Learning Ordinal–Hierarchical Constraints for Deep Learning Classifiers

Riccardo Rosati, Luca Romeo, Víctor Manuel Vargas, *Member, IEEE*,
Pedro Antonio Gutiérrez, *Senior Member, IEEE*, Emanuele Frontoni, *Senior Member, IEEE*,
and César Hervás-Martínez, *Senior Member, IEEE*

***Abstract*— Real-world classification problems may disclose different hierarchical levels where the categories are displayed in an ordinal structure. However, no specific deep learning (DL) models simultaneously learn hierarchical and ordinal constraints while improving generalization performance. To fill this gap, we propose the introduction of two novel ordinal–hierarchical DL methodologies, namely, the hierarchical cumulative link model (HCLM) and hierarchical–ordinal binary decomposition (HOBD), which are able to model the ordinal structure within different hierarchical levels of the labels. In particular, we decompose the hierarchical–ordinal problem into local and global graph paths that may encode an ordinal constraint for each hierarchical level. Thus, we frame this problem as simultaneously minimizing global and local losses. Furthermore, the ordinal constraints are set by two approaches [ordinal binary decomposition (OBD) and cumulative link model (CLM)] within each global and local function. The effectiveness of the proposed approach is measured on four real-use case datasets concerning industrial, biomedical, computer vision, and financial domains. The extracted results demonstrate a statistically significant improvement to state-of-the-art nominal, ordinal, and hierarchical approaches.**

***Index Terms*— Cumulative link model (CLM), deep learning (DL), hierarchical learning, ordinal binary decomposition (OBD), ordinal regression.**

Nomenclature

X	Input space ($\mathbb{R}$).
M	Number of training points.
H	Number of hierarchical levels of the task.
Q	Number of classes of the general problem.
Y	Set of classes of the general problem.
Y_L^h	Subset of local classes of the hth level.
Y_L	Set of all the local classes.
$\lvert Y_L^h \rvert$	Number of local classes of the hth hierarchical level.
Y_G	Subset of global classes.
$\lvert Y_G \rvert$	Number of global classes.
Y_T	Set of global and local classes.
y_i^h	ith class label of hth hierarchical level.
children(y_i^h)	Set of child classes of y_i^h.
$\mathscr{L}$	Set of all the possible loss functions.
L_L^h	Local loss related to local classes associated with the hth level.
L_G	Global loss related to global classes.
L_T	Total loss.

I. Introduction

IN MACHINE learning (ML), a classification task, is a supervised learning approach that consists in categorizing a set of data into predefined classes. Most ML and deep learning (DL) algorithms deal with classification problems by considering the target variable as a set of disjointed classes, not exploiting the potential structural properties of the data. Several significant real-world classification problems are naturally modeled as a hierarchical structure, where the target to be predicted follows a specific class hierarchy. Notable applications of hierarchical ML approaches include but are not limited to text categorization [1], text translation [2], natural language processing (NLP) [3], and bioinformatics [4].

Accordingly, many problems disclose an ordinal relation, i.e., the categories follow a specific order. However, in contrast with standard regression problems, the distance between them is not quantifiable a priori. For that reason, ordinal classification (also called ordinal regression) assumes great importance in several applications ranging from medical research [5], [6],

Manuscript received 17 May 2022; revised 5 April 2023 and 28 August 2023; accepted 28 January 2024. This work was supported within the research agreement between Università Politecnica delle Marche and Benelli Armi Spa for the "4USER Project" (User and Product Development: from Virtual Experience to Model Regeneration) funded on the POR MARCHE FESR 2014-2020-ASSE 1-OS 1-ACTION 1.1-INT. 1.1.1. Promotion of industrial research and experimental development in the areas of smart specialization-LINEA 2-Bando 2019, approved with DDPF 293 of 22/11/2019. This work has also been partially subsidized by "Agencia Española de Investigación (España)" (grant reference: PID2020-115454GB-C22/AEI/10.13039/501100011033). Víctor Manuel Vargas's research has been subsidized by the Formación de Profesorado Universitario (FPU) Predoctoral Program of the Spanish Ministry of Science, Innovation and Universities (MCIU), grant reference FPU18/00358. *(Corresponding author: Riccardo Rosati.)*

Riccardo Rosati is with the Department of Information Engineering, Università Politecnica delle Marche, 60131 Ancona, Italy (e-mail: r.rosati@pm.univpm.it).

Luca Romeo is with the Department of Economics and Law, Università degli Studi di Macerata, 62100 Macerata, Italy, and also with the Computational Statistics and Machine Learning Unit, Fondazione Istituto Italiano di Tecnologia Genova, 16163 Genoa, Italy.

Víctor Manuel Vargas, Pedro Antonio Gutiérrez, and César Hervás-Martínez are with the Department of Computer Science and Numerical Analysis, University of Córdoba, 14071 Córdoba, Spain.

Emanuele Frontoni is with the Department of Political Science, Communication and International Relations, Università degli Studi di Macerata, 62100 Macerata, Italy, and also with the VRAI Laboratory, Department of Information Engineering, Università Politecnica delle Marche, 60131 Ancona, Italy.

