(u) du$. Then, for any $a < b$, the function $\rho(x) = P(b - x) - P(a - x)$ is log-concave.*

Proof. One can write $\rho(x)$ as

$$\rho(x) = P(b - x) - P(a - x) = \int_{a-x}^{b-x} p(u) du = \int_a^b p(y + x) dy, \quad (11)$$

where we let $y = u - x$. Since $p(y + x)$ is log-concave in $\mathbb{R}^2$, one can apply Theorem A.1 with $A = [a, b]$ to conclude that $\rho(x)$ is log-concave in x . $\square$

This means that the log-concavity of g' leads to the convexity of $L(z, a, b) = -\log[g(b-z) - g(a-z)]$ w.r.t. z for any (a, b) satisfying $b > a$. One can further show that, if g' is strictly log-concave, then $L(z, a, b)$ is strictly convex in z : see Appendix A.2.

We are now ready to state the ONC theorem.

Theorem 4.2 (ONC). Assume that the inverse link function $g(x)$ defined on $\mathbb{R}$ is differentiable, and that its derivative g' is log-concave. Consider (6) with thresholds $\mathbf{b} = (b_0, b_1, \dots, b_Q)$ satisfying $b_0 < b_1 < \dots < b_Q$, and let $(\mathbf{w}^*, H^*)$ denote the global minimizer. Under the assumption $\lambda_w, \lambda_h > 0$, the following three properties hold:

(ONC1) For any class $q \in \mathcal{Y}$, the optimal features $\{\mathbf{h}_{q,i}^*\}_i$ in class q become identical:

$$\mathbf{h}_{q,i}^* = \mathbf{h}_q^*, \quad \forall i = 1, \dots, n_q.$$

In other words, the optimal features collapse to their within-class mean $\mathbf{h}_q^*$.

(ONC2) For any class q , the class mean $\mathbf{h}_q^*$ becomes parallel to $\mathbf{w}^*$, meaning that all class means collapse onto the one-dimensional subspace spanned by $\mathbf{w}^*$.

(ONC3) The optimal latent variables $z_q^* = (\mathbf{w}^*)^\top \mathbf{h}_q^*$ satisfy $z_1^* \leq z_2^* \leq \dots \leq z_Q^*$. Moreover, if g' is strictly log-concave and if $\mathbf{w}^* \neq \mathbf{0}$, then these inequalities hold strictly.

Proof. By a technical reason, we separately treat the two cases $w^* = 0$ and $w^* \neq 0$, and here provide only the derivation of ONC1 and 2, deferring the proof of ONC3 to Appendix B.

Thanks to the structure of (9), for any fixed w every $\mathbf{h}_{q,i}$ can be optimized separately from the other variables, and the objective function is identical for all $i \in \{1, \dots, n_q\}$. Its explicit form is

$$\arg \min_{\mathbf{h}_{q,i}} f_q(w, \mathbf{h}_{q,i}) = \arg \min_{\mathbf{h}} \left(L(w\mathbf{a}^\top \mathbf{h}, b_{q-1}, b_q) + \frac{\lambda_h}{2} \|\mathbf{h}\|_2^2 \right). \quad (12)$$

First suppose $w \neq 0$. Since $L(z, b_{q-1}, b_q)$ is proven to be convex in z through Theorem 4.1, $L(w\mathbf{a}^\top \mathbf{h}, b_{q-1}, b_q) =: L_q(w\mathbf{a}^\top \mathbf{h})$ is also convex in $\mathbf{h}$. Since the term $(\lambda_h/2)\|\mathbf{h}\|_2^2$ is strictly convex, the total objective function to be minimized is strictly convex w.r.t. $\mathbf{h}$. On the other hand, let $\mathbf{v}_q$ be the gradient of $L_q(w\mathbf{a}^\top \mathbf{h})$ at $\mathbf{h} = \mathbf{0}$. Thanks to the convexity of $L_q(w\mathbf{a}^\top \mathbf{h})$, one has

$$L_q(w\mathbf{a}^\top \mathbf{h}) - L_q(0) \geq \mathbf{v}_q^\top \mathbf{h}, \quad \forall \mathbf{h}, \quad (13)$$

which implies that the objective function is bounded from below:

$$L_q(w\mathbf{a}^\top \mathbf{h}) + \frac{\lambda_h}{2} \|\mathbf{h}\|_2^2 \geq L_q(0) + \frac{1}{2} \lambda_h \left\| \mathbf{h} + \frac{\mathbf{v}_q}{\lambda_h} \right\|_2^2 - \frac{1}{2\lambda_h} \|\mathbf{v}_q\|_2^2 \geq L_q(0) - \frac{1}{2\lambda_h} \|\mathbf{v}_q\|_2^2 > -\infty. \quad (14)$$

Hence, the strict convexity and the boundedness of $L_q(w\mathbf{a}^\top \mathbf{h}) + \frac{\lambda_h}{2} \|\mathbf{h}\|_2^2$ guarantee the uniqueness of the minimizer, proving ONC1. The proof of ONC2 is more straightforward. Let $\mathbf{h}_\parallel$ denote the projection of $\mathbf{h}$ on $\mathbf{a}$ and $\mathbf{h}_\perp = \mathbf{h} - \mathbf{h}_\parallel$. Then we have

$$L_q(w\mathbf{a}^\top \mathbf{h}) + \frac{\lambda_h}{2} \|\mathbf{h}\|_2^2 = L_q(w\mathbf{a}^\top \mathbf{h}_\parallel) + \frac{\lambda_h}{2} \|\mathbf{h}_\parallel\|_2^2 + \frac{\lambda_h}{2} \|\mathbf{h}_\perp\|_2^2. \quad (15)$$

Hence, the minimization of this w.r.t. $\mathbf{h}_\perp$ yields $\mathbf{h}_\perp^* = \mathbf{0}$, showing ONC2.

Next we assume $w = 0$. In this case, the dependence of the objective function on $\mathbf{h}_{q,i}$ only appears in the regularization term and the optimization thus yields $\mathbf{h}_{q,i}^* = \mathbf{0}$ for all q, i . Hence, the ONC properties appear trivially. $\square$

In contrast to ONC1 and 2, which only require the convexity of $L(z, a, b)$, ONC3 has a more quantitative information about the problem. Actually, the values of $w^*, \mathbf{z}^*$ are determined from a set of equations deduced from the stationarity condition of (9). Borrowing the terminology from statistical physics, we call this set of equations **Equations Of State (EOS)**. Analyzing EOS leads to a derivation of ONC3, but it is involved and is deferred to Appendix B. The solution of EOS exhibits some singularity at certain parameter values, and also some simple behaviors in certain limits. The next theorem summarizes these findings.

Theorem 4.3 (EOS, phase transition, and some limiting behaviors). *Consider the same situation as in Theorem 4.2. If the optimal norm value satisfies $w^* > 0$, w^* and the optimal latent variables $\mathbf{z}^*$ obey the following set of equations which we call EOS:*

$$\frac{g'(b_q - z_q^*) - g'(b_{q-1} - z_q^*)}{g(b_q - z_q^*) - g(b_{q-1} - z_q^*)} + \lambda_h \frac{z_q^*}{(w^*)^2} = 0, \quad q = 1, \dots, Q, \quad (16a)$$

$$\lambda_w w^* - \frac{\lambda_h}{(w^*)^3} \sum_{q=1}^Q \alpha_q (z_q^*)^2 = 0, \quad (16b)$$

where $\alpha_q = n_q/N$. Additionally assuming the continuity and monotonicity of w^* w.r.t. λ_h and λ_w , this EOS implies a phase transition with the phase boundary in the (λ_h, λ_w) -plane characterized by

$$\lambda_h \lambda_w = C := \sum_{q=1}^Q \alpha_q \left(\frac{g'(b_q) - g'(b_{q-1})}{g(b_q) - g(b_{q-1})} \right)^2. \quad (17)$$

Namely, for $\lambda_h \lambda_w \geq C$ the trivial solution $w^* = 0, \mathbf{z}^* = \mathbf{0}$ becomes the optimal solution to (9), while for $\lambda_h \lambda_w < C$ the nontrivial solution $w^* > 0, \mathbf{z}^* \neq \mathbf{0}$, which obey EOS, becomes the optimal one.

Moreover, this EOS admits a simple behavior emerging in the limit where the product $\lambda_h \lambda_w$ approaches zero. In that limit, $\mathbf{z}^*$ is determined by

$$g'(b_q - z_q^*) = g'(b_{q-1} - z_q^*), \quad q = 1, \dots, Q, \quad (18)$$

and one has $w^* = \Theta((\lambda_h/\lambda_w)^{1/4})$.

Proof. Applying ONC2, we have $z_q = w \mathbf{a}^\top \mathbf{h}_q$. Thus we may rewrite the squared norm $\|\mathbf{h}_q\|_2^2$ as z_q^2/w^2 and the optimization w.r.t. $\mathbf{w}$ and $(\mathbf{h}_q)_q$ in (9) can be reduced to those w.r.t. $w, \mathbf{z}$. Taking the stationarity condition w.r.t. $\mathbf{z}$ and w lead to EOS.

Next, we examine the phase transition and the phase boundary. One subtlety in analyzing the nature of the phase transition is that the trivial solution $(w^*, \mathbf{z}^*) = (0, \mathbf{0})$ does not satisfy EOS within the whole parameter region where it is optimal. However, thanks to the assumed continuity and monotonicity of w^* , exactly on the phase boundary the trivial solution must satisfy EOS. Therefore, we search for a condition under which EOS admits the trivial solution. By substituting (16a) into (16b) to eliminate the explicit dependence on w^* , we obtain the following equation:

$$\lambda_w = \frac{1}{\lambda_h} \sum_{q=1}^Q \alpha_q \left(\frac{g'(b_q - z_q^*) - g'(b_{q-1} - z_q^*)}{g(b_q - z_q^*) - g(b_{q-1} - z_q^*)} \right)^2. \quad (19)$$

Inserting the trivial solution $\mathbf{z}^* = \mathbf{0}$ into this leads to the phase boundary (17). Thanks to the assumed monotonicity, once the solution becomes the trivial one, it continues to be so above the boundary.

Finally, the limiting behavior is investigated. By inserting (16b) into (16a) to eliminate the explicit dependence on w^* , we have

$$\frac{g'(b_q - z_q^*) - g'(b_{q-1} - z_q^*)}{g(b_q - z_q^*) - g(b_{q-1} - z_q^*)} + \sqrt{\lambda_w \lambda_h} \frac{z_q^*}{\sqrt{\sum_{q'=1}^Q \alpha_{q'} (z_{q'}^*)^2}} = 0, \quad q = 1, \dots, Q. \quad (20)$$

This yields (18) when $\lambda_h \lambda_w$ is sent to zero, as long as $\mathbf{z}^* \neq \mathbf{0}$. Then, solving (16b) w.r.t. w^* , we have the scaling $w^* = \Theta((\lambda_h/\lambda_w)^{1/4})$. $\square$

The scaling $w^* = \Theta((\lambda_h/\lambda_w)^{1/4})$ means that w^* in the limit λ_h and/or $\lambda_w \rightarrow 0$ may diverge, vanish, or remain finite depending on how one takes the limit.

Equation (18) in the vanishing regularization limit is fairly striking since it provides a simple local relation between $\mathbf{b}$ and $\mathbf{z}^*$. Especially, if the inverse link function satisfies a symmetry $1 - g(x) = g(-x)$, which is the case for the logit and probit models, one has $g'(x) = g'(-x)$, and (18) thus implies

$$z_q^* = \frac{b_q + b_{q-1}}{2}. \quad (21)$$

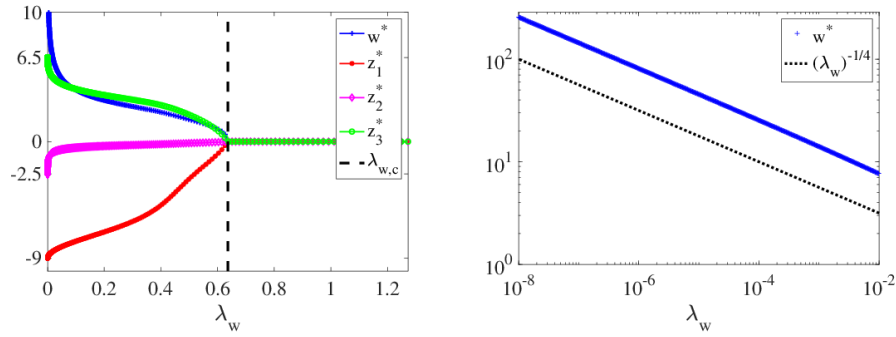


Figure 1: Solution behavior of EOS in the logit model for $Q = 3$ with $\mathbf{b} = (-10, -8, 3, 10)$ at $\lambda_h = 1$. (Left) w^* and z^* are plotted against λ_w on a linear scale. A clear phase transition appears at $\lambda_{w,c} = C/\lambda_h$ (vertical broken line), and the values of z^* in the limit $\lambda_w \rightarrow 0$ match well with the theoretical prediction ($z_q^* = (b_q + b_{q-1})/2$). (Right) w^* is plotted on a log-log scale in the small- λ_w region. A power-law divergence with exponent $-1/4$, corresponding to the scaling $w^* = \Theta((\lambda_h/\lambda_w)^{1/4})$ with fixed λ_h , is clearly observed.

This simple relation will be verified later in experiments using real-world datasets.

For illustration, we numerically solved (16) for the logit model and plotted the solution in Fig. 1. The analytical prediction about the critical point and the limiting behaviors were certainly reproduced.

5 Experiments

5.1 Experimental setting

Inverse link functions. Two symmetric inverse link functions, the logistic function and the normal CDF, which correspond to the logit and probit models, respectively, were treated in the experiment.

Datasets and neural networks. We used five tabular OR datasets with the largest number of data points—ER, LE, SW, CA, and WR—among those publicly available from Gutiérrez et al. [2016]. Thirty pre-defined training–validation splits with identical label distributions have been officially released, and we used them as-is in this study. Additionally, we conducted experiments on the UTKFace age estimation dataset [Zhang et al., 2017], which contains face images labeled with ages. We grouped ages into classes with five-year intervals. For the tabular datasets, we employed a multilayer perceptron with residual connections, while for UTKFace, we used ResNet101 and ResNet50 [He et al., 2016], and DenseNet201 [Huang et al., 2017] as backbones. The weight decay coefficient was set to small values, with the exact choice varying by condition. The motivation for this setting is that, in the small-regularization limit, a very simple result emerges as shown in (18), which facilitates experimental verification. Further details are provided in Appendix C.

Treatment of thresholds. We considered two cases: fixed and learnable thresholds.

For the fixed case, to ensure that the ignored tail probabilities are sufficiently small, the edge thresholds b_0 and b_Q were symmetrically fixed ($b_0 = -b_Q$) to sufficiently large values. The remaining thresholds were evenly spaced over the interval $[b_0, b_Q]$. Under this setting, we solved (5), where θ denotes the DNN parameters.

In the learnable case, we set $b_0 = -\infty$ and $b_Q = +\infty$ and learned b_q with $q = 1, \dots, Q-1$. To guarantee the strict ordering between the threshold values, we parameterized them with $\mathbf{s} \in \mathbb{R}^{Q-1}$ as

$$b_q(\mathbf{s}) := \sum_{j=1}^q \log(1 + e^{s_j}) - \frac{1}{Q-1} \sum_{j=1}^{Q-1} \log(1 + e^{s_j}), \quad q = 1, \dots, Q-1. \quad (22)$$

Correspondingly, we solved $\min_{\mathbf{w}, \theta, \mathbf{s}} (\mathcal{L}_{\text{NLL}}(\mathbf{w}, \theta, \mathbf{b}(\mathbf{s})) + \mathcal{R}(\mathbf{w}, \theta))$ instead of (5).

Evaluation metrics. We used two basic training metrics for evaluation: $\mathcal{L}_{\text{NLL}}$ and the mean absolute error (MAE) for label prediction, $\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|$ where $\hat{y}_i$ is the predicted label. Additionally, we evaluated classification accuracy, quadratic weighted kappa (QWK), within-1 accuracy (proportion of predictions within one class of the true label), and minimum sensitivity (worst-case per-class recall). Besides, let $\mathbf{h}_\theta(\cdot)$ denote the penultimate-layer output of our DNN, and let $\bar{\mathbf{h}}_q = (1/n_q) \sum_{(\mathbf{x}_i, y_i) \in D_q} \mathbf{h}_\theta(\mathbf{x}_i)$ and $\bar{\mathbf{h}} = (1/N) \sum_{i=1}^N \mathbf{h}_\theta(\mathbf{x}_i)$ represent the class-wise and global feature means, respectively. Using these, we introduced the following four quantitative indicators for ONC:

$$\text{ONC}_1 = \frac{(1/Q) \sum_{q=1}^Q \frac{1}{N_q} \sum_{(\mathbf{x}_i, y_i) \in D_q} \|\mathbf{h}_\theta(\mathbf{x}_i) - \bar{\mathbf{h}}_q\|_2}{(1/N) \sum_{i=1}^N \|\mathbf{h}_\theta(\mathbf{x}_i) - \bar{\mathbf{h}}\|_2}, \quad (23)$$

$$\text{ONC}_{2-1} = \frac{\sum_{q=1}^Q \|(\bar{\mathbf{h}}_q - \bar{\mathbf{h}}) - (\mathbf{u}^\top (\bar{\mathbf{h}}_q - \bar{\mathbf{h}})) \mathbf{u}\|_2^2}{\sum_{q=1}^Q \|\bar{\mathbf{h}}_q - \bar{\mathbf{h}}\|_2^2}, \quad \text{ONC}_{2-2} = 1 - \frac{|\mathbf{w}^\top \mathbf{u}|}{\|\mathbf{w}\|_2}, \quad (24)$$

$$\text{ONC}_3 = \frac{\sum_{q=1}^{Q-1} |b_q - (z_q + z_{q+1})/2|}{\sum_{q=1}^{Q-1} (b_{q+1} - b_q)}, \quad (25)$$

where $\mathbf{u}$ is the unit first principal component of $\{\bar{\mathbf{h}}_q - \bar{\mathbf{h}}\}_{q=1}^Q$. ONC_1 is the indicator for ONC1 and becomes zero when ONC1 exactly happens; ONC_{2-1} quantifies whether each class mean collapses onto the dominant one-dimensional subspace represented by $\mathbf{u}$, while ONC_{2-2} measures whether $\mathbf{w}$ also collapses onto the same subspace; since our experiments focus on the small-regularization regime under the use of symmetric g , ONC3 is expected to emerge in the form (21), and accordingly ONC_3 serves as an appropriate indicator for it.

5.2 Results

In the main text, the results for the ER and UTKFace datasets are shown; the logit model is used in both the cases, and ResNet101 among three backbones mentioned above is used for the UTKFace case. Experiments conducted under different settings also yielded consistent results and are reported in Appendix D. The results obtained using the clog-log model, as well as the quantitative comparison between the theoretical and experimental results—where the latter even exhibits phase transitions—are also reported in the same section.

Figures 2 and 3 plot the evolution of all evaluation metrics for ER and UTKFace, respectively.