This article has supplementary downloadable material available at https://doi.org/10.1109/TNNLS.2024.3360641, provided by the authors.

Digital Object Identifier 10.1109/TNNLS.2024.3360641

[7], [8], [9] to computer vision [10], risk analysis [11], and industrial applications [12].

Moreover, a hierarchical–ordinal problem could disclose different hierarchical levels where the categories of each level are displayed in an ordinal structure. Here, hierarchical levels refer to different layers or tiers of categorization. Each level contains classes that are more refined or specific than the previous level: it follows that the tiers are arranged in a way that represents a gradual increase in detail. For instance, one might be interested in classifying the quality of a material based on various criteria. Instead of having a single quality measure, it can be useful to categorize the quality at different levels of granularity or accuracy. Specifically, the top tier might encompass general categories such as “high quality,” “medium quality,” and “low quality.” These categories provide a broad evaluation of the material’s quality. Progressing to a lower level, in order to signify more specific quality distinctions, each general quality category may contain subcategories. Namely, within “high quality,” there might exist labels like “Excellent,” “very good,” and “good.” These subcategories offer a finer assessment of quality within each overarching category while displaying ordinal relationship specific to the corresponding level. However, simultaneously managing both the hierarchical and ordinal nature of a given task proves to be a nontrivial problem.

In this context, state-of-the-art ordinal DL strategies can be used to exploit the ordinal information and penalize misclassification errors when the predicted class is farther from the correct one, which is indeed the most relevant kind of error. However, they do not consider the multiple hierarchies of ordinal labels and how the weight of misclassification cost could be different also according to the different hierarchical levels. Hence, learning hierarchical and ordinal constraints using a single model remains a real challenge in the ML/DL literature. This article proposes simultaneously learning hierarchical–ordinal constraints using a DL methodology. The proposed approach is designed according to a novel hierarchical formulation that models local and global losses, where local losses act as auxiliary losses to strengthen the hierarchical–ordinal dependencies. To integrate ordinal relation within global and local losses, we propose to include a cumulative link model (CLM) [13] combined with the quadratic weighted kappa (QWK) loss [14] and an ordinal binary decomposition (OBD) approach [5] with the mean absolute error (MAE) loss. Our approaches represent an extension of ordinal methodologies for modeling hierarchical constraints. Indeed, the main novelty lies in decomposing hierarchical–ordinal problems into global- and local-ordinal tasks that the integration of CLM and QWK or OBD approaches can potentially solve.

Our main contribution can be summarized as follows.

1) We propose two novel hierarchical DL ordinal methodologies, namely, hierarchical CLM (HCLM) and hierarchical OBD (HOBD), which can model the ordinal structure of different hierarchical levels of the labels. In particular, we address the hierarchical problem by decomposing it into local- and global-ordinal tasks, which may potentially share common patterns that can be employed to improve the final classification performance.
2) We test the effectiveness and generalizability of our methodologies on four real-use case datasets related to the industrial [aesthetic quality control (AQC)], biomedical [vaccine priority administration (VPA)], computer vision [age estimation (AE)], and financial risk [corporate credit rating (CCR)] domains, all of them disclosing a natural hierarchical–ordinal structure of the classes. We also measure the performance with respect to state-of-the-art ordinal and hierarchical DL methodologies, by demonstrating a statistically significant gain of our proposed methodologies both for image and tabular data sources.

This article is organized as follows. In Section II, we review ordinal and hierarchical classification methodologies. We describe the proposed methodology in Section III, while in Section IV, we discuss how the approach is evaluated on the AQC, VPA, AE, and CCR datasets. Experimental procedure is reported in Section V, while the results are presented in Section VI. Finally, in Section VII, we provide conclusions of our findings, limitations, and future work.

II. Related Work

This section reviews the two lines of work based on which our proposal has been devised. The former is the work on ordinal classification methodologies (see Section II-A). The latter focuses on hierarchical classification approaches (see Section II-B). Our approaches provide an extension of the body of knowledge to solve a different task that combines characteristics of both lines of research, namely, hierarchical–ordinal problems. Existing ordinal frameworks were extended using local losses in a multiple hierarchies framework. Indeed, the multiple hierarchies of ordinal labels were included in HCLM and HOBD. This extension ensures different weights of misclassification cost for each hierarchical level.

A. Ordinal Classification

Ordinal classification problems are classification tasks where classes follow a natural order. Existing ML and DL approaches in the literature attempt to model ordinal constraints in different ways [15]. The most naive methods range from performing a simple regression using the class labels and rounding the values on the prediction phase to using a cost-sensitive (e.g., cost matrix) approach to evaluate multiclass classification models [16]. However, regression learners may depend on the values used for representing the labels, while the cost matrix can lead to different label representations, thus leading to different ambiguous solutions. Instead, our approach can directly model the ordering of labels by learning a unique representation of ordinal constraints.