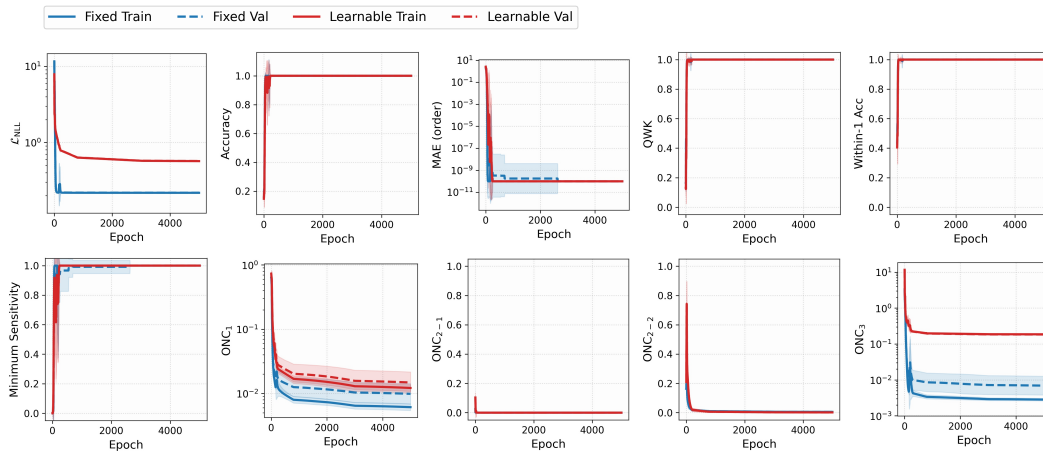


Figure 2: Epoch-wise average metrics curves for the ER dataset with the logit model.

For the ER dataset, both training and validation MAE approached zero, while accuracy, within-1 accuracy, QWK, and minimum sensitivity all approached one, indicating that all samples were correctly classified. ONC_{2-1} and ONC_{2-2} rapidly approached zero, showing that the feature vectors collapsed onto the one-dimensional subspace spanned by $\mathbf{w}$. As training proceeded, ONC_1 decreased steadily, confirming that features collapsed toward their class means. For ONC_3 we observed a clear

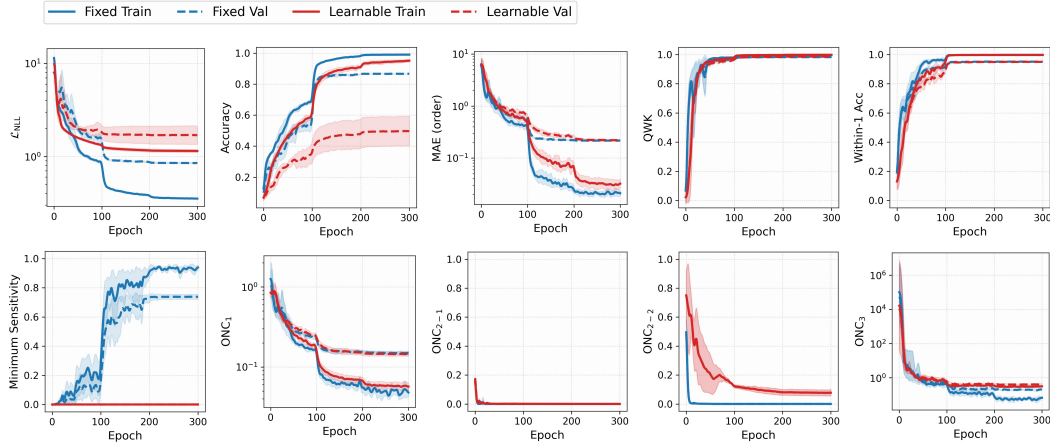


Figure 3: Epoch-wise average metrics curves for the UTKFace dataset with ResNet101 backbone.

difference between the two threshold strategies. With fixed thresholds, ONC_3 took a small value from an early stage and continued to decrease as training progressed. In contrast, with learnable thresholds, ONC_3 seemed to converge to a non-zero value. These observations indicate that the simple form of ONC_3 , given by (21), practically holds under fixed thresholds but does not hold under learnable ones.

For the more complex UTKFace dataset, the ONC metrics exhibit the same trends as observed on the ER dataset. Across metrics including $\mathcal{L}_{NLL}$, accuracy, MAE, and minimum sensitivity, fixed thresholds demonstrate better performance. Notably, both training and validation minimum sensitivity for learnable thresholds remain at zero throughout training, indicating that at least one class is completely ignored by the model. This phenomenon is consistently observed across the other two backbones presented in Appendix D.4.

To illustrate the ONC emergence, a visualization of the feature and latent space evolution throughout training are shown in Figs. 4 and 5. The red dashed lines denote the thresholds. Feature points are two-dimensional PCA-projected feature vectors and are color-coded by class, with validation features in lighter shades. Class means are highlighted with star markers, and the black arrow indicates the classifier weight. This visualization demonstrates the emergence of ONC_1 – 3 , but the ONC_3 behavior is clearly different between the two threshold strategies, as observed in the latent space.

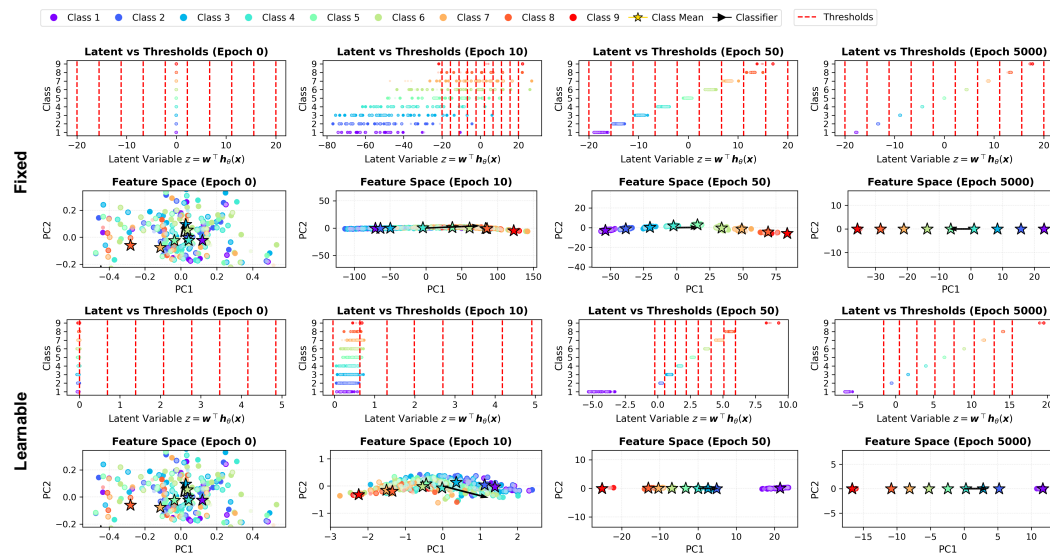


Figure 4: Latent and feature space visualization for the ER dataset with the logit model.

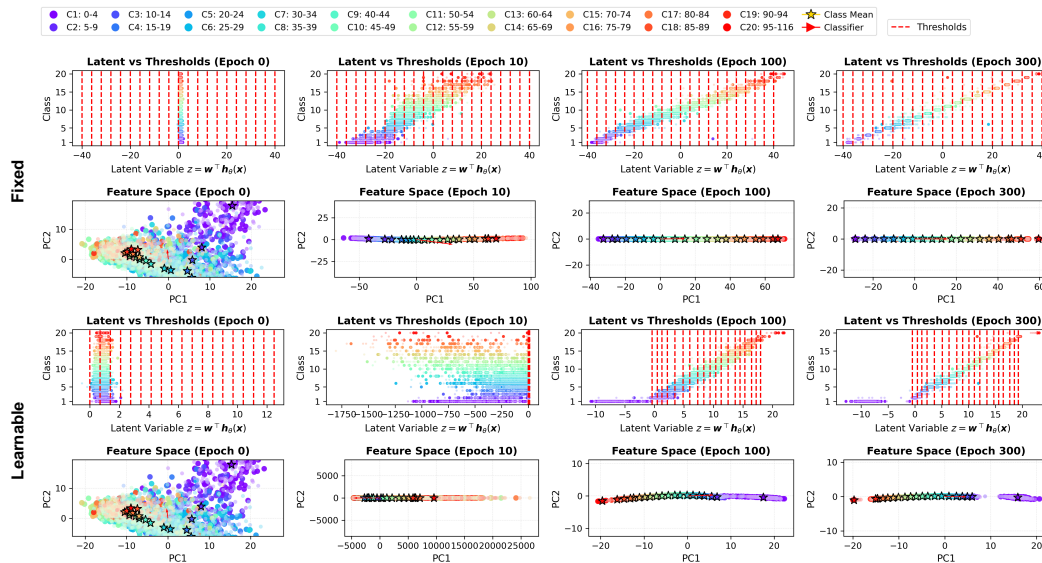


Figure 5: Latent and feature space visualization for the UTKFace dataset with ResNet101.

6 Discussion

Perspectives and future directions. ONC can also offer practical advantages. Although fixed thresholds were introduced primarily to establish ONC, as shown in Fig. 2, they tended to yield faster and more stable convergence compared with learnable thresholds. Moreover, as shown in Fig. 3, for the more complex UTKFace dataset, it can be seen that using fixed thresholds leads to a dramatic improvement in classification accuracy. Judging from the behavior of the minimum sensitivity, this improvement is largely attributed to better classification of the minority classes. This is actually natural since fixed thresholds provide a fairer allocation of the latent space—and hence of predicted probabilities—across all classes and they can offer greater robustness and generalization under label imbalance or label shift. We believe that this insight will be valuable also for practitioners.

Furthermore, the geometric structure induced by ONC can be directly utilized in the design of regularization terms or loss functions. For instance, adding lightweight penalties that attract each class mean toward the classifier axis or to the corresponding threshold midpoint may accelerate training, especially in scenarios with scarce labels or significant class imbalance. We leave such extensions as promising directions for future exploration.

Limitations. The theoretical development in Section 4 assumes that the thresholds $\mathbf{b}$ are fixed. Although our experimental results suggest that ONC1–2 also emerges even when $\mathbf{b}$ is learnable, this has not yet been theoretically established. Moreover, we believe that there exist only two phases—one with the trivial solution and the other with a non-trivial solution—but we have not succeeded in rigorously proving this. Instead, in Theorem 4.3, we circumvented this gap by assuming the continuity and monotonicity of w^* w.r.t. λ_w and λ_h .

During the revision, we realized that the current definition of ONC₃ (25) is only valid when the thresholds are fixed and evenly spaced in the latent space, which is the case for our fixed threshold experiments but is not for the others. Even though, our overall conclusion for the learnable threshold case remains unaffected because it is clear from other analyses including the latent space visualization that ONC3 behavior does not occur in the same manner as with fixed thresholds. Developing generally applicable ONC3 metrics is possible but the corresponding experiment is left for future work.

Conclusion. This study extended the NC theory to CLM-based OR through the UFM framework, to find ONC involving three hallmark properties. In the small-regularization limit, we also showed that a highly local and simple relation between the thresholds and the latent variables emerges. Experiments on real-world datasets confirmed the theoretical predictions and additionally demonstrated that fixed thresholds not only exhibit ONC but also lead to training speedup and better classification accuracy. These suggest that ONC can be an effective concept for generic OR tasks.

Acknowledgments and Disclosure of Funding

This work was partially supported by JST, CREST Grant number JPMJCF1862, Japan (TO, TT), JSPS KAKENHI under Grant 22K12179 (TO), Grant-in-Aid for Transformative Research Areas (A), “Foundation of machine learning physics” (22H05117) (TO), and Grant-in-Aid for Transformative Research Areas (A), “Shin-biology regulated by protein lifetime” (24H01895) (TT).

References

- Alan Agresti. *Analysis of Ordinal Categorical Data*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., 2nd edition, 2010. doi: 10.1002/9780470594001.
- George Andriopoulos, Zixuan Dong, Li Guo, Zifan Zhao, and Keith Ross. The prevalence of neural collapse in neural multivariate regression. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 126417–126451, 2024.
- Mark Bagnoli and Ted Bergstrom. Log-concave probability and its applications. *Economic Theory*, 26(2):445–469, August 2005.
- Arie Ben-David. Automatic generation of symbolic multi-attribute ordinal knowledge-based DSSs: Methodology and applications. *Decision Sciences*, 23(6):1357–1372, 1992. Data accessible at the OpenML repository <https://api.openml.org/d/1030> (ERA), <https://api.openml.org/d/1029> (LEV), and <https://api.openml.org/d/1028> (SWD).
- Ido Ben-Shaul, Ravid Shwartz-Ziv, Tomer Galanti, Shai Dekel, and Yann LeCun. Reverse engineering self-supervised learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 58324–58345, 2023.
- Marko Bohanec and Vladislav Rajkovič. Knowledge acquisition and explanation for multi-attribute decision making. In *Proceedings of the 8th International Workshop ‘Expert Systems and Their Applications’*, volume 1, pages 59–78, 1988. Data accessible at the UCI repository <https://archive.ics.uci.edu/dataset/19/car+evaluation>.
- Arne Brøndsted. *An Introduction to Convex Polytopes*, volume 90 of *Graduate Texts in Mathematics*. Springer, 1983.
- Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, November 2009. Data accessible at the UCI repository <https://archive.ics.uci.edu/dataset/186/wine+quality>.
- Hien Dang, Tho Tran Huu, Stanley Osher, Hung The Tran, Nhat Ho, and Tan Minh Nguyen. Neural collapse in deep linear networks: from balanced to imbalanced data. In *ICML’23: Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 6873–6947. PMLR, 2023.
- Hien Dang, Tho Tran, Tan Nguyen, and Nhat Ho. Neural collapse for cross-entropy class-imbalanced learning with unconstrained ReLU features model. In *ICML’24: Proceedings of the 41th International Conference on Machine Learning*, volume 235, pages 10017–10040. PMLR, 2024.
- Egbert Dierker. Competition for customers. In William A. Barnett, Bernard Cornet, Claude d’Aspremont, Jean Gabszewicz, and Andreu Mas-Colell, editors, *Equilibrium Theory and Applications: Proceedings of the Sixth International Symposium in Economic Theory and Econometrics*, pages 383–402. Cambridge University Press, 1991.
- Manuel Dorado-Moreno, Pedro Antonio Gutiérrez, and Cesar Martínez. Ordinal classification using hybrid artificial neural networks with projection and kernel basis functions. In Emilio Corchado, Václav Snášel, Ajith Abraham, Michael Woźniak, Manuel Graña, and Sung-Bae Cho, editors, *Hybrid Artificial Intelligent Systems, 7th International Conference*, volume 7209 of *Lecture Notes in Computer Science*, pages 319–330. Springer, 2012. ISBN 978-3-642-28930-9. doi: 10.1007/978-3-642-28931-6_31.

- Weinan E and Stephan Wojtowysch. On the emergence of simplex symmetry in the final and penultimate layers of neural network classifiers. In *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*, volume 145, pages 270–290. PMLR, 2020.
- Cong Fang, Hangfeng He, Qi Long, and Weijie J. Su. Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training. *Proceedings of the National Academy of Sciences*, 118(43):e2103091118, October 2021. doi: 10.1073/pnas.2103091118. URL <https://www.pnas.org/doi/abs/10.1073/pnas.2103091118>.
- Tomer Galanti, András György, and Marcus Hutter. On the role of neural collapse in transfer learning. In *International Conference on Learning Representations 2022*, 2022.
- P. A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernández-Navarro, and C. Hervás-Martínez. Ordinal regression methods: Survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):127–146, January 2016. doi: 10.1109/TKDE.2015.2457911.
- Hangfeng He and Weijie J Su. A law of data separation in deep learning. *Proceedings of the National Academy of Sciences*, 120(36):e2221704120, 2023.
- K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, December 2015.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.
- Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. In Alexander J. Smola, Peter Bartlett, Bernhard Schölkopf, and Dale Schuurmans, editors, *Advances in Large Margin Classifiers*, chapter 7. The MIT Press, 2000. doi: 10.7551/mitpress/1113.003.0010.
- Wanli Hong and Shuyang Ling. Neural collapse for unconstrained feature model under cross-entropy loss with imbalanced data. *Journal of Machine Learning Research*, 25(192):1–48, 2024. URL <http://jmlr.org/papers/v25/23-1215.html>.
- Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. doi: 10.1109/CVPR.2017.243.
- Like Hui, Mikhail Belkin, and Preetum Nakkiran. Limitations of neural collapse for understanding generalization in deep learning. *arXiv preprint arXiv:2202.08384v1 [cs.LG]*, 2022.
- László Leindler. On a certain converse of Hölder’s inequality. II. *Acta Scientiarum Mathematicarum*, 33:217–223, 1972.
- Pengyu Li, Xiao Li, Yutong Wang, and Qing Qu. Neural collapse in multi-label learning with pick-all-label loss. In *ICML’24: Proceedings of the 41th International Conference on Machine Learning*, volume 235, pages 28060–28094. PMLR, 2024a.
- Xiao Li, Sheng Liu, Jinxin Zhou, Xinyu Lu, Carlos Fernandez-Granda, Zhihui Zhu, and Qing Qu. Understanding and improving transfer learning of deep models via neural collapse. *Transactions on Machine Learning Research*, May 2024b.
- Jianfeng Lu and Stefan Steinerberger. Neural collapse under cross-entropy loss. *Applied and Computational Harmonic Analysis*, 59:224–241, July 2022. doi: 10.1016/j.acha.2021.12.011.
- Wojciech Masarczyk, Mateusz Ostaszewski, Ehsan Imani, Razvan Pascanu, Piotr Miłoś, and Tomasz Trzcinski. The tunnel effect: Building data representations in deep neural networks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 76772–76805, 2023.
- Dustin G. Mixon, Hans Parshall, and Jianzong Pi. Neural collapse with unconstrained features. *Sampling Theory, Signal Processing, and Data Analysis*, 20(2):11, 2022.

- Bac Nguyen, Chieh-Hsin Lai, Yuhta Takida, Naoki Murata, Toshimitsu Uesaka, Stefano Ermon, and Yuki Mitsufuji. Mitigating embedding collapse in diffusion models for categorical data. *arXiv preprint arXiv:2410.14758v1 [cs.LG]*, 2024.
- Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, September 2020. doi: 10.1073/pnas.2015509117. URL <https://www.pnas.org/doi/abs/10.1073/pnas.2015509117>.
- Liam Parker, Emre Onal, Anton Stengel, and Jake Intrater. Neural collapse in the intermediate hidden layers of classification neural networks. *arXiv preprint arXiv:2308.02760v1 [cs.LG]*, 2023.
- András Prékopa. On logarithmic concave measures and functions. *Acta Scientiarum Mathematicarum*, 34:335–343, 1973.
- Akshay Rangamani, Marius Lindegaard, Tomer Galanti, and Tomaso A. Poggio. Feature learning in deep classifiers through intermediate neural collapse. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28729–28745. PMLR, 2023.
- Barry Simon. *Convexity: An Analytic Viewpoint*, volume 187 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2011.
- Peter Sűkenűk, Marco Mondelli, and Christoph H Lampert. Deep neural collapse is provably optimal for the deep unconstrained features model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 52991–53024, 2023.
- Peter Sűkenűk, Christoph Lampert, and Marco Mondelli. Neural collapse versus low-rank bias: Is deep neural collapse really optimal? In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 138250–138288, 2024.
- Christos Thrampoulidis, Ganesh Ramachandra Kini, Vala Vakilian, and Tina Behnia. Imbalance trouble: Revisiting neural-collapse geometry. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27225–27238, 2022.
- Tom Tirer and Joan Bruna. Extended unconstrained features model for exploring deep neural collapse. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning*, volume 162, pages 21478–21505. PMLR, 2022.
- Victor Manuel Vargas, Pedro Antonio Gutiérrez, and César Hervás-Martínez. Cumulative link models for deep ordinal classification. *Neurocomputing*, 401:48–58, August 2020.
- Jan Verwaeren, Willem Waegeman, and Bernard De Baets. Learning partial ordinal class memberships with kernel-based proportional odds models. *Computational Statistics & Data Analysis*, 56:928–942, April 2012. doi: 10.1016/j.csda.2010.12.007.
- Robert Wu and Vardan Papyan. Linguistic collapse: Neural collapse in (large) language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 137432–137473, 2024.
- Zhifei Zhang, Yang Song, and Hairong Qi. Age Progression/Regression by Conditional Adversarial Autoencoder. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4352–4360, Los Alamitos, CA, USA, July 2017. IEEE Computer Society. doi: 10.1109/CVPR.2017.463. URL <https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.463>.

Jinxin Zhou, Chong You, Xiao Li, Kangning Liu, Sheng Liu, Qing Qu, and Zhihui Zhu. Are all losses created equal: A neural collapse perspective. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 31697–31710, 2022.

Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A geometric analysis of neural collapse with unconstrained features. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29820–29834, 2021.

A Log-concavity

A.1 Prékopa's theorem

We first recall a theorem, due to Prékopa [1973], which is derived from the Prékopa-Leindler inequality (Theorem A.2) on log-concave functions. It basically states that marginalization preserves the log-concavity. We use it to prove Theorem 4.1 in the main text.

Theorem A.1 (Theorem 6 of Prékopa [1973]). *Let $f(\mathbf{x}, \mathbf{y})$ be a function of $(n + m)$ variables where $\mathbf{x}$ and $\mathbf{y}$ are n - and m -dimensional, respectively. Suppose that f is log-concave on $\mathbb{R}^{n+m}$ and let A be a convex subset of $\mathbb{R}^m$. Then the function of $\mathbf{x}$ defined by*

$$\int_A f(\mathbf{x}, \mathbf{y}) d\mathbf{y} \quad (26)$$

is log-concave on $\mathbb{R}^n$.

A.2 Strict log-concavity

Here we show that the strict log-concavity of $g'(x)$ on $\mathbb{R}$ ensures the strict convexity of $L(z, a, b)$ in z . Although proving it would be easy if one can assume differentiability of g' , as demonstrated in Appendix B, it holds even without the differentiability assumption, as shown in the following.

We start by recalling the Prékopa-Leindler inequality [Prékopa, 1973, Leindler, 1972].

Theorem A.2 (Theorem 3 of Prékopa [1973]). *Suppose that $\lambda_i, i = 1, \dots, k$, are positive constants satisfying $\sum_{i=1}^k \lambda_i = 1$. Let $f_1, \dots, f_k$ be nonnegative and Borel measurable functions on $\mathbb{R}^n$, and let*

$$r(\mathbf{t}) = \sup_{\sum_{i=1}^k \lambda_i \mathbf{x}_i = \mathbf{t}} f_1(\mathbf{x}_1) \cdots f_k(\mathbf{x}_k), \quad \mathbf{t} \in \mathbb{R}^n. \quad (27)$$

Then the function $r(\mathbf{t})$ is Lebesgue measurable and we have

$$\int_{\mathbb{R}^n} r(\mathbf{t}) d\mathbf{t} \geq \prod_{i=1}^k \left(\int_{\mathbb{R}^n} f_i^{1/\lambda_i}(\mathbf{x}_i) d\mathbf{x}_i \right)^{\lambda_i}. \quad (28)$$

We need some more preparation.

Lemma 1. Let $f : D \rightarrow [0, \infty)$ be a strictly log-concave function on a convex set D . If there exists $\mathbf{x}_0 \in D$ for which $f(\mathbf{x}_0) = 0$, then $\mathbf{x}_0$ is an extreme point of D .

Proof. We prove that if $\mathbf{x}_0$ is not an extreme point of D (that is, $\mathbf{x}_0$ can be represented as a convex combination of $\mathbf{x}, \mathbf{y} \in D$ such that $\mathbf{x}_0 = \lambda \mathbf{x} + (1 - \lambda)\mathbf{y}$, $\exists \lambda \in (0, 1)$), then $f(\mathbf{x}_0)$ cannot be zero. From the strict log-concavity of f , one has

$$f(\mathbf{x}_0) = f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) > f(\mathbf{x})^\lambda f(\mathbf{y})^{1-\lambda} \geq 0, \quad (29)$$

showing that $f(\mathbf{x}_0)$ is positive. $\square$

It should be noted that Lemma 1 implies that any strictly log-concave function on $\mathbb{R}^n$ is strictly positive.

Let A be a subset of $\mathbb{R}^n$. For $\mathbf{a} \in \mathbb{R}^n$, let $A_{\mathbf{a}} = \{\mathbf{x} + \mathbf{a} \mid \mathbf{x} \in A\}$ and $A_{\setminus \mathbf{a}} = A \setminus A_{\mathbf{a}} = A \cap \overline{(A_{\mathbf{a}})^c}$.

Lemma 2. Let A be a convex subset of $\mathbb{R}^n$ and $\mathbf{a} \in \mathbb{R}^n$ be a non-zero vector. Then one has:

- (a) If $A_{\setminus \mathbf{a}} = \emptyset$, then one has $\{\mathbf{x} - t\mathbf{a} \mid \mathbf{x} \in A, t \in [0, \infty)\} \subset A$.
- (b) Consider the set $A_{\setminus t\mathbf{a}}$ indexed by $t \in \mathbb{R}$. If $A_{\setminus t_0\mathbf{a}} = \emptyset$ for some $t_0 > 0$, then $A_{\setminus t\mathbf{a}} = \emptyset$ holds for any $t \geq 0$.
- (c) $A_{\setminus t\mathbf{a}}$ is nondecreasing on $[0, \infty)$ (that is, for any $0 \leq t_1 < t_2$ one has $A_{\setminus t_1\mathbf{a}} \subset A_{\setminus t_2\mathbf{a}}$) and nonincreasing on $(-\infty, 0]$.

Proof. We first prove (a). $A_{\setminus \mathbf{a}} = \emptyset$ implies $A \subset A_{\mathbf{a}}$, that is, for any $\mathbf{x} \in A$ one has $\mathbf{x} - \mathbf{a} \in A$. By induction, for any $\mathbf{x} \in A$ one has $\mathbf{x} - m\mathbf{a} \in A$ for any $m \in \{0, 1, \dots\}$. Due to the convexity of A , one obtains $\mathbf{x} - t\mathbf{a} \in A$ for any $t \geq 0$, proving (a).

We next prove (b). From (a), the condition $A_{\setminus t_0\mathbf{a}} = \emptyset$ implies that for any $\mathbf{x} \in A$ one has $\mathbf{x} \in A_{t\mathbf{a}}$ for any $t \geq 0$, which in turn yields $A_{\setminus t\mathbf{a}} = \emptyset$ for any $t \geq 0$.

We now prove (c). Fix $0 \leq t_1 < t_2$. Take any point $\mathbf{x} \in A_{\setminus t_1\mathbf{a}}$, which implies that $\mathbf{x} \in A$ and $\mathbf{x} - t_1\mathbf{a} \notin A$. We show that $\mathbf{x} - t_2\mathbf{a} \notin A$. For this purpose, assume, to the contrary, that $\mathbf{x} - t_2\mathbf{a} \in A$ holds. One has

$$\mathbf{x} - t_1\mathbf{a} = \left(1 - \frac{t_1}{t_2}\right)(\mathbf{x} - t_2\mathbf{a}) + \frac{t_1}{t_2}\mathbf{x}, \quad (30)$$

which shows that $\mathbf{x} - t_1\mathbf{a}$ is a convex combination of $\mathbf{x}$ and $\mathbf{x} - t_2\mathbf{a}$, both lying in A . Due to the convexity of A , it would follow that $\mathbf{x} - t_1\mathbf{a} \in A$, which is a contradiction. We have thus proved that for any $\mathbf{x} \in A_{\setminus t_1\mathbf{a}}$, one has $\mathbf{x} - t_2\mathbf{a} \notin A$, and hence $\mathbf{x} \in A_{\setminus t_2\mathbf{a}}$. It in turn implies $A_{\setminus t_1\mathbf{a}} \subset A_{\setminus t_2\mathbf{a}}$, proving that $A_{\setminus t\mathbf{a}}$ is nondecreasing on $[0, \infty)$. The statement that $A_{\setminus t\mathbf{a}}$ is nonincreasing on $(-\infty, 0]$ can be proved in the same manner. $\square$

For a subset A of $\mathbb{R}^n$, let $\mu(A)$ denote its volume.

Lemma 3. Let A be a convex subset of $\mathbb{R}^n$, and $\mathbf{a} \in \mathbb{R}^n$ be a non-zero vector.

- (a) If $\mu(A_{\setminus t_0\mathbf{a}}) = 0$ for some $t_0 > 0$, then one has $\mu(A_{\setminus t\mathbf{a}}) = 0$ for all $t \geq 0$.
- (b) If $\mu(A_{\setminus t_0\mathbf{a}}) > 0$ for some $t_0 > 0$, then one has $\mu(A_{\setminus t\mathbf{a}}) > 0$ for all $t > 0$.

Proof. We first prove (a). The condition $\mu(A_{\setminus t_0\mathbf{a}}) = 0$ implies that for almost every $\mathbf{x} \in A$ one should have $\mathbf{x} \in A_{t_0\mathbf{a}}$, or equivalently, $\mathbf{x} - t_0\mathbf{a} \in A$. One then has $\mathbf{x} - 2t_0\mathbf{a} \in A$ for almost every such $\mathbf{x} \in A$ satisfying $\mathbf{x} - t_0\mathbf{a} \in A$. By induction, for almost every $\mathbf{x} \in A$ one has $\mathbf{x} - mt_0\mathbf{a} \in A$ for any $m \in \{1, 2, \dots\}$. Due to the convexity of A , for almost every $\mathbf{x} \in A$ one has $\{\mathbf{x} - t\mathbf{a} \mid t \geq 0\} \subset A$. It then implies that, for any $t \geq 0$ and for almost every $\mathbf{x} \in A$, one has $\mathbf{x} \in A_{t\mathbf{a}}$, showing that $\mu(A_{\setminus t\mathbf{a}}) = 0$ holds for all $t \geq 0$.

Proving (b) is straightforward by noting that if one has $\mu(A_{\setminus t_0\mathbf{a}}) > 0$ and $\mu(A_{\setminus t_1\mathbf{a}}) = 0$ for $t_0, t_1 > 0$, $t_0 \neq t_1$, the latter would imply, via (a), $\mu(A_{\setminus t\mathbf{a}}) = 0$ for all $t \geq 0$, which is in conflict with the former. $\square$

We need the following assumption.

Assumption 1. The subset A of $\mathbb{R}^n$ is convex. Furthermore, for any non-zero vector $\mathbf{a} \in \mathbb{R}^n$, either $\mu(A_{\setminus \mathbf{a}})$ or $\mu(A_{\setminus -\mathbf{a}})$ is positive.

The last positivity assumption in Assumption 1 does not always hold (consider the case where A is a half-space and $\mathbf{a}$ is parallel to the boundary of A). Lemma 4 below gives a necessary and sufficient condition for that.

Lemma 4. Let A be a convex subset of $\mathbb{R}^n$ with $\mu(A) > 0$. Assume that A has n supporting hyperplanes $H_1, \dots, H_n$ with their normal vectors $\mathbf{n}_1, \dots, \mathbf{n}_n$ forming a basis of $\mathbb{R}^n$. Then for any non-zero vector $\mathbf{a} \in \mathbb{R}^n$, either $\mu(A_{\setminus \mathbf{a}})$ or $\mu(A_{\setminus -\mathbf{a}})$ is positive. Conversely, if the normal vectors of the supporting hyperplanes of A do not span $\mathbb{R}^n$, then there exists $\mathbf{a} \neq \mathbf{0}$ with which $\mu(A_{\setminus \mathbf{a}}) = \mu(A_{\setminus -\mathbf{a}}) = 0$ holds.

Proof. We first prove the former statement. As $\{\mathbf{n}_1, \dots, \mathbf{n}_n\}$ is a basis of $\mathbb{R}^n$, not all the inner products $\{\langle \mathbf{n}_i, \mathbf{a} \rangle\}_{i=1, \dots, n}$ are simultaneously zero for any non-zero vector $\mathbf{a}$. Assume without loss of generality that $\langle \mathbf{n}_1, \mathbf{a} \rangle \neq 0$. Let $\mathbf{p} \in H_1$ be a boundary point of A . Then one has $H_1 = \{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{n}_1, \mathbf{x} - \mathbf{p} \rangle = 0\}$. Take $-\mathbf{n}_1$ in place of $\mathbf{n}_1$ if necessary, one can assume that $\{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{n}_1, \mathbf{x} - \mathbf{p} \rangle \geq 0\} \supset A$ holds.

Take a point $\mathbf{z}$ in the interior of A , and a ball $B_r(\mathbf{z})$ of small enough radius $r > 0$ centered at $\mathbf{z}$, so that $B_r(\mathbf{z}) \subset A$ holds. Consider the hyperplane $H = \{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{n}_1, \mathbf{x} - \mathbf{z} \rangle = 0\}$. It is parallel to H_1 and passes through $\mathbf{z}$. It cuts the set A into two parts, each of which contains the half of the ball $B_r(\mathbf{z})$ and is thus of positive volume. Let $A' = \{\mathbf{x} \in A \mid \langle \mathbf{n}_1, \mathbf{x} - \mathbf{z} \rangle < 0\}$, that is, A' is the set of

points in A which resides on the same side of H as the point p . One consequently has $\mu(A') > 0$ since A' contains the half-ball.

Let

$$p' = p + ta, \quad t = \frac{\langle n_1, z - p \rangle}{\langle n_1, a \rangle}. \quad (31)$$

The point p' is on H because

$$\langle n_1, p' - z \rangle = \langle n_1, p - z \rangle + \frac{\langle n_1, z - p \rangle}{\langle n_1, a \rangle} \langle n_1, a \rangle = 0 \quad (32)$$

holds. Furthermore, H is a supporting hyperplane of A_{ta} : noting that $p' + p + ta \in A_{ta} = \{x + ta \mid x \in A\}$, one has

$$\begin{aligned} \langle n_1, x + ta - z \rangle &= \langle n_1, x - p \rangle + \langle n_1, p + ta - z \rangle \\ &= \langle n_1, x - p \rangle + \langle n_1, p' - z \rangle \geq 0, \end{aligned} \quad (33)$$

where the last inequality is due to $A \subset \{x \mid \langle n_1, x - p \rangle \geq 0\}$ and (32). It implies that $A_{ta} \subset \{x \mid \langle n_1, x - z \rangle \geq 0\}$ holds. It should be noted that the point p lies on the opposite side of H to A_{ta} , which can be confirmed by noting that $z \in A \subset \{x \mid \langle n_1, x - p \rangle \geq 0\}$ implies $\langle n_1, p - z \rangle < 0$. One consequently has $A_{ta} \cap A' = \emptyset$, which furthermore implies $A' \subset \overline{(A_{ta})}$.

One therefore has $A_{\setminus ta} = A \cap \overline{(A_{ta})} \supset A'$, and thus $\mu(A_{\setminus ta}) \geq \mu(A') > 0$. One can then apply Lemma 3 to show that either $\mu(A_{\setminus a})$ or $\mu(A_{\setminus -a})$ is positive.

We next prove the converse. Let W be the linear span of the normals of the supporting hyperplanes of A , and let $\pi_W : \mathbb{R}^n \rightarrow W$ be the orthogonal projection onto W . The supporting hyperspace of A with the normal $n \in W$ can be represented as $c_n \geq \langle n, x \rangle = \langle n, \pi_W(x) \rangle$. Since the closure A° of A is convex and thus is the intersection of its supporting hyperspaces [Brøndsted, 1983, Theorem 4.5], one has

$$\begin{aligned} A^\circ &= \bigcap_{n \in W} \{x \mid \langle n, x \rangle \leq c_n\} \\ &= \bigcap_{n \in W} \{x \mid \langle n, \pi_W(x) \rangle \leq c_n\} \\ &= \bigcap_{n \in W} (\{y \in W \mid \langle n, y \rangle \leq c_n\} + W^\perp) \\ &= \left(\bigcap_{n \in W} \{y \in W \mid \langle n, y \rangle \leq c_n\} \right) + W^\perp, \end{aligned} \quad (34)$$

showing that A° is a cylinder since $W^\perp \neq \{0\}$. It immediately implies that for any $a \in W^\perp$ one has $(A^\circ)_{\setminus a} = \emptyset$, and consequently, $\mu(A_{\setminus a}) = 0$. $\square$

The positivity assumption in Assumption 1 is automatically satisfied if A is convex and $\mu(A) \in (0, \infty)$, as shown in the next lemma.

Lemma 5. Let A be a convex subset of $\mathbb{R}^n$, and $a \in \mathbb{R}^n$ be a non-zero vector. Assume $\mu(A) \in (0, \infty)$. Then one has $\mu(A_{\setminus a}) > 0$.

Proof. The condition $\mu(A) \in (0, \infty)$ implies that A is bounded and that its closure A° is compact. For any nonzero $n \in \mathbb{R}^n$, the function $x \mapsto \langle n, x \rangle$ is continuous, and it attains its maximum c_n on A° . Then $\langle n, x \rangle = c_n$ is the supporting hyperplane of A with normal n . One can then observe that, by taking $n_1 = a = n$ in the proof of Lemma 4, one has $\mu(A_{\setminus n}) > 0$. $\square$

We now state the main theorem.

Theorem A.3. Assume that $f(x)$ is strictly log-concave on $\mathbb{R}^n$. For a subset A of $\mathbb{R}^n$ satisfying Assumption 1, let $\rho(x)$ be defined by

$$\rho(x) = \int_A f(u - x) du. \quad (35)$$

Then $\rho(x)$ is also strictly log-concave.

Proof. Take arbitrary $\mathbf{x}_0, \mathbf{x}_1 \in \mathbb{R}^n$ with $\mathbf{x}_1 - \mathbf{x}_0 = \boldsymbol{\delta} \neq \mathbf{0}$ and let $\mathbf{x}_\lambda = (1 - \lambda)\mathbf{x}_0 + \lambda\mathbf{x}_1$. We will show the strict inequality $\rho(\mathbf{x}_\lambda) > \rho(\mathbf{x}_0)^{1-\lambda}\rho(\mathbf{x}_1)^\lambda$ to hold for any $\lambda \in (0, 1)$, which proves the theorem. Let

$$F_\lambda(\mathbf{u}) = f(\mathbf{u} - \mathbf{x}_\lambda) \mathbb{1}(\mathbf{u} \in A). \quad (36)$$

One then has

$$\rho(\mathbf{x}_\lambda) = \int_A f(\mathbf{u} - \mathbf{x}_\lambda) d\mathbf{u} = \int_{\mathbb{R}^n} F_\lambda(\mathbf{u}) d\mathbf{u}. \quad (37)$$

$F_\lambda(\mathbf{u})$ is log-concave in $\mathbf{u}$ because it is a product of the two log-concave functions $f(\mathbf{u} - \mathbf{x}_\lambda)$ and $\mathbb{1}(\mathbf{u} \in A)$ (note that A is convex due to Assumption 1), and because multiplication of log-concave functions preserves log-concavity.

Lemma 6. For any $\lambda \in (0, 1)$ and any $\mathbf{u}_0, \mathbf{u}_1 \in \mathbb{R}^n$, let $\mathbf{u}_\lambda = (1 - \lambda)\mathbf{u}_0 + \lambda\mathbf{u}_1$. One then has

$$F_\lambda(\mathbf{u}_\lambda) \geq F_0(\mathbf{u}_0)^{1-\lambda} F_1(\mathbf{u}_1)^\lambda, \quad (38)$$

with strict inequality when $F_\lambda(\mathbf{u}_\lambda) > 0$ and $\mathbf{u}_1 - \mathbf{u}_0 \neq \boldsymbol{\delta}$.