Recent state-of-the-art works move toward two approaches: OBD and threshold models combining ordinal loss functions. The OBD method decomposes the ordinal problem into multiple binary subproblems. OBD approaches can be divided into multiple models and multioutput single models. The multiple models (e.g., [17]) solve the ordinal classification problem by independently employing each binary classification model and combining the predicted variables into the final label. One of the significant drawbacks of multiple model approaches lies in the required number of binary classifiers that depends on the cardinality of the ordinal label, and the training effort increases when the task discloses a more complex structure. The multiple-output single model (e.g., [18]) leverages the property of a neural network to handle multiple labels in a single stage. However, all OBD methods are faced with the challenge of combining the results of all decompositions into a single final classification output. Error correcting output codes (ECOCs) are suited for this task, as this approach considers all the outputs equally in the final decision. Barbero-Gómez et al. [5] proposed a unified convolutional neural network (CNN) architecture with a global optimization criterion. Although based on the approach proposed in [5], our HOBD extends the global optimization criterion by including different local and global hierarchical losses, which effectively map hierarchical–ordinal constraints in a nominal classification framework. As we will see in the experiments, our HOBD method performs favorably over state-of-the-art OBD competitors (i.e., [5]).

On the other hand, threshold models represent a widely used approach for ordinal tasks. These models assume the presence of an underlying latent continuous variable, from which distinct ranks emerge through the adjustment of specific thresholds. Thus, in this framework, the value of the latent variable and the thresholds must be learned from the data. Some approaches, such as the classical proportional odds model, fall into the CLM framework, a probabilistic family of models for predicting probabilities of groups of contiguous categories, considering the ordinal scale. An extension of the CLM model to the DL scenario was proposed in [13] by using the deep neural network's output as the value of the latent variable, which in turn determines the rank. Unlike OBD, threshold models provide a single mapping vector composed of several thresholds equal to the number of classes. However, threshold models and CLM do not lead to computing the cumulative probability at different scales, which can be associated with different hierarchical levels. Our approach can overcome this drawback by estimating the cumulative probability for each hierarchical level. Also, the performed experiments demonstrate the effectiveness of our HCLM approach to the state-of-the-art threshold model competitor [13].

Ranking-based hashing framework [19] was also proposed in the literature to learn hash functions with deep neural networks by exploring the ordinal structure of feature and label dimensions. Differently from binary quantization-based hashing methods, the ordinal structure is not represented a priori but instead learned during the training stage of a deep neural network. Also, it is worth noting that the ranking-based hashing approach differs from ordinal classification. This is due to its nature as a learning-to-rank strategy, which typically requires a more extensive training process. This is primarily because the approach involves generating intermediate representations, known as hash codes.

Finally, numerous of these approaches involve ordinal loss functions [14], [20]. These functions incorporate the ordinal nature of the labels into the error function, aiming to penalize significant misclassification errors and encourage unimodality in the probability distribution generated by the model.

B. Hierarchical Classification

In the ML literature, hierarchical classification problems are usually addressed using a hierarchical multilabel classification (HMC) approach. In HMC, every instance may belong to multiple classes simultaneously. These classes are arranged in a hierarchical structure, implying that an example associated with a particular subclass belongs simultaneously to all the superclasses in the hierarchy. The state-of-the-art approaches include traditional ML models [21] and neural networks architectures [22], [23], [24], in which the hierarchical structure is explored in different ways. The most naive approach (global classifier) [22] is to employ a single classifier for modeling the entire class hierarchy, where the objective is to predict the classes associated with the leaves of the hierarchy without considering upper levels. However, a significant limitation of this approach is that it completely ignores the class relationships and any hierarchical constraints while typically predicting only the leaf nodes (flat classification). A different approach (local classifiers) [25], [26], [27] is to employ a set of classifiers for each node or each parent level. Thus, each classifier is specialized in solving the classification task associated with the child nodes. However, unlike our approach, where a single model is used to learn the overall problem, local classifiers require a more significant computation effort for learning separated multiple models [28], especially in situations where the complexity of the task increases. This evidence is also confirmed in our experiment as our HOBD and HCLM methods perform better than the state-of-the-art global and local alternatives [22], [25], [26], [27]. Another significant difference from all the above-cited works is how the proposed approach models the labels of each hierarchical level. Although state-of-the-art works employ nominal classification models, we consider the possibility of learning ordinal relationships among labels.

More related to our work is the HMC approach in [29], which also leverages a single hierarchical multilabel neural network architecture (HMCN) capable of simultaneously optimizing their local and global loss functions to model the hierarchical structure of the classification task while penalizing hierarchical violations. The benefit of this strategy is to decompose local and global classes that may potentially represent labels of different nature/hierarchy. Unlike their definition of global classes (which include all the classes in the hierarchy), our global loss only considers the leaf nodes of our hierarchical problem, i.e., the classes associated with the final classification problem we want to solve. Their formulation addresses two primary concerns. First, the HMC approach proposed in [29] necessitates a postprocessing step for guaranteeing that all predictions adhere to the hierarchical path. This postprocessing penalizes predictions that violate the hierarchy during the training phase. This requirement is not