Proof of Lemma 6. Fix $\lambda \in (0, 1)$. Assume that $\mathbf{u}_0, \mathbf{u}_1$ are such that $\mathbf{u}_\lambda \in A$ and $\mathbf{u}_1 - \mathbf{u}_0 \neq \boldsymbol{\delta}$ hold. One then has $F_\lambda(\mathbf{u}_\lambda) = f(\mathbf{u}_\lambda - \mathbf{x}_\lambda) > 0$, where the positivity of f is due to Lemma 1. For such $\mathbf{u}_0, \mathbf{u}_1$, one has

$$\begin{aligned} F_\lambda(\mathbf{u}_\lambda) &= f(\mathbf{u}_\lambda - \mathbf{x}_\lambda) = f((1 - \lambda)(\mathbf{u}_0 - \mathbf{x}_0) + \lambda(\mathbf{u}_1 - \mathbf{x}_1)) \\ &> f(\mathbf{u}_0 - \mathbf{x}_0)^{1-\lambda} f(\mathbf{u}_1 - \mathbf{x}_1)^\lambda \\ &\geq F_0(\mathbf{u}_0)^{1-\lambda} F_1(\mathbf{u}_1)^\lambda, \end{aligned} \quad (39)$$

where the first inequality is due to the strict log-concavity of f and the condition $\mathbf{u}_0 - \mathbf{x}_0 \neq \mathbf{u}_1 - \mathbf{x}_1$ which is derived from $\mathbf{u}_1 - \mathbf{u}_0 \neq \boldsymbol{\delta} = \mathbf{x}_1 - \mathbf{x}_0$, and where the second inequality results from multiplication with the indicator functions.

For $\mathbf{u}_0, \mathbf{u}_1$ such that $\mathbf{u}_\lambda \in A$ and $\mathbf{u}_1 - \mathbf{u}_0 = \boldsymbol{\delta}$ hold, one has $\mathbf{u}_1 - \mathbf{x}_1 = \mathbf{u}_0 - \mathbf{x}_0$ and $\mathbf{u}_\lambda - \mathbf{x}_\lambda = (1 - \lambda)(\mathbf{u}_1 - \mathbf{x}_1) + \lambda(\mathbf{u}_0 - \mathbf{x}_0) = \mathbf{u}_0 - \mathbf{x}_0$, so that $f(\mathbf{u}_\lambda - \mathbf{x}_\lambda)$ is constant for $\lambda \in [0, 1]$, and hence the inequality (38) holds. For $\mathbf{u}_0, \mathbf{u}_1$ such that $\mathbf{u}_\lambda \notin A$ holds, either $\mathbf{u}_0$ or $\mathbf{u}_1$ should lie outside A , so that one has the equality $F_\lambda(\mathbf{u}_\lambda) = F_0(\mathbf{u}_0)^{1-\lambda} F_1(\mathbf{u}_1)^\lambda = 0$. $\square$

We return to the proof of Theorem A.3. Let

$$r(\mathbf{u}) = \sup_{\mathbf{u}_0, \mathbf{u}_1: (1-\lambda)\mathbf{u}_0 + \lambda\mathbf{u}_1 = \mathbf{u}} F_0(\mathbf{u}_0)^{1-\lambda} F_1(\mathbf{u}_1)^\lambda. \quad (40)$$

Then for $\mathbf{u} \notin A$ one has $F_\lambda(\mathbf{u}) = r(\mathbf{u}) = 0$. On the other hand, for $\mathbf{u} \in A_{\setminus \lambda \boldsymbol{\delta}}$, one cannot have $\mathbf{u}_0, \mathbf{u}_1 \in A$ which satisfy both $\mathbf{u}_1 - \mathbf{u}_0 = \boldsymbol{\delta}$ and $(1 - \lambda)\mathbf{u}_0 + \lambda\mathbf{u}_1 = \mathbf{u}$: indeed, since $\mathbf{u}_0 \in A$, if one lets $\mathbf{u}_1 = \mathbf{u}_0 + \boldsymbol{\delta}$, one has $\mathbf{u} = (1 - \lambda)\mathbf{u}_0 + \lambda\mathbf{u}_1 = \mathbf{u}_0 + \lambda\boldsymbol{\delta} \in A_{\lambda \boldsymbol{\delta}}$, which should not lie in $A_{\setminus \lambda \boldsymbol{\delta}}$. This, together with Lemma 6, in turn implies the strict inequality $F_\lambda(\mathbf{u}) > r(\mathbf{u})$ for $\mathbf{u} \in A_{\setminus \lambda \boldsymbol{\delta}}$. One can similarly show the strict inequality $F_\lambda(\mathbf{u}) > r(\mathbf{u})$ to hold for $\mathbf{u} \in A_{\setminus -(1-\lambda)\boldsymbol{\delta}}$ as well.

As we have shown the strict inequality $F_\lambda(\mathbf{u}) > r(\mathbf{u})$ to hold for $\mathbf{u}$ in the set $A_{\setminus \lambda \boldsymbol{\delta}} \cup A_{\setminus -(1-\lambda)\boldsymbol{\delta}}$ which has a positive volume for $\boldsymbol{\delta} \neq \mathbf{0}$ due to Assumption 1, one has

$$\begin{aligned} \rho(\mathbf{x}_\lambda) &= \int_{\mathbb{R}^n} F_\lambda(\mathbf{u}) d\mathbf{u} \\ &> \int_{\mathbb{R}^n} r(\mathbf{u}) d\mathbf{u} \\ &\geq \left(\int_{\mathbb{R}^n} F_0(\mathbf{u}) d\mathbf{u} \right)^{1-\lambda} \left(\int_{\mathbb{R}^n} F_1(\mathbf{u}) d\mathbf{u} \right)^\lambda \\ &= \rho(\mathbf{x}_0)^{1-\lambda} \rho(\mathbf{x}_1)^\lambda, \end{aligned} \quad (41)$$

where the first inequality is due to the strict inequality shown above, and where the second inequality is derived by applying the Prékopa-Leindler inequality (Theorem A.2). This proves the strict log-concavity of ρ . $\square$

Strict log-concavity of the function

$$g(b-z) - g(a-z) = \int_a^b g'(u-z) du \quad (42)$$

in z under the assumption of strict log-concavity of g' is immediate from Theorem A.3. It proves the strict convexity of $L(z, a, b) = -\log[g(b-z) - g(a-z)]$ in z as well.

It should be noted that the function $f(\mathbf{u} - \mathbf{x}) \mathbb{1}(\mathbf{u} \in A)$, whose integral with respect to $\mathbf{u} \in \mathbb{R}^n$ yields $\rho(\mathbf{x})$, is itself log-concave but not strictly log-concave in $(\mathbf{x}, \mathbf{u})$, so that one cannot apply the argument in Prékopa [1973] on the strict log-concavity to our case.

B Derivation of ONC3

We show in this section the ordering $z_1^* \leq z_2^* \leq \dots \leq z_Q^*$ of the optimal latent variables. For this purpose, we discuss how the minimizer of the optimization problem

$$\min_x \left(L(x, a, b) + \frac{\lambda}{2} x^2 \right), \quad \lambda > 0, \quad a < b, \quad (43)$$

with the function $L(x, a, b)$ being of the form

$$L(x, a, b) = -\log[P(b-x) - P(a-x)], \quad (44)$$

behaves as one changes a, b , where P is an indefinite integral of a function p , which is log-concave on $\mathbb{R}$. It is because the optimal latent variable $z_q^* = w \mathbf{a}^\top \mathbf{h}_q^*$ given $w > 0$ is determined as the optimal solution of the following minimization:

$$\min_x \left(L(x, b_{q-1}, b_q) + \frac{\lambda_h}{2w^2} x^2 \right), \quad (45)$$

with $L(x, a, b)$ defined as in (44) using $P(z) = \int_{-\infty}^z g(u) du$.

Theorem 4.1 ensures that $L(x, a, b)$ is convex in x for any a, b , which in turn ensures that the minimizer $\hat{x}$ of (43) is unique. We show in the following that the minimizer $\hat{x} = \hat{x}(a, b)$ is monotonically non-decreasing in a and b . This monotonicity will prove the desired ordering of $\{z_q^*\}_q$.

The minimizer $\hat{x}$ satisfies the stationarity condition

$$L_x(\hat{x}, a, b) + \lambda \hat{x} = 0, \quad (46)$$

where the subscript x of L denotes the partial derivative of L with respect to x . It should be noted that particularizing (46) in the optimization of the latent variable z_q yields EOS (16a).

We first assume differentiability of p . Taking the derivative of both sides of (46) w.r.t. a , one has

$$L_{xx}(\hat{x}, a, b) \hat{x}_a + L_{xa}(\hat{x}, a, b) + \lambda \hat{x}_a = 0, \quad (47)$$

yielding

$$\hat{x}_a = -\frac{L_{xa}(\hat{x}, a, b)}{L_{xx}(\hat{x}, a, b) + \lambda}. \quad (48)$$

Similarly, one has

$$\hat{x}_b = -\frac{L_{xb}(\hat{x}, a, b)}{L_{xx}(\hat{x}, a, b) + \lambda}. \quad (49)$$

Since $L(x, a, b)$ is convex in x , one has $L_{xx}(x, a, b) \geq 0$. One also has

$$\begin{aligned} L_x(x, a, b) &= \frac{p(b-x) - p(a-x)}{P(b-x) - P(a-x)}, \\ L_{xa}(x, a, b) &= -\frac{p'(a-x)}{P(b-x) - P(a-x)} + \frac{[p(b-x) - p(a-x)]p(a-x)}{[P(b-x) - P(a-x)]^2} \\ &= -\frac{p(a-x)}{P(b-x) - P(a-x)} \left[\frac{p'(a-x)}{p(a-x)} - \frac{p(b-x) - p(a-x)}{P(b-x) - P(a-x)} \right], \end{aligned} \quad (50)$$

$$\begin{aligned} L_{xb}(x, a, b) &= \frac{p'(b-x)}{P(b-x) - P(a-x)} - \frac{[p(b-x) - p(a-x)]p(b-x)}{[P(b-x) - P(a-x)]^2} \\ &= \frac{p(b-x)}{P(b-x) - P(a-x)} \left[\frac{p'(b-x)}{p(b-x)} - \frac{p(b-x) - p(a-x)}{P(b-x) - P(a-x)} \right], \end{aligned} \quad (51)$$

$$\begin{aligned} L_{xx}(x, a, b) &= -\frac{p'(b-x) - p'(a-x)}{P(b-x) - P(a-x)} + \frac{[p(b-x) - p(a-x)]^2}{[P(b-x) - P(a-x)]^2} \\ &= -[L_{xa}(x, a, b) + L_{xb}(x, a, b)]. \end{aligned} \quad (52)$$

Since $p(u)$ is assumed log-concave, $(\log p(u))' = p'(u)/p(u)$ is monotonically non-increasing. One therefore has, via the technique used in [Dierker, 1991, Proof of Lemma 1] and Bagnoli and Bergstrom [2005],

$$\begin{aligned} \frac{p'(a-x)}{p(a-x)} [P(b-x) - P(a-x)] &= \frac{p'(a-x)}{p(a-x)} \int_{a-x}^{b-x} p(u) du \\ &\geq \int_{a-x}^{b-x} \frac{p'(u)}{p(u)} p(u) du \\ &= \int_{a-x}^{b-x} p'(u) du \\ &= p(b-x) - p(a-x), \end{aligned} \quad (53)$$

which, together with (50), implies that $L_{xa}(x, a, b) \leq 0$ holds. Combined with $L_{xx}(x, a, b) \geq 0$ and (48), it in turn proves $\hat{x}_a \geq 0$. Similarly, one has

$$\begin{aligned} \frac{p'(b-x)}{p(b-x)} [P(b-x) - P(a-x)] &= \frac{p'(b-x)}{p(b-x)} \int_{a-x}^{b-x} p(u) du \\ &\leq \int_{a-x}^{b-x} \frac{p'(u)}{p(u)} p(u) du \\ &= \int_{a-x}^{b-x} p'(u) du \\ &= p(b-x) - p(a-x), \end{aligned} \quad (54)$$

which, together with (51), implies that $L_{xb}(x, a, b) \leq 0$ holds. Combined with (49), it in turn proves $\hat{x}_b \geq 0$.

We next discuss the case where p is not necessarily differentiable. Since $p(u)$ is log-concave, $q(u) = -\log p(u)$ is convex, so that it is continuous and differentiable except on a countable set. Let $\phi(u)$ be any function such that $\phi(u)$ takes a value in the subderivative of $q(u)$ for any u . As $p(u) = e^{-q(u)}$, one has, at any point u at which $p(u)$ is differentiable,

$$p'(u) = -q'(u)e^{-q(u)} = -\phi(u)p(u). \quad (55)$$

One therefore has, for any $b > a$,

$$p(b) - p(a) = - \int_a^b \phi(u)p(u) du, \quad (56)$$

which can be proved in the same way as [Simon, 2011, Theorem 1.28]. On the other hand, we know that $\phi(u)$ is monotonically non-decreasing in u . One thus has

$$\phi(a)[P(b) - P(a)] = \phi(a) \int_a^b p(u) du \leq \int_a^b \phi(u)p(u) du \leq \phi(b) \int_a^b p(u) du = \phi(b)[P(b) - P(a)], \quad (57)$$

which, via replacing (a, b) with $(a - x, b - x)$, proves inequalities corresponding to (53) and (54).

One has therefore proven the following proposition.

Proposition B.1. For $a < b$, let $L(x, a, b)$ be as defined in (44), and let $\hat{x}(a, b)$ be the minimizer of the optimization problem (43). Then, for $a < b$ and $a' < b'$ with $a \leq a'$ and $b \leq b'$, one has $\hat{x}(a, b) \leq \hat{x}(a', b')$.

We would like to note that if p is differentiable and *strictly* log-concave, then we have strict inequalities in the above proposition, which in turn implies, via (52), that $L_{xx}(x, a, b) \gtrless 0$. This constitutes an alternative proof of the strict convexity of $L(x, a, b)$ in x under the differentiability assumption. Even without the differentiability assumption, one can note that, under the strict log-concavity of p , $\phi(u)$ is increasing, so that the strict inequalities hold in (57), proving the strict inequalities in the above proposition as well.

The above argument proves the first half of ONC3, that is, for any fixed w one has $z_q^* \leq z_{q+1}^*$ for $q \in \{1, \dots, Q - 1\}$. Furthermore, if g' is strictly log-concave, one has the strict ordering when $w \neq 0$: $z_q^* < z_{q+1}^*$ for $q \in \{1, \dots, Q - 1\}$, thereby proving the latter half of ONC3.

C Details of the Experimental Setup

Dataset statistics. We used the five publicly available real-world OR tabular datasets of Gutiérrez et al. [2016]¹—ER, LE, and SW (Employee rejection/acceptance, Lecturers evaluation, and Social workers decisions (public domain) [Ben-David, 1992]), CA (Car evaluation (CC BY 4.0) [Bohanec and Rajkovič, 1988]), and WR (Wine quality—Red (CC BY 4.0) [Cortez et al., 2009])—exactly as released. The website offers 30 pre-defined training–validation hold-out splits whose label distributions are identical across the two partitions. We ran our experiments on all the 30 splits for each dataset and report the averages.

Table 1 summarizes the key statistics of the five datasets used in our study. For consistency, all ordinal labels were remapped to consecutive integers starting from one. We constructed input vectors by concatenating two types of preprocessed attributes: one-hot encoded categorical attributes and normalized numerical attributes.

Table 1: Summary of tabular datasets used in the experiments. Attr. denotes attributes and Input dim. denotes the dimension of input vectors.

Dataset (code)	Subset	#Samples (per split)	#Attr.	Input dim.	#Classes (Q)	Distribution (counts per label)
ERA (ER)	Train	750	4	13	9	[1:69, 2:106, 3:136, 4:129, 5:118, 6:89, 7:66, 8:23, 9:14]
	Val	250				[1:23, 2:36, 3:45, 4:43, 5:40, 6:29, 7:22, 8:8, 9:4]
LEV (LE)	Train	750	4	9	5	[0:70, 1:210, 2:302, 3:148, 4:20]
	Val	250				[0:23, 1:70, 2:101, 3:49, 4:7]
SWD (SW)	Train	750	10	14	4	[2:24, 3:264, 4:299, 5:163]
	Val	250				[2:8, 3:88, 4:100, 5:54]
Car (CA)	Train	1296	6	21	4	[acc:288, good:52, unacc:907, vgood:49]
	Val	432				[acc:96, good:17, unacc:303, vgood:16]
Wine (WR)	Train	1199	11	17	6	[3:8, 4:39, 5:510, 6:479, 7:150, 8:13]
	Val	400				[3:2, 4:14, 5:171, 6:159, 7:49, 8:5]

Additionally, we conducted experiments on the UTKFace dataset (non-commercial research use) [Zhang et al., 2017], a large-scale face image dataset for age estimation. The dataset contains 23,708 facial images with age labels. We grouped ages into 20 ordinal classes using 5-year intervals. The dataset exhibits natural class imbalance, with the most frequent class (25–29 years) containing 5,034 samples and the rarest class (95–116 years) containing only 67 samples. Following standard practice, we performed a stratified 80/20 train-validation split to maintain class distribution across subsets. All images were resized to 224×224 pixels and normalized using ImageNet statistics for pre-trained backbone compatibility. Table 2 presents the detailed class distribution.

¹<https://www.uco.es/grupos/ayrna/orreview>.

Table 2: Class distribution of UTKFace dataset with 5-year age grouping.

Class	Age	Train	Val	Class	Age	Train	Val
0	[0–4]	1,733	434	10	[50–54]	1,076	269
1	[5–9]	716	179	11	[55–59]	763	191
2	[10–14]	471	118	12	[60–64]	586	146
3	[15–19]	753	189	13	[65–69]	469	117
4	[20–24]	1,848	462	14	[70–74]	298	75
5	[25–29]	4,027	1,007	15	[75–79]	261	65
6	[30–34]	1,832	458	16	[80–84]	190	47
7	[35–39]	1,797	450	17	[85–89]	214	53
8	[40–44]	945	236	18	[90–94]	82	20
9	[45–49]	851	213	19	[95–116]	54	13
Total:				18,966		4,742	

Overparameterized network. Here we describe the neural network architecture used in our experiments. For the tabular datasets, we employed a multilayer perceptron with residual connections as follows:

- An input $x \in \mathbb{R}^d$ is first mapped to a 128-dimensional representation by a linear layer, then passed through a parametric rectified linear unit (PReLU) [He et al., 2015].
- It then passes through four residual blocks, each defined as

$$x \mapsto x + \text{PReLU}(W_2 \text{PReLU}(W_1 x + b_1) + b_2), \quad W_1, W_2 \in \mathbb{R}^{128 \times 128}, \quad b_1, b_2 \in \mathbb{R}^{128}.$$

- It subsequently passes through three consecutive linear layers with linear activation, yielding the 64-dimensional feature $h_\theta(x)$.
- Finally, a linear layer without bias—whose weight vector is the classifier weight analyzed in this study—maps the $h_\theta(x)$ to the one-dimensional latent variable $z = w^\top h_\theta(x)$.

The PReLU activations together with the linear tail give the network enough flexibility to map inputs to any location in the feature space, aligning with the UFM assumption. Figure 6 illustrates the architecture of the network.