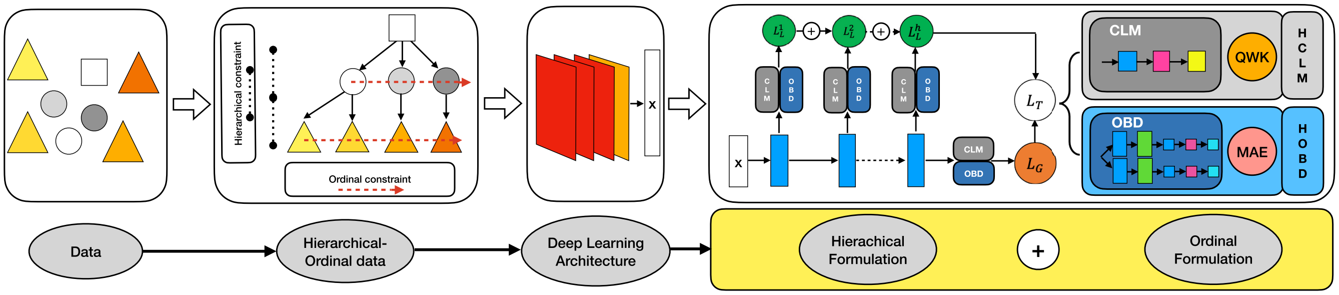

Fig. 1. Overview of the proposed method. The figure describes the general formulation by which the hierarchical–ordinal problem is decomposed and modulated via HOBD and HCLM alternatively. HCLM: hierarchical CLM. HOBD: hierarchical OBD. CLM: cumulative link model. OBD: ordinal binary decomposition. QWK: quadratic weighted kappa loss. MAE: mean absolute error. L_G: global loss. L_L^h: local loss. L_T: total loss. x: input feature vector. Red: conv2D layer. Blue: dense layer. Green: dropout. Pink: BN. Orange: pooling. Yellow: CLM layer. Light blue: sigmoid activation.

necessary for our approach because our method naturally models only the admissible paths (i.e., it is impossible to obtain inconsistent global predictions' labels). At the same time, our approach ensures the modeling of local paths by minimizing each local loss associated with the corresponding hierarchical level and providing related local predictions. Moreover, our formulation allows incorporating ordinal constraints within global and local losses, assuming that global losses' ordinal dependencies can differ from local losses. In this way, the natural setting of this task is considered. Furthermore, the method models different ordinal relationships between various local losses associated with different hierarchical levels. For instance, a discrete ordinal rating scale can be different for each hierarchy level.

III. Methods

This section sets the notation we used to formulate our approach (see Section III-A). Afterward, we describe the formulation of the proposed method in Sections III-B and III-C and the prediction phase in Section III-D.

The overall methodology is described in Fig. 1. The hierarchical–ordinal problem is represented using a graph structure. Our hierarchical problem is decomposed by local and global classes to learn consistently different classes in the hierarchy. Note that the order of nodes within a level in Fig. 1 reflects the natural ordinal structure of the classes. As can be checked in the last level of the VPA hierarchy, this natural order in the labels does not necessarily correspond with the order defined by the corresponding superclasses. Accordingly, the ordinal constraints are integrated using CLM and OBD approaches. Thus, our framework leads to the design of two methodologies called HCLM and HOBD for solving generic and real-world hierarchical–ordinal problems.

A. Notation

The adopted notations are described in the Nomenclature.

Fig. 2 shows an example of general hierarchical constraint settings regarding graph structure. In our problem definition, the labels of each child node associated with different parent nodes can be completely different, reflecting other structures (i.e., $h | \bigcap_{i=1}^{|Y_L^h|} \text{children}(y_i^h) = \emptyset$).

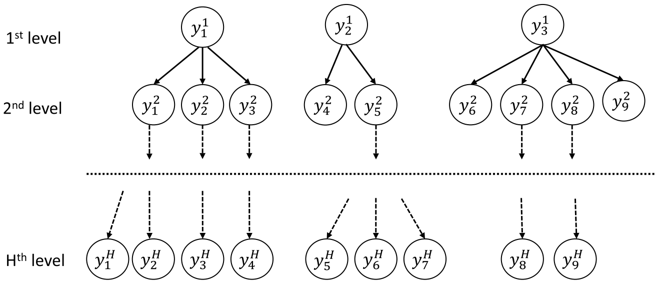

Fig. 2. Example of the definitions proposed for global and local classes when describing the hierarchical task. In this example, the global classes consist on the set $Y_G = \{y_1^H, y_2^H, y_3^H, y_4^H, y_5^H, y_6^H, y_7^H, y_8^H, y_9^H, y_4^2, y_6^2, y_9^2, \ldots\}$, while the local classes consist of $Y_L^1 = \{y_1^1, y_2^1, y_3^1\}$, $Y_L^2 = \{y_1^2, y_2^2, y_3^2, y_4^2, y_5^2, y_6^2, y_7^2, y_8^2, y_9^2\}, \ldots,$ and the relationships between levels are described by $\text{children}(y_1^1) = \{y_1^2, y_2^2, y_3^2\}$, $\text{children}(y_2^1) = \{y_4^2, y_5^2\}$, $\text{children}(y_3^1) = \{y_6^2, y_7^2, y_8^2, y_9^2\}$, $\text{children}(y_4^2) = \emptyset$, $\text{children}(y_6^2) = \emptyset$, $\text{children}(y_9^2) = \emptyset, \ldots.$