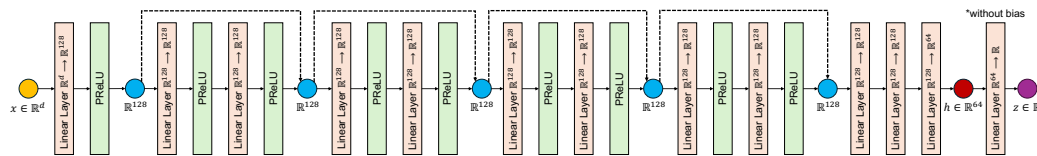


Figure 6: Architecture of the overparameterized network for tabular datasets.

For the UTKFace dataset, we employed pre-trained ResNet50, ResNet101 [He et al., 2016], and DenseNet201 [Huang et al., 2017] as backbones. To better satisfy the UFM assumption, we augmented each backbone with two additional linear layers that maintain the same dimension as the backbone output. The mapped features $h_\theta(x)$ are then passed through a final linear layer without bias to produce the one-dimensional latent variable $z = w^\top h_\theta(x)$, where w is the classifier weight analyzed in this study.

Other experimental settings. For the tabular datasets, all the models were trained for 5000 epochs with the Adam optimizer, starting from the initial learning rates listed in Table 3. We made the learning rate to decay by a factor of 0.1 at epochs 200, 800, and 3000. We used a batch size of 2048 and applied a weight decay of 5×10^{-3} to all network parameters (except that the learnable thresholds, when present, received zero weight decay). For each dataset we tested the four possible combinations of the three link functions (logit, probit, clog-log) and the two threshold strategies (fixed, learnable). Running every configuration on the 30 predefined hold-out splits yielded a total of 600 training–validation runs.

To ensure that the training error reached (near-)zero—the regime where feature collapse is observed—we tuned only the initial learning rate and, for fixed thresholds, the threshold range. Because the logistic function has heavier tails than the normal CDF, the logit runs used a wider fixed range $[-20, 20]$, whereas the probit runs used $[-2, 2]$. The complete hyper-parameter grid is summarized in Table 3.

For the UTKFace dataset, models were trained for 300 epochs using the Adam optimizer with a batch size of 512 per GPU across 5 GPUs (total effective batch size 2560). We tested two threshold strategies (fixed, learnable), using only the logit link function. Starting from the initial learning rates listed in Table 4, the learning rate was decayed by a factor of 0.1 at epochs 100 and 200. Each configuration was repeated with three different random seeds. The other hyper-parameter settings for each backbone are summarized in Table 4.

Experiments compute resources. For the tabular datasets, every run was executed on a single NVIDIA RTX A6000 (48 GB) GPU, taking roughly 5–10 minutes and requiring only a few hundred megabytes of GPU memory. For the UTKFace dataset, training was performed across 5 NVIDIA RTX A6000 GPUs using distributed data parallel, with each configuration taking approximately 6 hours to complete 300 epochs.

Table 3: Hyper-parameter settings for tabular datasets with three link functions (logit, probit, clog-log) and two threshold strategies (fixed, learnable).

Dataset	Link function	Thresholds	Threshold range (when fixed)	Initial learning rate
ER	logit	fixed	$[-20, 20]$	1×10^{-2}
	logit	learnable	–	1×10^{-2}
	probit	fixed	$[-2, 2]$	1×10^{-3}
	probit	learnable	–	1×10^{-3}
	clog-log	fixed	$[-2, 2]$	5×10^{-3}
	clog-log	learnable	–	5×10^{-3}
LE	logit	fixed	$[-20, 20]$	1×10^{-2}
	logit	learnable	–	1×10^{-2}
	probit	fixed	$[-2, 2]$	5×10^{-3}
	probit	learnable	–	5×10^{-3}
	clog-log	fixed	$[-2, 2]$	5×10^{-3}
	clog-log	learnable	–	5×10^{-3}
SW	logit	fixed	$[-20, 20]$	1×10^{-2}
	logit	learnable	–	1×10^{-2}
	probit	fixed	$[-2, 2]$	5×10^{-3}
	probit	learnable	–	5×10^{-3}
	clog-log	fixed	$[-2, 2]$	5×10^{-3}
	clog-log	learnable	–	5×10^{-3}
CA	logit	fixed	$[-20, 20]$	1×10^{-2}
	logit	learnable	–	1×10^{-2}
	probit	fixed	$[-2, 2]$	5×10^{-3}
	probit	learnable	–	5×10^{-3}
	clog-log	fixed	$[-2, 2]$	5×10^{-3}
	clog-log	learnable	–	5×10^{-3}
WR	logit	fixed	$[-20, 20]$	1×10^{-2}
	logit	learnable	–	1×10^{-2}
	probit	fixed	$[-2, 2]$	1×10^{-3}
	probit	learnable	–	1×10^{-3}
	clog-log	fixed	$[-2, 2]$	1×10^{-3}
	clog-log	learnable	–	1×10^{-3}

Table 4: Hyper-parameter settings for UTKFace experiments with logit link function and two threshold strategies (fixed, learnable).

Backbone	Thresholds	Threshold range (when fixed)	Initial learning rate	Weight decay
ResNet50	fixed	$[-40, 40]$	1×10^{-3}	1×10^{-4}
	learnable	–	1×10^{-3}	1×10^{-4}
ResNet101	fixed	$[-40, 40]$	1×10^{-3}	1×10^{-3}
	learnable	–	1×10^{-3}	1×10^{-3}
DenseNet201	fixed	$[-40, 40]$	5×10^{-4}	1×10^{-4}
	learnable	–	5×10^{-4}	1×10^{-4}

D Additional experimental results

D.1 Results with the logit model

This section presents the experimental outcomes obtained using the logistic function (i.e., $g(x) = (1 + e^{-x})^{-1}$), which corresponds to the logit model. Figures 7–10 show the evolution of evaluation-metric curves for the datasets LE, SW, CA, and WR, respectively. Figures 11–14 show visualization of the latent and feature spaces for the datasets LE, SW, CA, and WR, respectively. These exhibit a consistent behavior with the one of the ER dataset in the main text (Figs. 2 and 4).

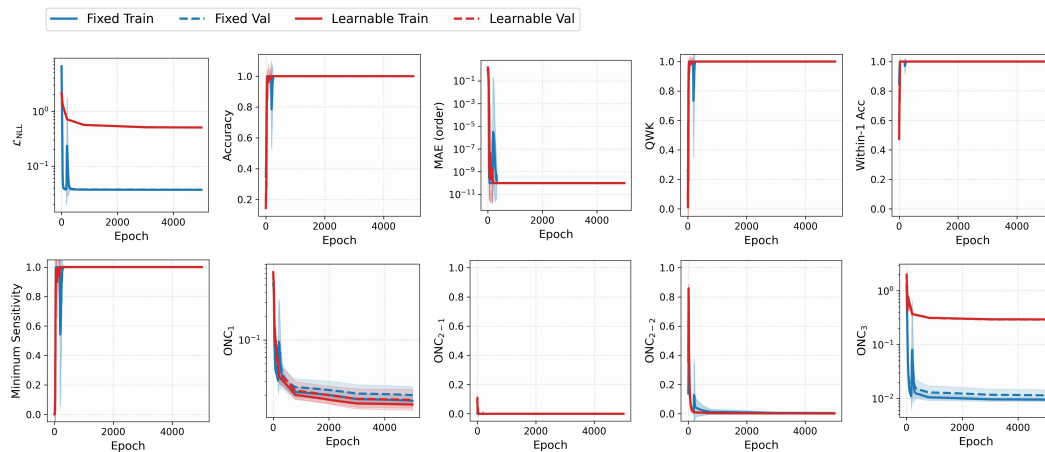


Figure 7: Epoch-wise average metrics curves for the LE dataset with the logit model, comparing fixed- and learnable-threshold models.

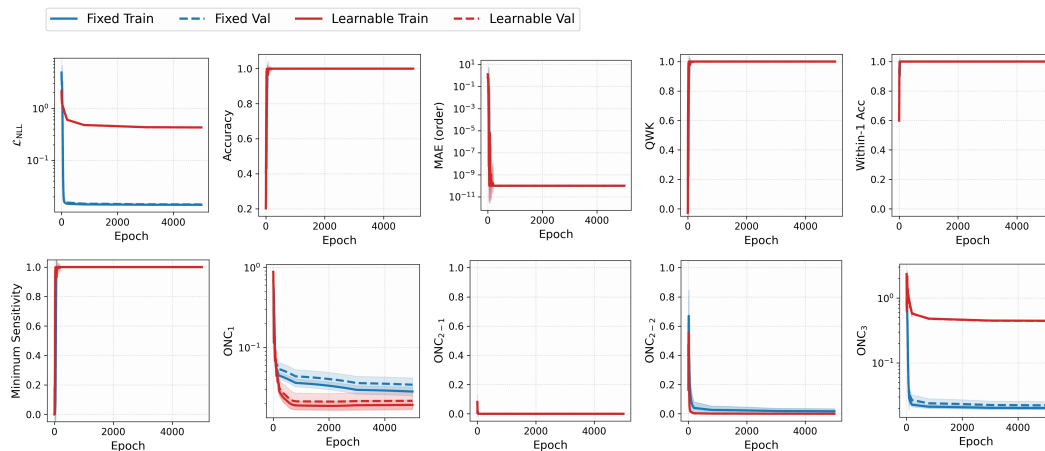


Figure 8: Epoch-wise average metrics curves for the SW dataset with the logit model, comparing fixed- and learnable-threshold models.

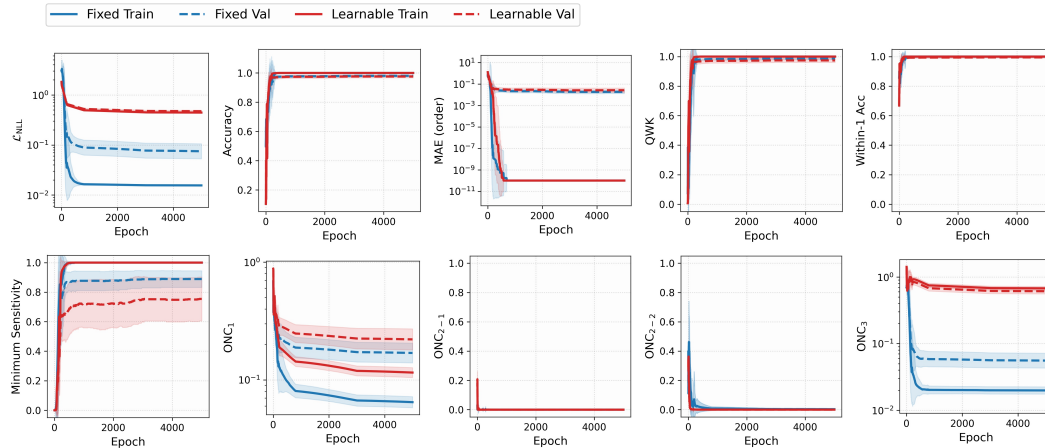


Figure 9: Epoch-wise average metrics curves for the CA dataset with the logit model, comparing fixed- and learnable-threshold models.

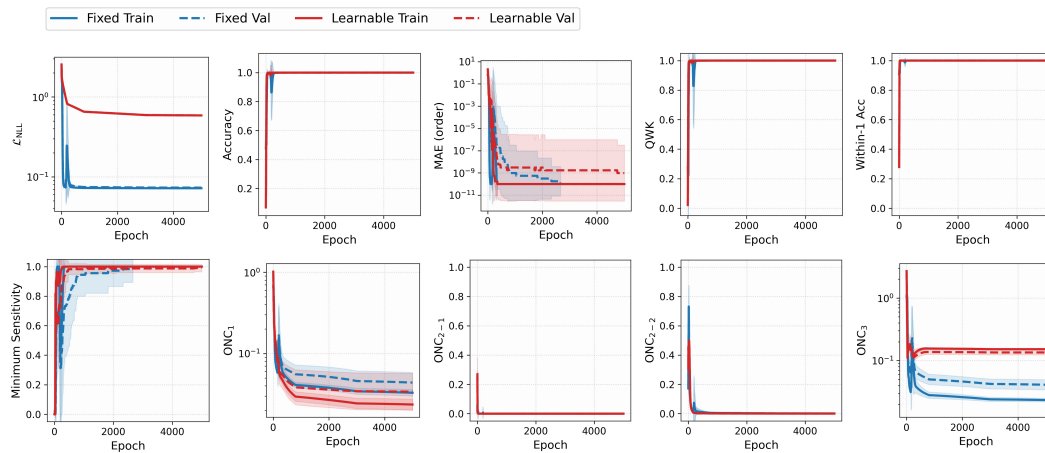


Figure 10: Epoch-wise average metrics curves for the WR dataset with the logit model, comparing fixed- and learnable-threshold models.

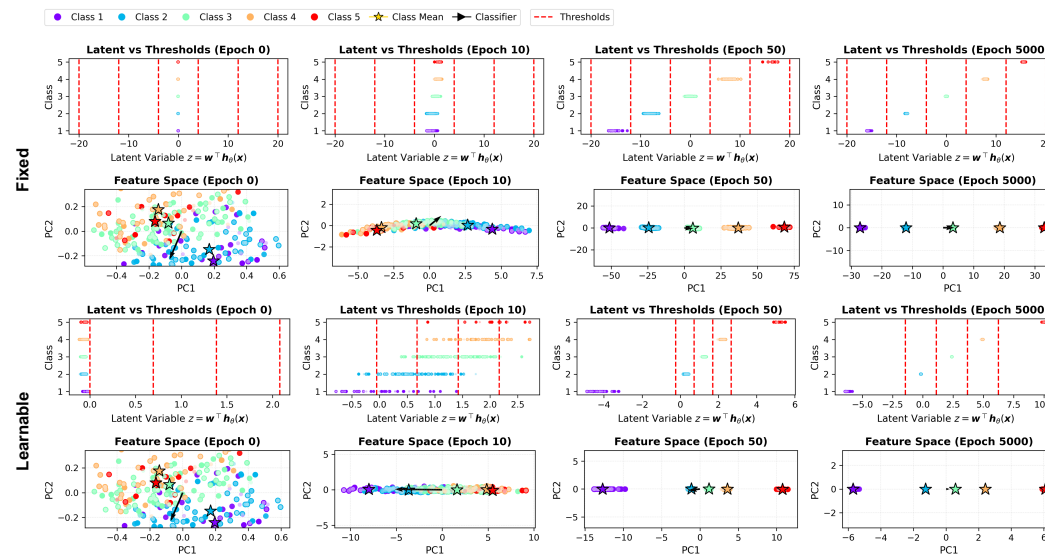


Figure 11: Visualization of the latent and feature spaces for the LE dataset using the logit model, comparing fixed- and learnable-threshold models.

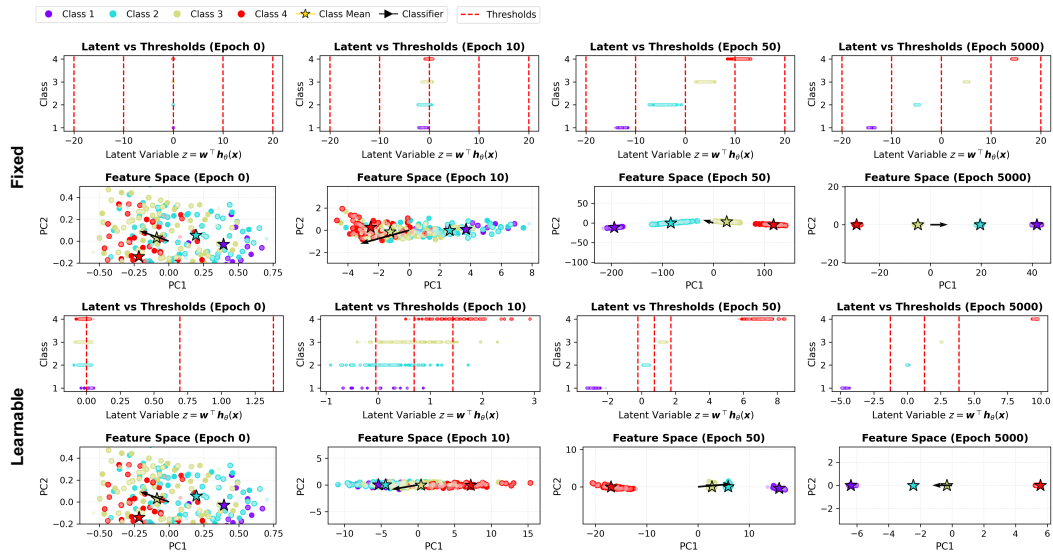


Figure 12: Visualization of the latent and feature spaces for the SW dataset using the logit model, comparing fixed- and learnable-threshold models.

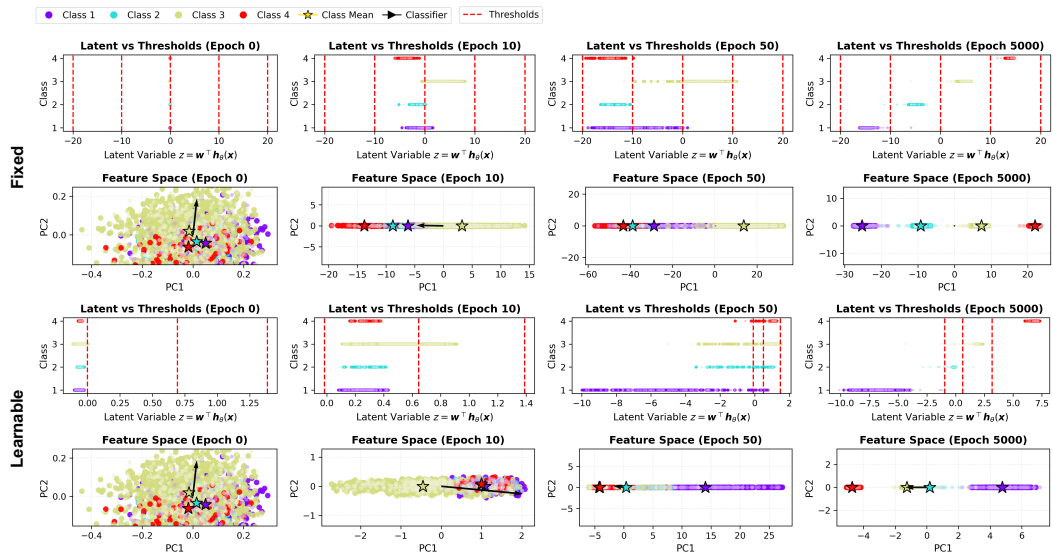


Figure 13: Visualization of the latent and feature spaces for the CA dataset using the logit model, comparing fixed- and learnable-threshold models.

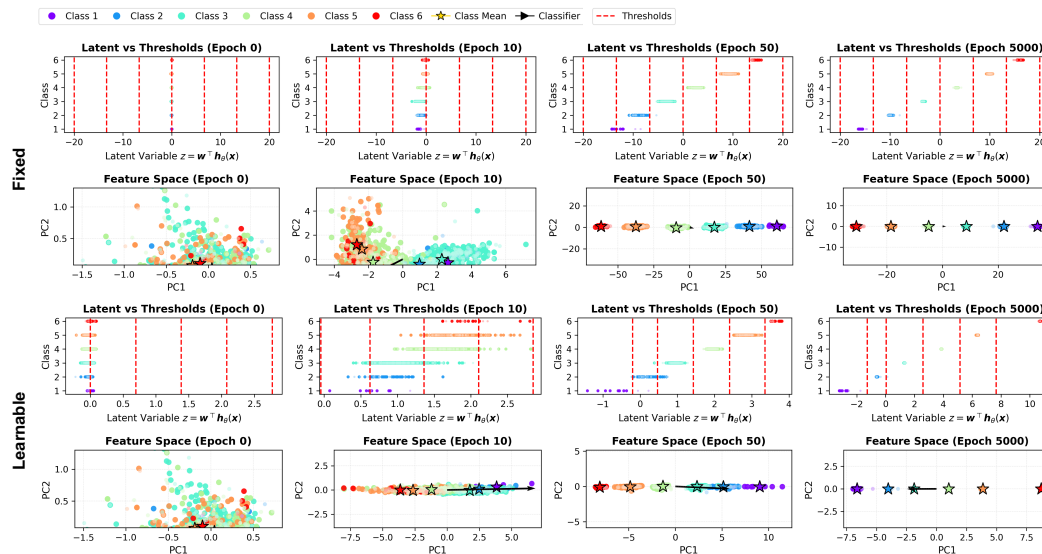


Figure 14: Visualization of the latent and feature spaces for the WR dataset using the logit model, comparing fixed- and learnable-threshold models.