Definition 1: The local classes of the hth level of the hierarchy (Y_L^h) are defined as follows:

$$Y_L^h = \bigcup_{i=1}^{|Y_L^h|} y_i^h \tag{1}$$

where $h \in \{1, 2, \ldots, H-1\}$. Note that the local classes include all the nodes not in the last hierarchical level (see Fig. 2).

Definition 2: The global classes (Y_G) are defined as follows:

$$Y_G = \bigcup_{h=1}^{H} \left[\bigcup_{i|\text{children}(y_i^h)=\emptyset} y_i^h \right] \tag{2}$$

where children(·) represents the set of child classes of a given node, and those nodes fulfilling $\text{children}(y_i^h) = \emptyset$ correspond to leaves of the graph. Indeed, global classes consist of classes with no descendants (see Fig. 2), thus reflecting the original categories of the classification problem without considering the parent nodes.

This definition of local and global classes allows dealing with classification problems that can also be arranged on different hierarchical levels. For the example described in Fig. 2, the global classes consist on the set $Y_G = \{y_1^H, y_2^H, y_3^H, y_4^H, y_5^H, y_6^H, y_7^H, y_8^H, y_9^H, y_4^2, y_6^2, y_9^2, \ldots\}$, while the local classes set is composed by $Y_L^1 = \{y_1^1, y_2^1, y_3^1\}$, $Y_L^2 = \{y_1^2, y_2^2, y_3^2, y_4^2, y_5^2, y_6^2, y_7^2, y_8^2, y_9^2\}, \ldots$

Definition 3: The definition of local and global classes leads to the following:

$$Y_T = Y_L \cup Y_G \tag{3}$$

where

$$Y_L = \bigcup_{h=1}^{H-1} Y_L^h. \tag{4}$$

Note that those classes that are leaves but not placed on the Hth level (last level) will simultaneously be global and local classes. The reason is that they are part of the original global classification task but should also be considered to represent the ordinal structure of the corresponding level.

B. Hierarchical Formulation

In our formulation, we combine local and global losses as follows:

$$L_T = \beta \frac{\sum_{h=1}^{H-1} L_L^h}{H-1} + (1-\beta) L_G \tag{5}$$

where $\beta \in [0, 1]$ is the hyperparameter that regulates the tradeoff regarding local and global information. L_L^h refers to the local loss computed according to the hth level

$$L_L^h \in \mathscr{L}\left(X, Y_L^h\right) \tag{6}$$

where $\mathscr{L}(X, Y_L^h)$ is the set of loss functions computed using the input data, X, and the classes of the hth hierarchical level, Y_L^h.

On the other hand, L_G refers to the global loss computed:

$$L_G \in \mathscr{L}(X, Y_G) \tag{7}$$

where $\mathscr{L}(X, Y_G)$ considers, in this case, the set of global classes.

In our formulation, we aggregate and minimize the local losses (i.e., specific models responsible for the prediction of local classes) and the global loss (i.e., global model responsible for the prediction of global classes). The rationale behind this choice lies in maximizing a consistent global prediction at the final hierarchical level and obtaining a consistent prediction for each local node. Local losses model the local-ordinal information for each hierarchical level while the global loss function keeps track of the label dependency in the hierarchy as a whole by also taking into account the global-ordinal information for all leaf nodes. The detailed formulation is described in Section III-C.

C. Hierarchical–Ordinal Formulation

In this section, we present the proposed hierarchical–ordinal methodologies, namely, HCLM (see Section III-C1) and HOBD (see Section III-C2) as an extension of state-of-the-art CLM and OBD approaches.

1) Hierarchical CLM: The following conditions hold.

a) Background on cumulative link model: CLM [30] is a set of threshold models that are suited to posterior model probabilities of a given label $y \in Y = \{y_1, y_2, \ldots, y_Q\}$, taking into account the ordinal relation defined by the problem. The input data $\mathbf{x} \in X$ are projected into a 1-D space denoted as $f(\mathbf{x}) \in \mathbb{R}$ and divided into Q intervals by $Q-1$ thresholds, where Q is the number of classes of the general problem. According to [13], these thresholds are learned during the training process from the following equation:

$$t_q = t_1 + \sum_{i=2}^{q-1} \gamma_i^2, \quad q = 1, \ldots, Q-1 \tag{8}$$

where t_1 determines the first threshold (which splits the first and the second classes) and $\boldsymbol{\gamma} = \{\gamma_2, \ldots, \gamma_i, \ldots, \gamma_{Q-2}\}$ parameters are used to obtain the rest of the thresholds based on the first one. All of them are learned along with the model weights through gradient descent optimization. This formulation ensures that the constraints $t_1 \leq t_2 \leq \cdots \leq t_{Q-1}$ are fulfilled and, assuming $t_0 = -\infty$ and $t_Q = +\infty$, all the output space is covered. The cumulative probability obtained by the CLM for any given class y_q and input data $\mathbf{x}$ is calculated as follows:

$$h^{-1}\left(P\left(y \preceq y_q|\mathbf{x}\right)\right) = t_q - f(\mathbf{x}), \quad q = 1, 2, \ldots, Q-1 \tag{9}$$

where $f(\cdot)$ is a mapping function of the input vector learned during the training phase and $h^{-1}(\cdot)$ is a monotonic function that is known as link function. In this case, the logit function, defined by the equation, has been implemented as a link function

$$h^{-1}(p) = \operatorname{logit}(p) = \log \frac{p}{1-p} \tag{10}$$

where $p = P(y \preceq y_q|\mathbf{x})$. Finally, cumulative probabilities can be used to obtain standard posterior probabilities by simply subtracting them

$$P(y = y_1|\mathbf{x}) = P(y \preceq y_1|\mathbf{x}) \tag{11}$$

$$P\left(y = y_q|\mathbf{x}\right) = P\left(y \preceq y_{q+1}|\mathbf{x}\right) - P\left(y \preceq y_q|\mathbf{x}\right) \tag{12}$$

where $q \in \{2, \ldots, Q-1\}$ and, by definition, $P(y \preceq y_Q|\mathbf{x}) = 1$.

b) Our hierarchical CLM approach: Notice how in our methodology, we integrated the CLM for modeling both the local and global losses [see (5)]; thus, the candidate classes Y become Y_G and Y_L, and the number of classes Q becomes $|Y_G|$ and $|Y_L^h|$, for the global and local losses, respectively.

The employed elementary loss function is the QWK loss [14], a continuous version of the weighted Kappa metric, which is suitable for ordinal problems as it weights the errors differently depending on the distance from the correct class.

The QWK local loss computed for the hth level is defined as follows:

$$L_{\mathrm{L}}^h = \frac{\sum_{m=1}^{M} \sum_{y_i \in Y_L^h} \omega_{i,y_m} P(y = y_i|\mathbf{x}_m)}{\sum_{i=1}^{|Y_L^h|} \frac{M_i}{M} \sum_{y_j \in Y_L^h} \left(\omega_{i,j} \sum_{m=1}^{M} P\left(y = y_j|\mathbf{x}_m\right)\right)} \tag{13}$$

where $\mathbf{x}_m$ is the input data of the mth sample, M is the total number of training samples, M_i is the number of training samples of the ith local class, $P(y = y_i|\mathbf{x}_m)$ is the model posterior probability that the mth sample belongs to local class y_i, and $\omega_{i,j}$ are the elements of the penalization matrix (for the quadratic version, $\omega_{i,j} = ((i-j)^2/(|Y_L^h|-1)^2)$).

Similarly, the QWK global loss is defined as follows:

$$L_G = \frac{\sum_{m=1}^{M}\sum_{y_i\in Y_G}\omega_{i,y_i}P(y=y_i|\mathbf{x}_m)}{\sum_{i=1}^{|Y_G|}\frac{M_i}{M}\sum_{y_j\in Y_G}\left(\omega_{i,j}\sum_{m=1}^{M}P\big(y=y_j|\mathbf{x}_m\big)\right)} \quad (14)$$

where $P(y=y_i|\mathbf{x}_m)$ is the model conditional probability that the mth sample belongs to global class y_i.

2) Hierarchical OBD:

a) Background on ordinal binary decomposition: OBD is an ordinal approach based on decomposing the classification task of Q classes of the original problem into a set of $Q-1$ binary problems. In particular, each problem q consists on verifying if $y \succ y_q$ conditioned to $\mathbf{x}$ where $1 \le q < Q$ [5]. The individual probability of belonging to a specific class $P(y = y_q)$ is computed as a function of the cumulative probabilities $P(y \succ y_q)$ as follows:

$$\begin{aligned} P(y=y_1) &= 1 - P(y \succ y_1)\\ P\big(y=y_q\big) &= P\big(y \succ y_{q-1}\big) - P\big(y \succ y_q\big), \quad \forall 1 \le q < Q\\ P\big(y=y_Q\big) &= P\big(y \succ y_{Q-1}\big). \end{aligned}$$

The output code for each class is defined as the coordinates of a vertex of a hypercube in $Q-1$ dimensions. To decide the class associated with a sample $\mathbf{x}$, the class with the nearest code according to some distance is considered.

b) Our hierarchical OBD approach: In our proposed HOBD, we employed the MAE as distance and thus as elementary loss. For the local losses, the MAE computed for the hth hierarchical level is defined as follows:

$$L_L^h = \frac{1}{|Y_L^h|-1}\sum_{y_i\in Y_L^h}|1\{y \succ y_i\} - P(y \succ y_i|\mathbf{x})| \quad (15)$$

where $\mathbf{x}$ is the input data matrix, $P(y \succ y_i|\mathbf{x})$ is the model posterior cumulative probability for local class y_i, and $1\{y \succ y_i\}$ is the corresponding target vector.