D.2 Results with the probit model

This section presents the experimental outcomes obtained using the normal CDF (i.e., $g(x) = \Phi(x)$), which corresponds to the probit model. Figures 15–19 show the evolution of evaluation-metric curves for the datasets ER, LE, SW, CA, and WR, respectively. Figures 20–24 show visualization of the latent and feature spaces for the datasets ER, LE, SW, CA, and WR, respectively. These again show a consistent behavior with that in the main text.

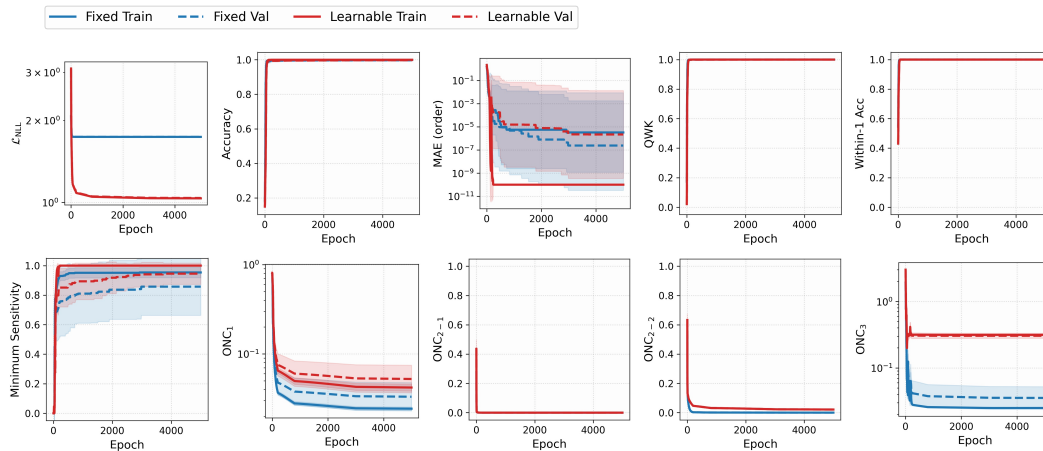


Figure 15: Epoch-wise average metrics curves for the ER dataset with the probit model, comparing fixed- and learnable-threshold models.

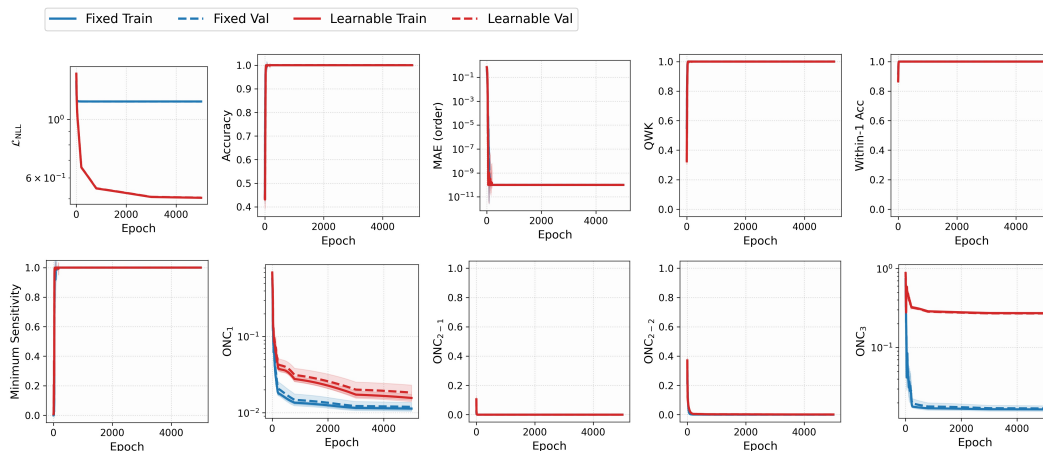


Figure 16: Epoch-wise average metrics curves for the LE dataset with the probit model, comparing fixed- and learnable-threshold models.

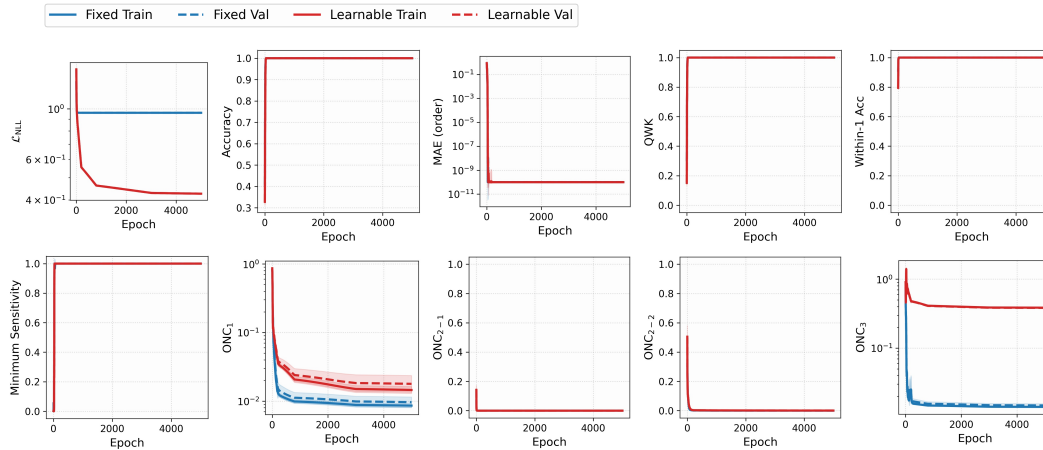


Figure 17: Epoch-wise average metrics curves for the SW dataset with the probit model, comparing fixed- and learnable-threshold models.

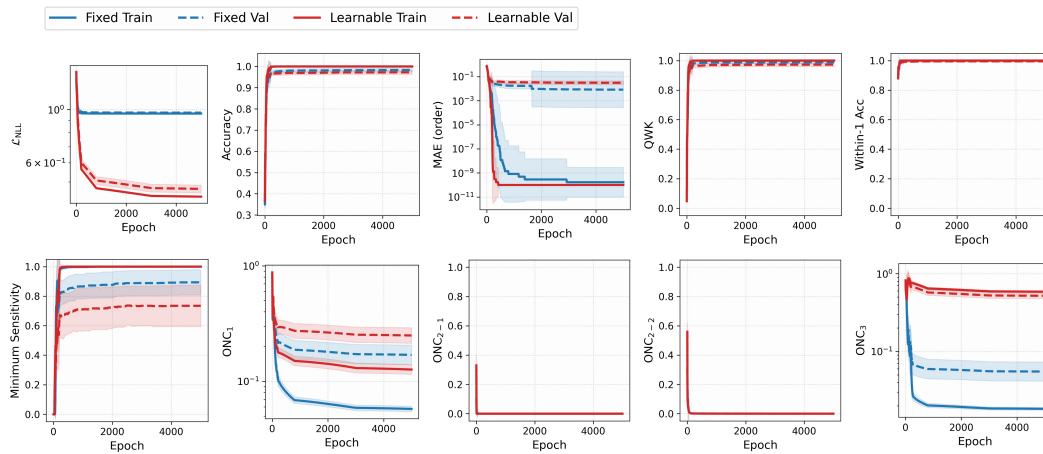


Figure 18: Epoch-wise average metrics curves for the CA dataset with the probit model, comparing fixed- and learnable-threshold models.

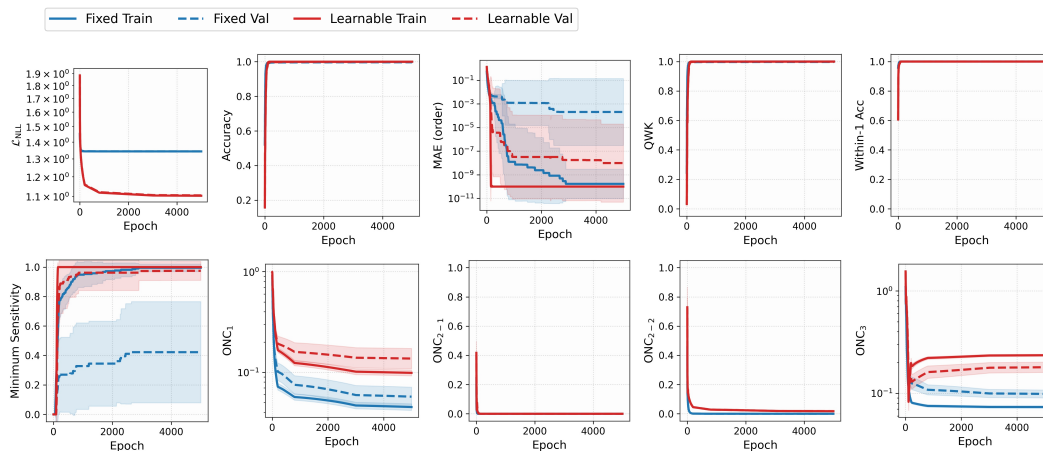


Figure 19: Epoch-wise average metrics curves for the WR dataset with the probit model, comparing fixed- and learnable-threshold models.

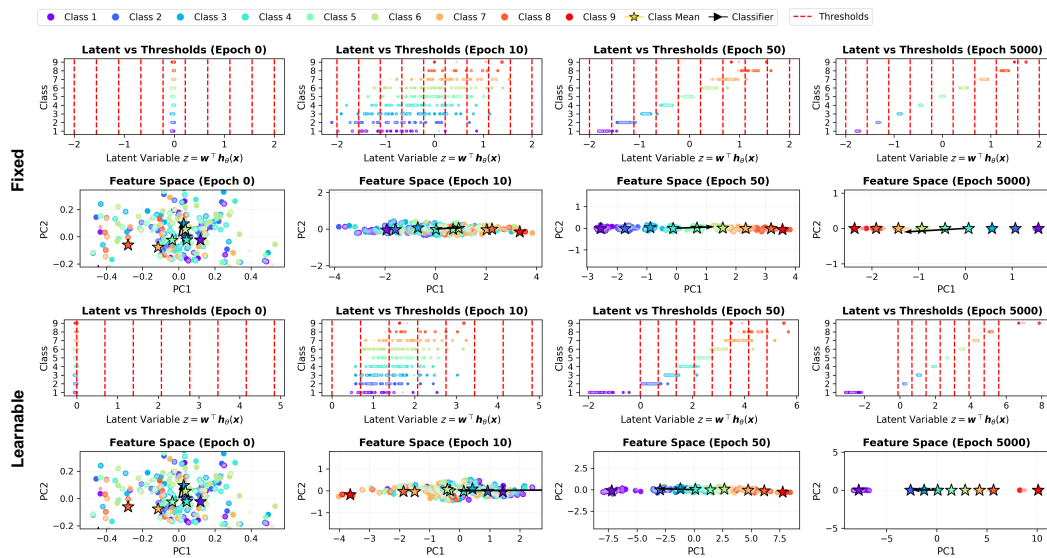


Figure 20: Visualization of the latent and feature spaces for the ER dataset using the probit model, comparing fixed- and learnable-threshold models.

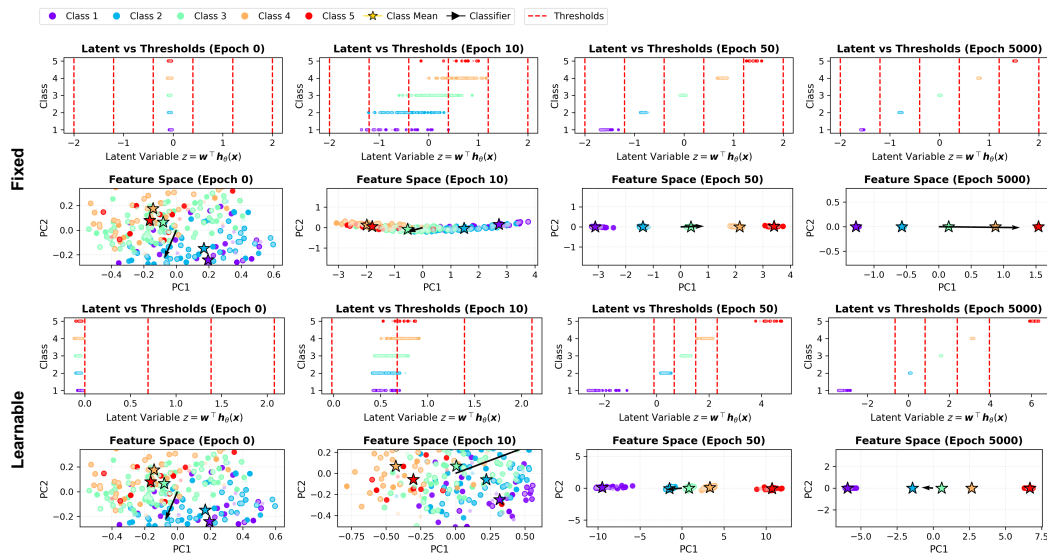


Figure 21: Visualization of the latent and feature spaces for the LE dataset using the probit model, comparing fixed- and learnable-threshold models.

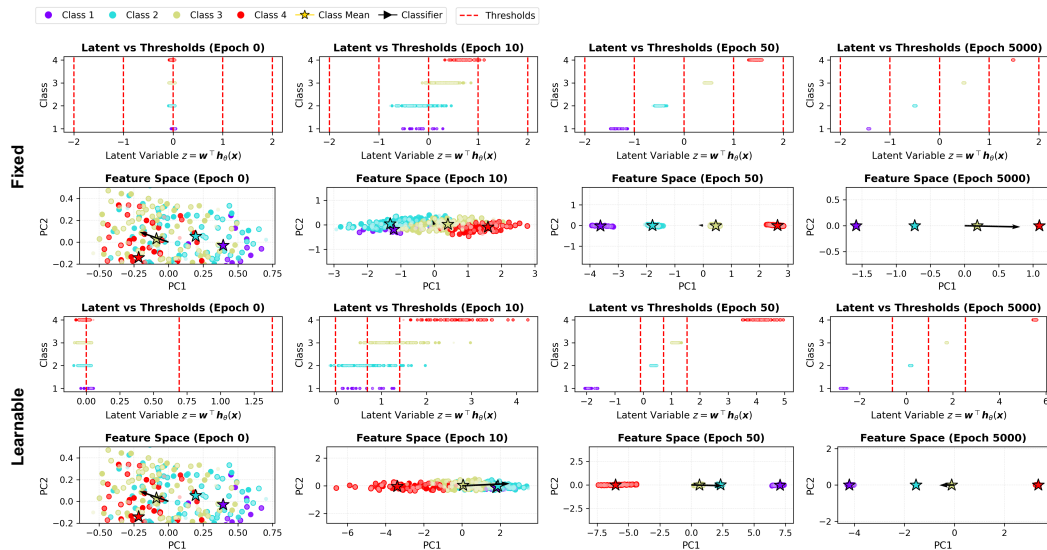


Figure 22: Visualization of the latent and feature spaces for the SW dataset using the probit model, comparing fixed- and learnable-threshold models.

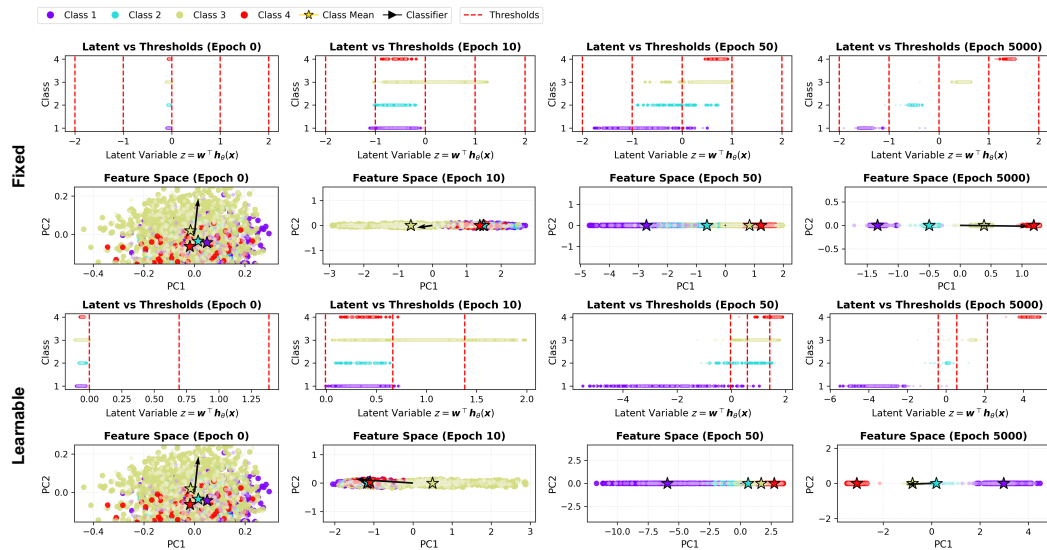


Figure 23: Visualization of the latent and feature spaces for the CA dataset using the probit model, comparing fixed- and learnable-threshold models.

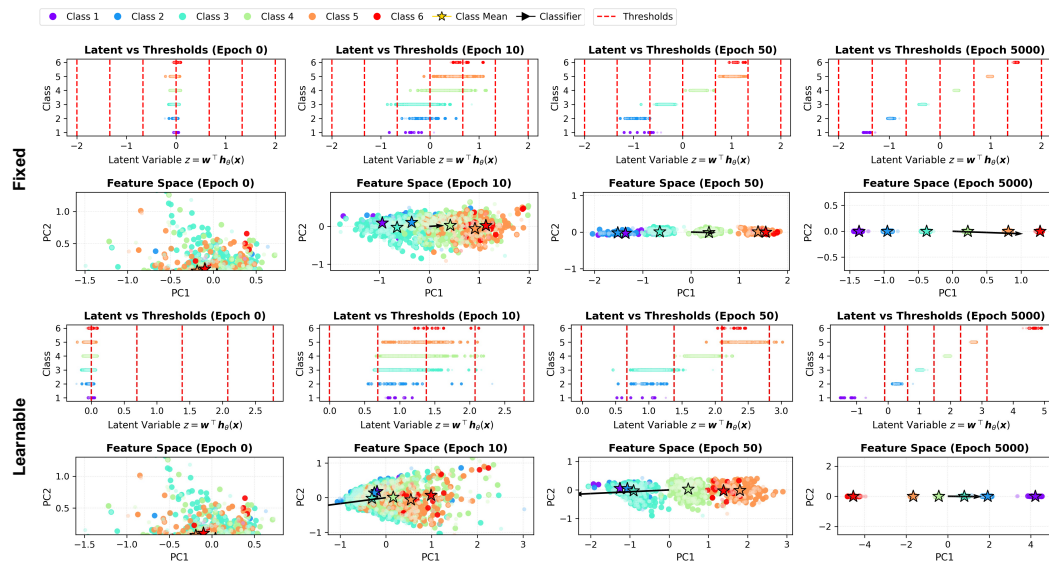


Figure 24: Visualization of the latent and feature spaces for the WR dataset using the probit model, comparing fixed- and learnable-threshold models.