The MAE global loss is defined as follows:

$$L_G = \frac{1}{|Y_G|-1}\sum_{y_i\in Y_G}|1\{y \succ y_i\} - P(y \succ y_i|\mathbf{x})| \quad (16)$$

where $P(y \succ y_i|\mathbf{x})$ is the model posterior cumulative probability for global class y_i and $1\{y \succ y_i\}$ is the corresponding target vector.

D. Prediction

According to the proposed formulation, prediction can be obtained for local or global classes. Predicted classes are obtained differently depending on the methodology (HCLM or HOBD). As CLM provides us with posterior probabilities, the predicted classes of HCLM are obtained by

$$\hat{y}_L^h = \operatorname*{argmax}_{y_i\in Y_L^h} P(y=y_i|\mathbf{x}) \quad (17)$$

$$\hat{y}_G = \operatorname*{argmax}_{y_i\in Y_G} P(y=y_i|\mathbf{x}). \quad (18)$$

In the case of HOBD, we compute the distance between the model posterior probability vector and the ground truth of global and local labels. The predicted label is the one that has the minimum distance and can be computed for the global and local classes as follows:

$$\hat{y}_L^h = \operatorname*{argmin}_{y\in Y_L^h} ||\mathbf{p}_L^h - \mathbf{t}_L^h(y)|| \quad (19)$$

$$\hat{y}_G = \operatorname*{argmin}_{y\in Y_G} ||\mathbf{p}_G - \mathbf{t}_G(y)|| \quad (20)$$

where $\mathbf{p}_L^h = (P(y \succ y_i|\mathbf{x}) : y_i \in Y_L^h)$ and $\mathbf{p}_G = (P(y \succ y_i|\mathbf{x}) : y_i \in Y_G)$ are two vectors containing all posterior cumulative probabilities (associated with the independent binary subproblems of OBD) for global and local classes, respectively, and $\mathbf{t}_L^h(y) = (1\{y \succ y_i\} : y_i \in Y_L^h)$ and $\mathbf{t}_G(y) = (1\{y \succ y_i\} : y_i \in Y_G)$ are the corresponding target vectors for global and local classes, respectively.

Note that, in our experiments, we have evaluated only the prediction for global classes. This choice is based on the fact that these classes are the most important ones in terms of cost for the considered real problems, and the prediction can be directly compared with the results obtained with a nonhierarchical approach. However, local predictions could also be useful for other practical contexts. Therefore, we evidenced how the proposed method can provide local prediction by considering the AQC dataset.

IV. Materials

We based the evaluation of the proposed approach on four real-use case datasets, including industrial (see Section IV-A), biomedical (see Section IV-B), computer vision (see Section IV-C), and financial risk (see Section IV-D) domains. These four datasets were selected based on the hierarchical and ordinal nature they offer for evaluating our approach. All employed datasets disclose a different hierarchical structure originating from a real-world task. Indeed, the target to be predicted follows a specific class hierarchy for each task (see Fig. 3). In particular, the AQC and VPA datasets disclose two hierarchical levels, while the AE and CCR datasets include three hierarchical levels. In all datasets, categories for each hierarchical level are arranged on an ordinal scale.

A. AQC Dataset

The evaluation of the quality of a manufactured product (quality control task) is usually taken by the expert operator that arranges the quality classes in a hierarchy from most general to most specific shapes. The quality control task is done by considering, for instance, the quality of the material by inspecting different factors such as aesthetic defects, grains, and other specific details. Thus, a different hierarchical level in this context may correspond to a particular aspect that should be considered for the overall quality process. Furthermore, as in the case of the AQC task, the quality classes assume a natural ordering where usually the first category represents the worst quality, while the last class indicates the maximum quality. Thus, misclassifying a pattern in the furthest quality classes may be more penalized than classifying it in adjacent classes.