D.3 Results with the clog-log model

This section presents the experimental outcomes obtained using the Gumbel CDF (i.e., $g(x) = 1 - e^{-e^x}$), which corresponds to the clog-log model.

Unlike the symmetric link functions (logit and probit), the clog-log link is asymmetric. The solution to (18) thus no longer places optimal latent variables at the simple midpoint between adjacent thresholds, but shifts them by an intrinsic offset.

As the examined fixed threshold cases so far, we assume the uniformly spaced fixed thresholds $b_q^{\text{fix}} = b_0^{\text{fix}} + q\Delta_b^{\text{fix}}$ where $\Delta_b^{\text{fix}} = (b_Q^{\text{fix}} - b_0^{\text{fix}})/Q$. In this case, we have the optimal solution as

$$z_q^* = \frac{b_q^{\text{fix}} + b_{q-1}^{\text{fix}}}{2} + \sigma \quad (58)$$

for a constant offset σ . To verify this, we substitute the above expression into (18), which requires $g'(b_q^{\text{fix}} - z_q^*) = g'(b_{q-1}^{\text{fix}} - z_q^*)$. Computing the arguments gives:

$$b_q^{\text{fix}} - z_q^* = b_q^{\text{fix}} - \frac{b_q^{\text{fix}} + b_{q-1}^{\text{fix}}}{2} - \sigma = \frac{\Delta_b^{\text{fix}}}{2} - \sigma, \quad (59)$$

and

$$b_{q-1}^{\text{fix}} - z_q^* = b_{q-1}^{\text{fix}} - \frac{b_q^{\text{fix}} + b_{q-1}^{\text{fix}}}{2} - \sigma = -\frac{\Delta_b^{\text{fix}}}{2} - \sigma. \quad (60)$$

Substituting these into (18), we obtain:

$$g'\left(\frac{\Delta_b^{\text{fix}}}{2} - \sigma\right) = g'\left(-\frac{\Delta_b^{\text{fix}}}{2} - \sigma\right). \quad (61)$$

Since this equation depends only on Δ_b^{fix} and not on q , the same offset σ applies uniformly to all class intervals. In practice, σ can be computed as $\sigma = z_1^* - (b_1^{\text{fix}} + b_0^{\text{fix}})/2$ once z_1^* is determined from (18).

Accordingly, for the uniformly spaced fixed thresholds, we introduce a modified ONC3 metric that accounts for the asymmetry of the link function:

$$\text{ONC}_{3,\text{clog-log}} = \frac{\sum_{q=1}^{Q-1} |b_q^{\text{fix}} - b_q^{\text{ideal}}|}{\sum_{q=1}^{Q-1} (b_{q+1}^{\text{fix}} - b_q^{\text{fix}})}, \quad (62)$$

where the ideal threshold position b_q^{ideal} is defined as

$$b_q^{\text{ideal}} = \frac{z_q + z_{q+1}}{2} - \sigma. \quad (63)$$

Note: To derive this expression, recall that for uniformly spaced fixed thresholds, the optimal latent variables satisfy $z_q^* = (b_q^{\text{fix}} + b_{q-1}^{\text{fix}})/2 + \sigma$ and $z_{q+1}^* = (b_{q+1}^{\text{fix}} + b_q^{\text{fix}})/2 + \sigma$. Adding these two equations yields:

$$z_q^* + z_{q+1}^* = \frac{b_q^{\text{fix}} + b_{q-1}^{\text{fix}}}{2} + \frac{b_{q+1}^{\text{fix}} + b_q^{\text{fix}}}{2} + 2\sigma = \frac{b_{q-1}^{\text{fix}} + 2b_q^{\text{fix}} + b_{q+1}^{\text{fix}}}{2} + 2\sigma. \quad (64)$$

For uniformly spaced thresholds with $b_{q+1}^{\text{fix}} - b_q^{\text{fix}} = b_q^{\text{fix}} - b_{q-1}^{\text{fix}} = \Delta_b^{\text{fix}}$, we have $b_{q-1}^{\text{fix}} + b_{q+1}^{\text{fix}} = 2b_q^{\text{fix}}$. Substituting this gives:

$$z_q^* + z_{q+1}^* = \frac{4b_q^{\text{fix}}}{2} + 2\sigma = 2b_q^{\text{fix}} + 2\sigma. \quad (65)$$

Rearranging, we obtain:

$$b_q^{\text{fix}} = \frac{z_q^* + z_{q+1}^*}{2} - \sigma. \quad (66)$$

This establishes the form of the ideal threshold position.

In the above analysis, the assumption that fixed thresholds are evenly spaced, but this assumption does not hold for learnable thresholds. Even for symmetric link functions such as the logit model, the

same limitation applies; however, in the main text, we have presented the results without resolving this issue. The rationale for this choice is discussed in Section 6.

Fortunately, for symmetric cases, the error caused by this simplification is not severe, and therefore the conclusions in the main text remain valid. In contrast, for asymmetric link functions, the error introduced by this issue is expected to be more significant. Thus, for the general case—including learnable thresholds—it is necessary to appropriately extend ONC3 and define a new quantity that can properly quantify the phenomenon. However, doing so would make systematic comparison with the existing experiments in the main text difficult, as it would create a mismatch from the experimental setup in the main text. Therefore, for the clog-log model, we only quantify ONC3 for the fixed-threshold case and omit the ONC3 quantification for the learnable-threshold case.

Figures 25–29 show the evolution of evaluation-metric curves for the datasets ER, LE, SW, CA, and WR, respectively. Due to the limitation for learnable thresholds, we only present $\text{ONC}_{3,\text{clog-log}}$ curves for the fixed threshold model. Figures 30–34 show visualization of the latent and feature spaces for the datasets ER, LE, SW, CA, and WR, respectively. These results align with the ones in the case of the symmetric link functions, suggesting the wide applicability of ONC.

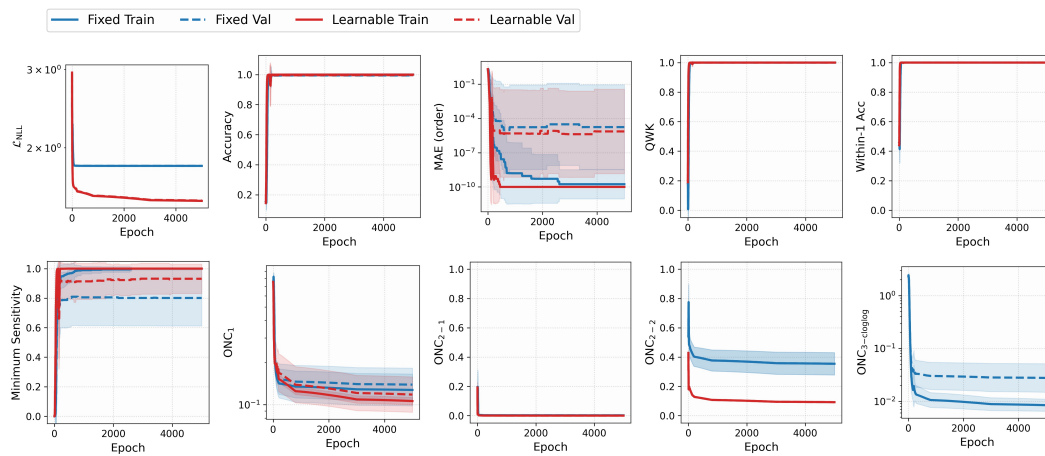


Figure 25: Epoch-wise average metrics curves for the ER dataset with the clog-log model, comparing fixed- and learnable-threshold models.

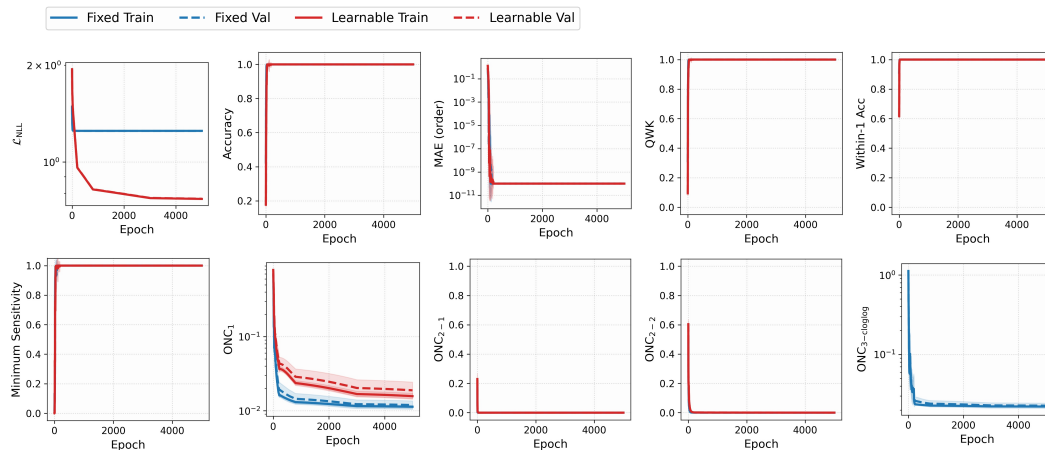


Figure 26: Epoch-wise average metrics curves for the LE dataset with the clog-log model, comparing fixed- and learnable-threshold models.

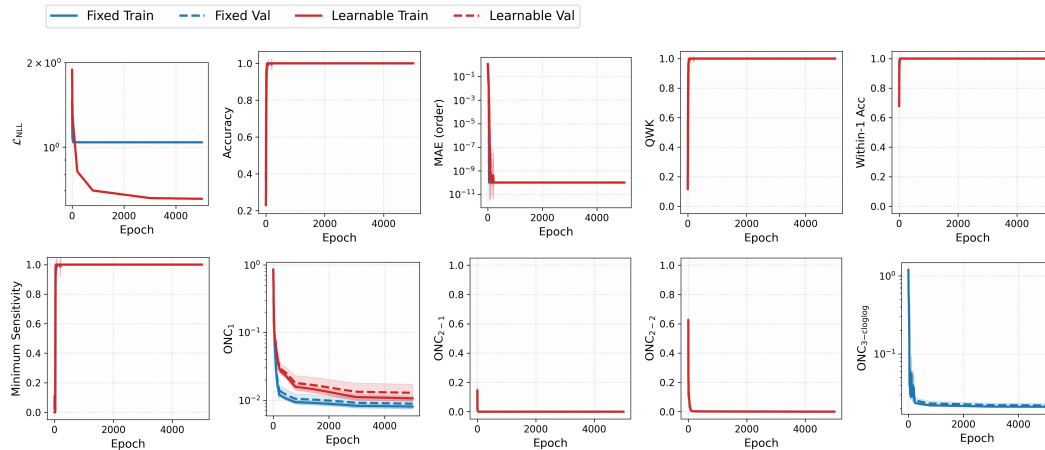


Figure 27: Epoch-wise average metrics curves for the SW dataset with the clog-log model, comparing fixed- and learnable-threshold models.

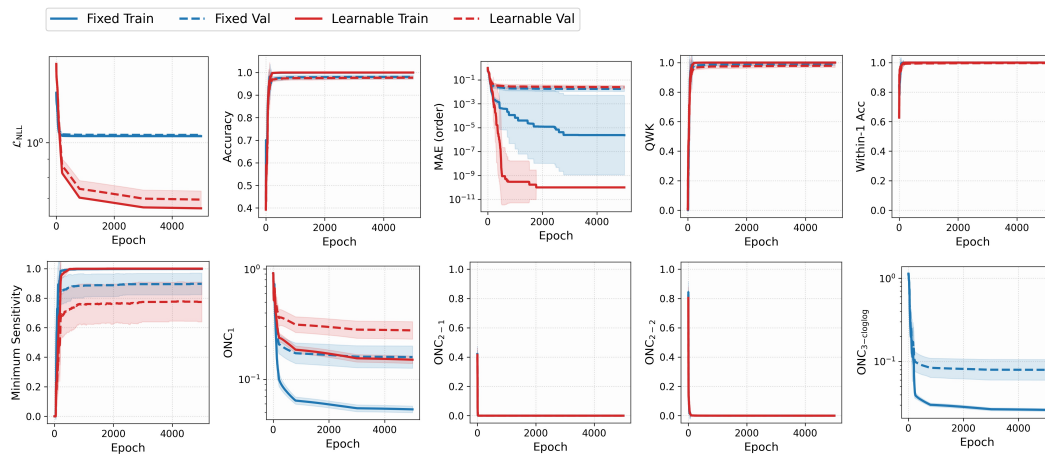


Figure 28: Epoch-wise average metrics curves for the CA dataset with the clog-log model, comparing fixed- and learnable-threshold models.

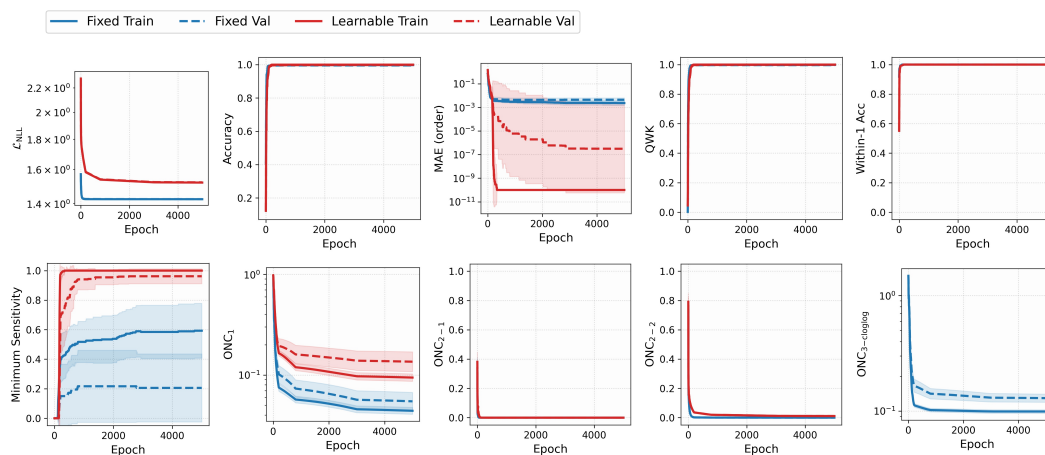


Figure 29: Epoch-wise average metrics curves for the WR dataset with the clog-log model, comparing fixed- and learnable-threshold models.

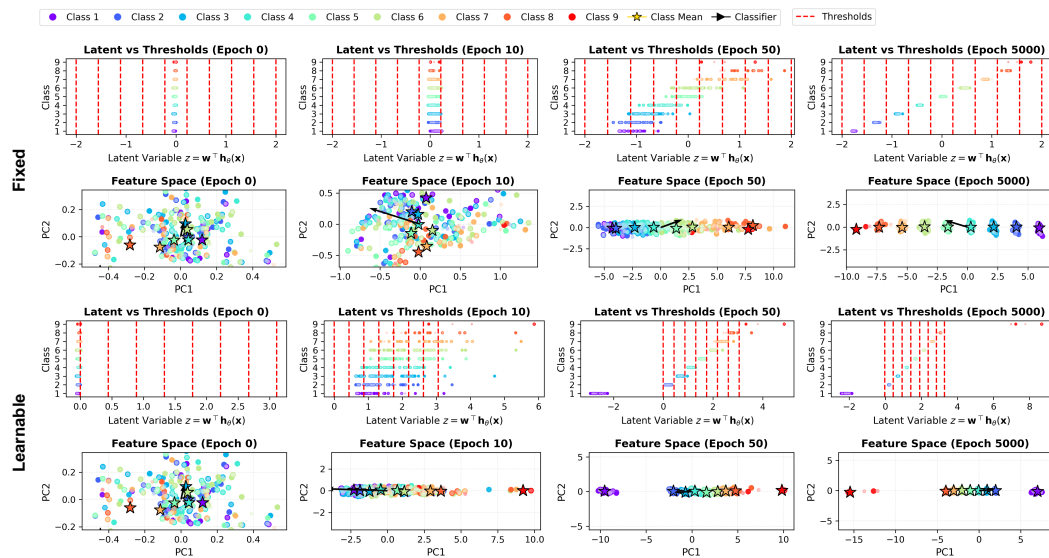


Figure 30: Visualization of the latent and feature spaces for the ER dataset using the clog-log model, comparing fixed- and learnable-threshold models.

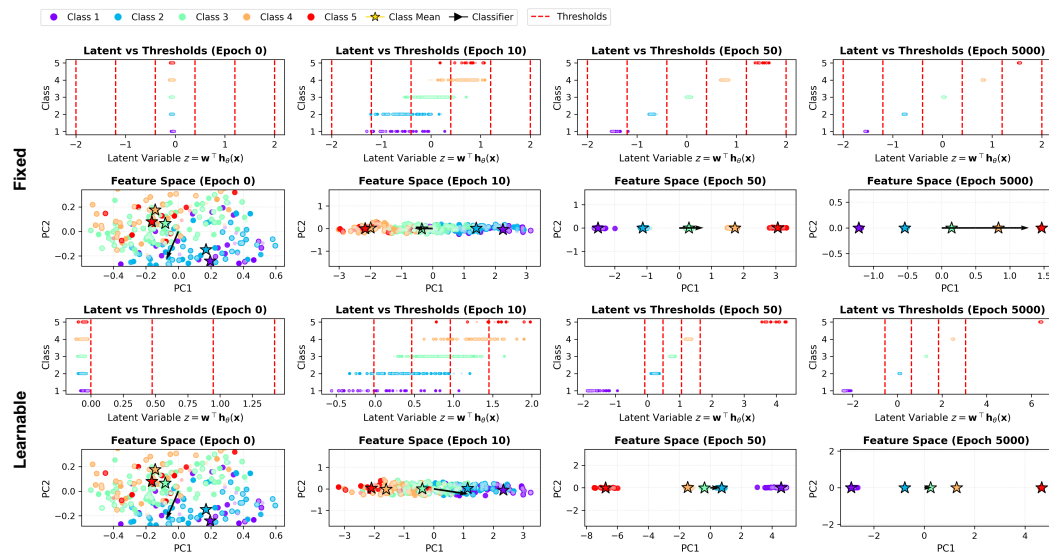


Figure 31: Visualization of the latent and feature spaces for the LE dataset using the clog-log model, comparing fixed- and learnable-threshold models.

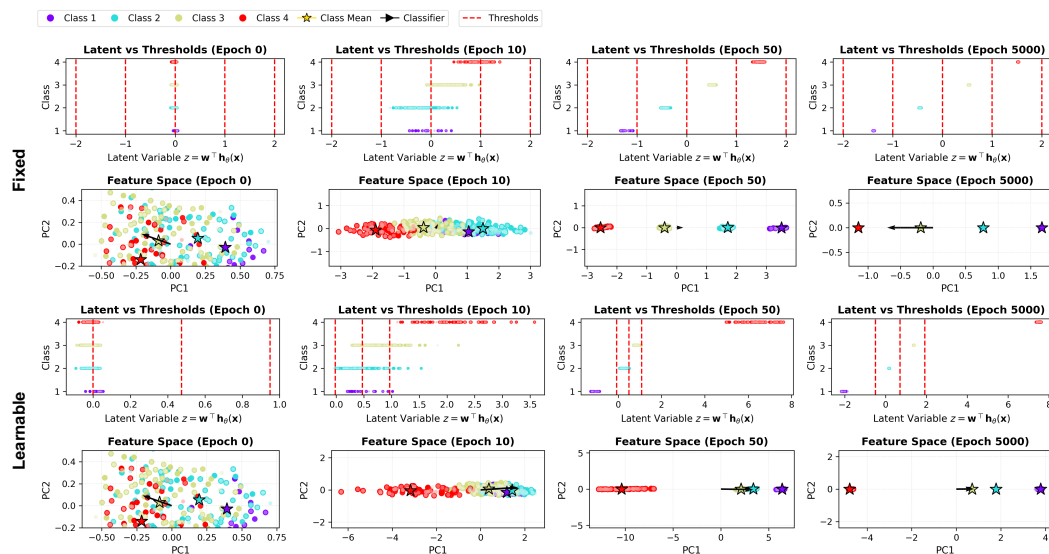


Figure 32: Visualization of the latent and feature spaces for the SW dataset using the clog-log model, comparing fixed- and learnable-threshold models.

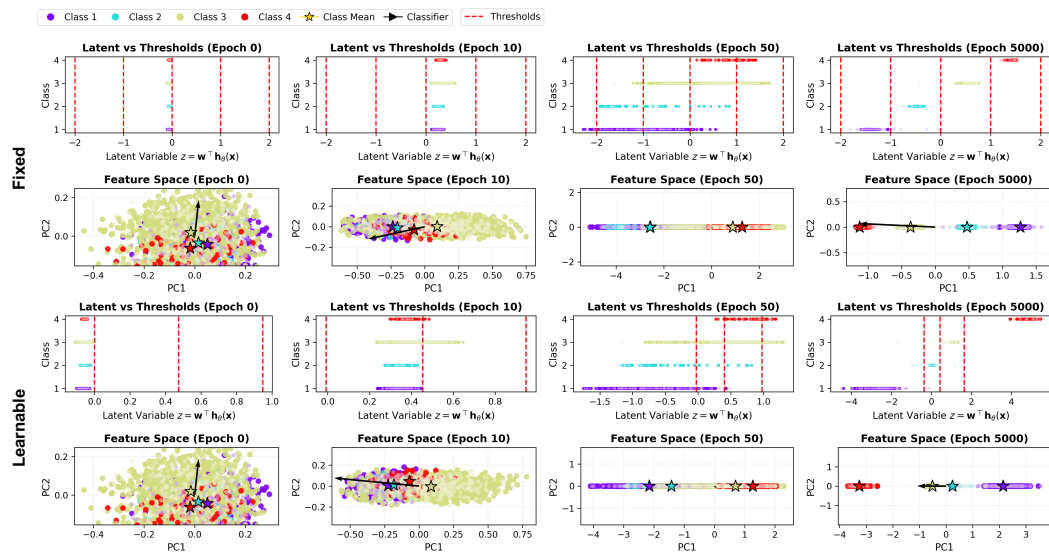


Figure 33: Visualization of the latent and feature spaces for the CA dataset using the clog-log model, comparing fixed- and learnable-threshold models.

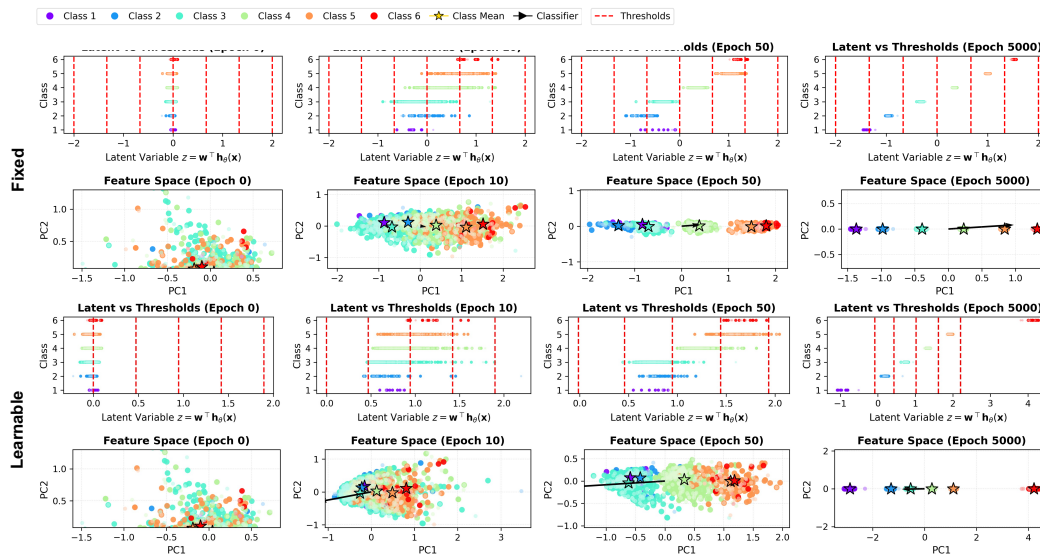


Figure 34: Visualization of the latent and feature spaces for the WR dataset using the clog-log model, comparing fixed- and learnable-threshold models.

D.4 Results on UTKFace dataset with the ResNet50 and DenseNet201 backbones

This section presents the experimental outcomes for the UTKFace dataset using the logit model. Figures 35 and 36 show the evolution of evaluation-metric curves for the ResNet50 and DenseNet201 backbones, respectively. Figures 37 and 38 show visualization of the latent and feature spaces for the ResNet50 and DenseNet201 backbones, respectively (one random seed per backbone). These results are essentially consistent with those obtained using ResNet101 in the main text, suggesting that the effectiveness of ONC and fixed thresholds broadly holds even for complex backbone networks used in practice.

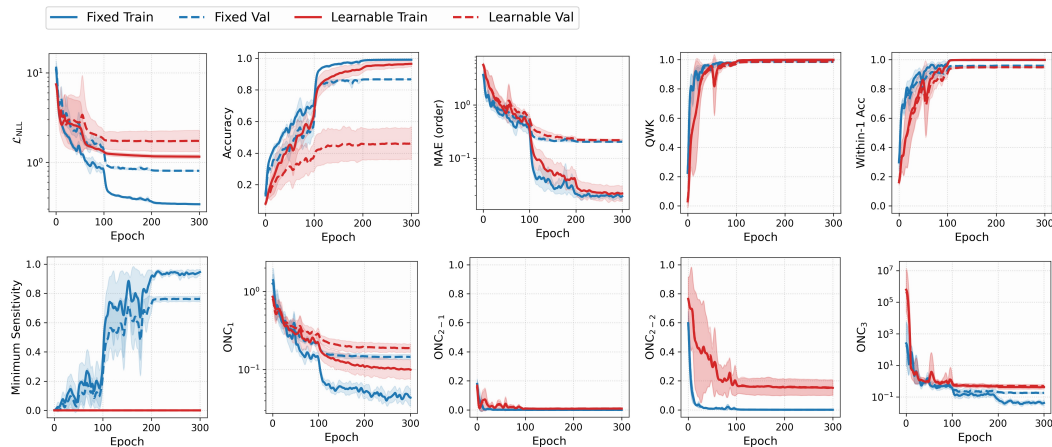


Figure 35: Epoch-wise average metrics curves for the UTKFace dataset with the ResNet50 backbone.

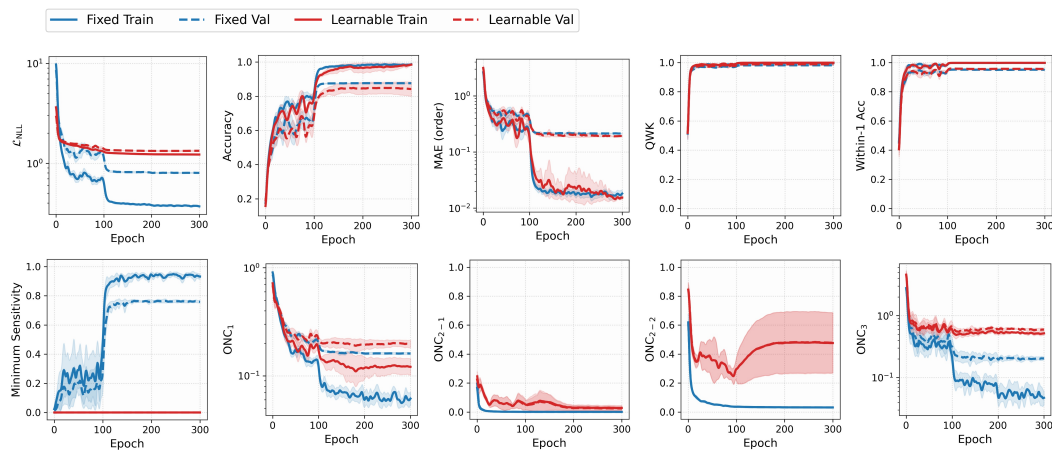


Figure 36: Epoch-wise average metrics curves for the UTKFace dataset with the DenseNet201 backbone.

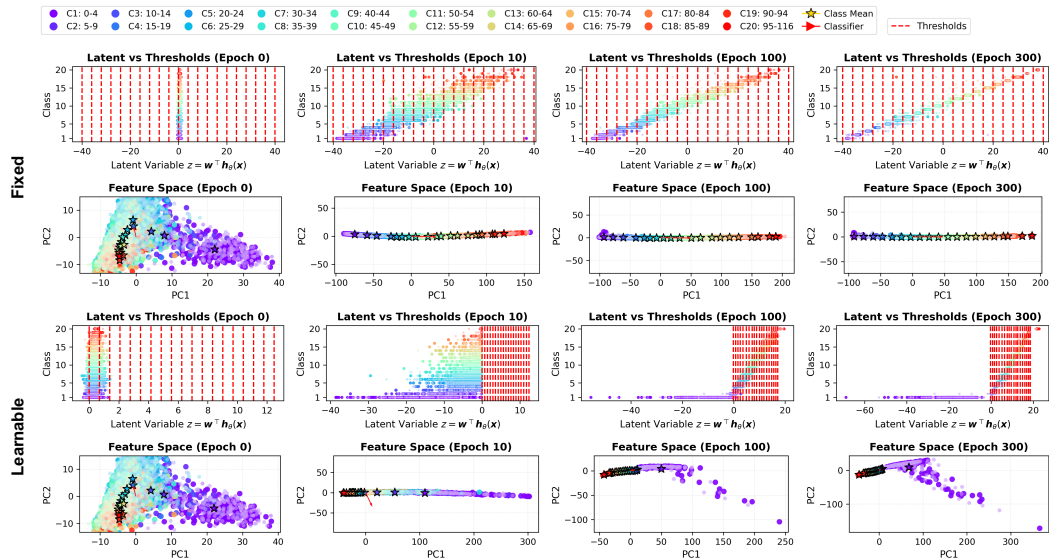


Figure 37: Visualization of the latent and feature spaces for the UTKFace dataset with the ResNet50 backbone.

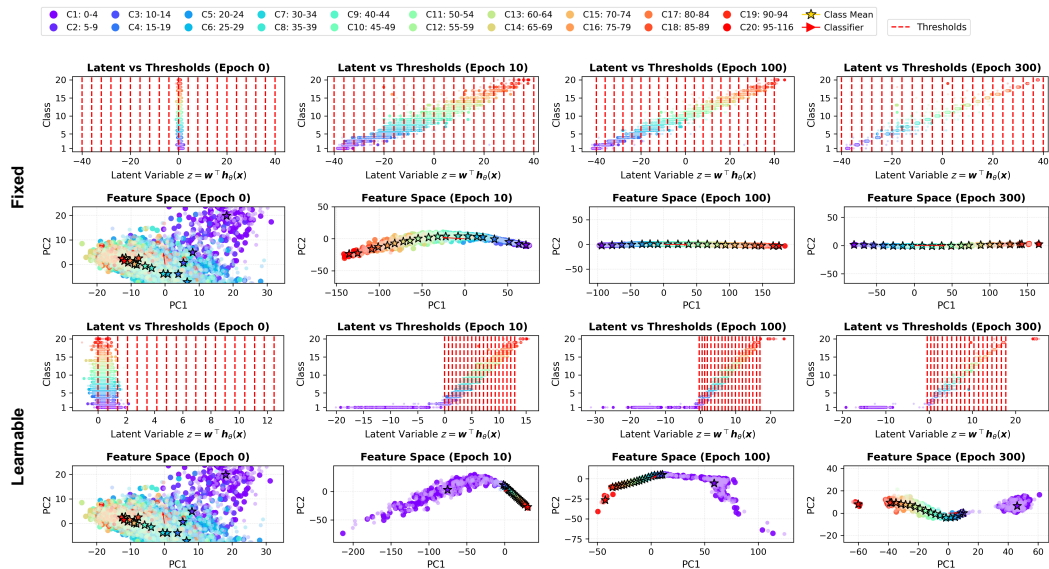


Figure 38: Visualization of the latent and feature spaces for the UTKFace dataset with the DenseNet201 backbone.

D.5 Comparison of theoretical and experimental results

We here compare theoretical predictions from EOS (16) with experimental results on the LE dataset ($Q = 5$ classes) as a representative example, assuming the logit model. For the experiment, we use the same network architecture and the dataset splits as the main text, but the training objective is slightly changed to the following one:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{NLL}} + \frac{\lambda_h}{2N} \sum_{i=1}^N \|\mathbf{h}_\theta(\mathbf{x}_i)\|_2^2 + \frac{\lambda_w}{2} \|\mathbf{w}\|_2^2 + \lambda_\theta \|\boldsymbol{\theta}\|^2, \quad (67)$$

where $\mathcal{L}_{\text{NLL}}$ denotes the negative log-likelihood, and $\lambda_\theta = 5 \times 10^{-3}$ and $\lambda_h = 0.01$ are the regularization coefficients for the backbone network parameter $\boldsymbol{\theta}$ and the feature vector $\mathbf{h}_\theta$ (output of the backbone network), respectively: the latter regularization term was absent in the experiments in the other sections but is here introduced to make a direct quantitative comparison with our theory. We trained the whole network under the fixed thresholds set as $\mathbf{b} = (-20, -12, -4, 4, 12, 20)$ for different 35 values of $\lambda_w \in [0, 10^4]$.

The results of the experiment (datapoints with error bars) are given in Fig. 39. The left panel

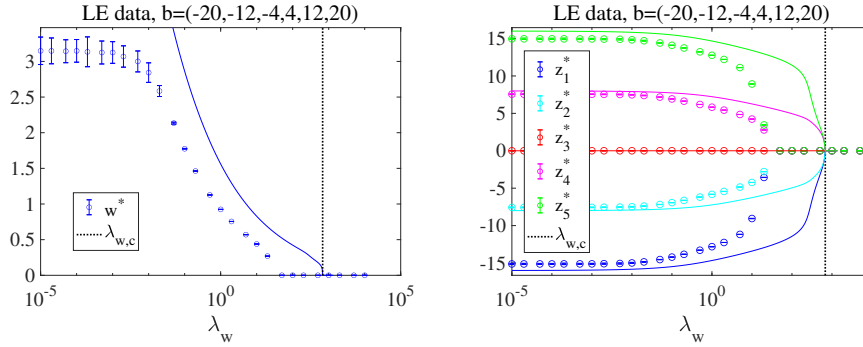


Figure 39: Theory (solid curves from EOS) versus experiment (datapoints, 30-holdout average with error bars) for the LE dataset. Left: $\|\mathbf{w}^*\|$ versus λ_w . Right: z_q^* versus λ_w .

shows w^* while the right one exhibits z^* . In the plots, the theoretical predictions (solid curves) are simultaneously shown: they are computed from EOS (16) with the above parameter values ($\mathbf{b}$, λ_h), the dataset size ratios $\alpha = (0.094, 0.281, 0.472, 0.231, 0.031)$ of the LE dataset, and the respective λ_w value. The right panel shows good agreement for z^* in the small- λ_w region where the relation $z_q^* \rightarrow (b_q + b_{q-1})/2$ holds, but the left panel exhibits a huge gap in $w^* = \|\mathbf{w}^*\|_2$: the DNN exhibits a saturating behavior at small λ_w while the theory predicts the divergence, implying that our theory is quantitatively not accurate. On the other hand, both the theoretical and experimental results show similar qualitative dependence on λ_w : they vary monotonically with λ_w , and in both cases, phase transitions occur at certain specific values of λ_w , though the locations of the transition points are quantitatively different. We confirmed that similar behavior is also observed for the other four tabular datasets.

In summary, although the theory developed in this study does not quantitatively predict the behavior of actual DNNs with full accuracy, it can qualitatively explain the observed phenomena.

NeurIPS Paper Checklist

1. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We have provided a paragraph in the Discussion section to discuss the limitations of the theoretical results, including the assumption of the fixed threshold, as well as that on the number of phases in the (λ_h, λ_w) -plane. Additionally, we acknowledge that the current definition of ONC3 is only valid when the thresholds are fixed and evenly spaced in the latent space, which is the case for our fixed threshold experiments but is not for the others. We note that developing generally applicable ONC3 metrics is possible but the corresponding experiment is left for future work.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

2. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We have provided all the assumptions for the theorems in the main text (Theorems 4.1, 4.2, and 4.3), as well as those in the appendices (Theorems A.1, A.2, A.3 and Proposition B.1). We have also provided complete proofs to them except Theorems A.1 and A.2 (which are not our original contribution), and do believe that they are all correct.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.

- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

3. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: Section 5 and Appendix C provide all the information necessary for reproducibility. We provide the URL for downloading the datasets in Appendix C, along with detailed information for tabular datasets (see Table 1) and UTKFace dataset (see Table 2). Figure 6 shows the specific structure of the neural network for tabular datasets, and the backbone architectures for UTKFace are detailed in the text. Other settings, including the number of epochs, learning rate (see Tables 3 and 4), learning rate scheduling strategy, regularization coefficients, batch size, etc. are also clearly listed in Appendix C. Loss functions are defined in Section 3 and evaluation metrics are defined in Section 5. The link functions, thresholding strategies, and fixed-threshold ranges are also detailed in Tables 3 and 4. **Finally, the supplementary material contains the complete source code, enabling exact reproduction of all the experiments.**

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

4. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: Our study used publicly available datasets. For the tabular datasets, the URL for downloading is shown in Appendix C. The UTKFace dataset is also publicly available. We have included the complete experimental code for the tabular datasets with logit and probit models in the supplementary material, along with the necessary instructions and environment setup for reproduction.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

5. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: Section 5 and Appendix C provide detailed experimental settings.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

6. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: For the tabular datasets, publicly available datasets following 30-holdout were used; every experiment was conducted on all the training–validation hold-out splits and the results were averaged, while the metric curves included the corresponding error bands. For UTKFace, each configuration was repeated with three different random seeds, and the results were averaged with error bands shown in the metric curves.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

7. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: Appendix C lists the computational resources for each experiment, including the GPU type, the GPU memory and runtime of each run, and other relevant details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

8. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [\[Yes\]](#)

Justification: We did not do any experiments involving human subjects. We have confirmed the copyright information of all the datasets used in our experiments, and mentioned it in Appendix C. This paper is of theoretical nature, so that there should be no societal impact.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

9. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[NA\]](#)

Justification: This paper is of theoretical nature, so that there should be no societal impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

10. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper is of theoretical nature, so that there should be no such risks of misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

11. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have properly credited the creators of all the datasets used in our experiments in Appendix C. All the license information has been explicitly mentioned in Appendix C as well.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

12. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: The code used in the experiments is uploaded in the supplementary material with run scripts, the license, and other related files.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

13. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: We did not do any experiments involving human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

14. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: We did not do any experiments involving human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

15. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core methodology and experimental pipeline of this study do not employ any LLMs. LLMs were used exclusively for manuscript editing, AI search, and routine debugging assistance during code development; hence, they do not constitute an important, original, or non-standard component of the research.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.