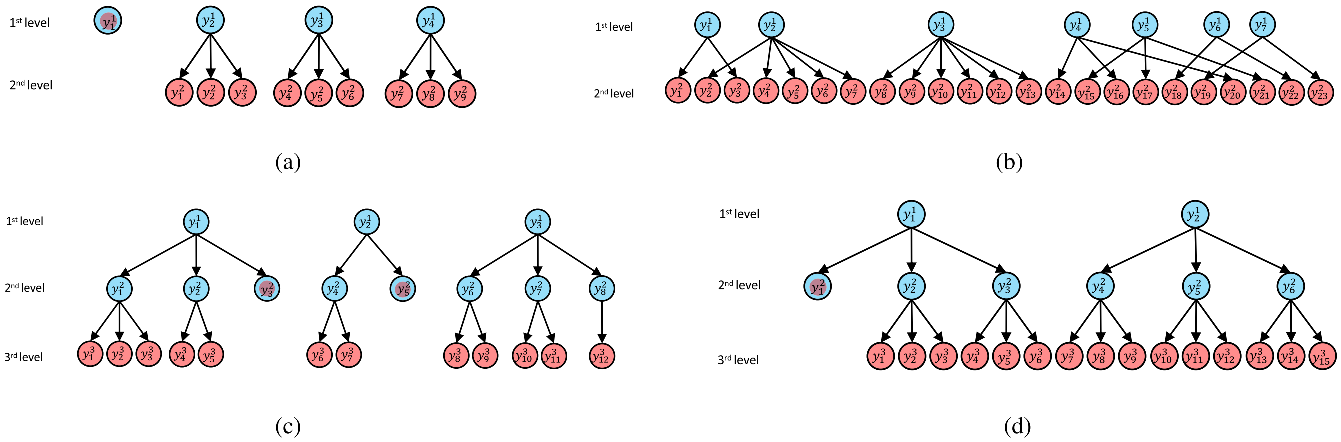

Fig. 3. Structure of the classes for the proposed datasets. In blue and red, the local and global classes are shown, respectively. The classes exploited in both global and local losses are indicated with both colors. (a) AQC dataset. (b) VPA dataset. (c) AE dataset. (d) CCR dataset.

The collected dataset [31] comprises 2120 color images of 1060 wooden stocks, which undergo an AQC phase before being assembled on sport rifles. This process consists of assigning a specific grade to each item according to the wood grain and its aesthetic properties, ensuring that each type of rifle model complies with the production requirements by assembling a specific grade class. The categories are grouped into four macro classes (1–4) and each of these, except class 1, is divided into three micro labels ($-$, c, and $+$). The proposed problem contains ten aggregated global classes with an ordinal relation: 1, 2^-, 2^c, 2^+, 3^-, 3^c, 3^+, 4^-, 4^c, and 4^+. Notice how the problem is structured on two hierarchical levels: the first class has not been divided into micro labels because the company usually produces model series with higher quality classes. Table I in the supplementary material shows the distribution of the global classes. It is worth noting that a different ordinal structure is present in the two hierarchical levels. For the AQC task, the local classes consist on $Y_L^1 = \{y_1^1, y_2^1, y_3^1, y_4^1\} = \{1, 2, 3, 4\}$, while the global classes consist on the set $Y_G = \{y_1^1, y_1^2, y_2^2, y_3^2, y_4^2, y_5^2, y_6^2, y_7^2, y_8^2, y_9^2\} = \{1, 2^-, 2^c, 2^+, 3^-, 3^c, 3^+, 4^-, 4^c, 4^+\}$ [see Fig. 3(a)].

B. VPA Dataset

Clinical decision support is increasingly in demand to extract relevant information from electronic health records (EHRs) to support territorial clinical medicine and general practitioners (GPs). The FIMMG_COVID dataset [32] was derived from the FIMMG EHR database, which is integrated into the FIMMG Netmedica cloud architecture. It consists of EHR data belonging to 17 147 patients collected from 11 different GPs to estimate the COVID-19 VPA in the population. The classes are represented by the priority classes (PCs) defined by the GPs according to a unified priority selection criteria (i.e., age, dysautonomia, chronic pathologies, and obesity), which considers the age-related comorbidities to have a more critical weight than the age itself. The *anagraphic*, *monitoring*, and *pathologies* tables were selected according to the GP's suggestions as to the most potential discriminative (predictors) for classifying the severity of chronic disease and thus the PCs, resulting in a total of 27 independent variables. In particular, the candidate features are sex, age, weight, height, waist circumference, systolic blood pressure, diastolic blood pressure, and one-hot encoding of the 20 most frequent International Classification of Disease (ICD-9) categories. It is worth noting that these features do not reflect any information about the severity of the disease. Also, this dataset shows a natural hierarchical order where the first hierarchical level represents different age ranges, while the second level represents PCs, as reported in Table I in the supplementary material. We excluded one of the original PCs (i.e., highly vulnerable patients), which is not directly influenced by age. In this case, the global PC classes assume an ordinal structure that is different from the local age-range classes and does not only depend on age. Moreover, the natural order in the global labels does not necessarily correspond with the order defined by the corresponding superclasses [see Fig. 3(b)]. Our formulation allows taking into account this natural hierarchical–ordinal setting. Hence, for the VPA task, the local classes consist on $Y_L^1 = \{y_1^1, y_2^1, y_3^1, y_4^1, y_5^1, y_6^1, y_7^1\} = \{\text{Over90, 80–89, 70–79, 60–69, 50–59, 30–49, 16–29}\}$, while the global classes consist on the set $Y_G = \{y_1^2, y_2^2, y_3^2, y_4^2, y_5^2, y_6^2, y_7^2, y_8^2, y_9^2, y_{10}^2, y_{11}^2, y_{12}^2, y_{13}^2, y_{14}^2, y_{15}^2, y_{16}^2, y_{17}^2, y_{18}^2, y_{19}^2, y_{20}^2, y_{21}^2, y_{21}^2, y_{23}^2\} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23\}$ [see Fig. 3(b)].

C. AE Dataset

The UTKFace dataset contains over 20 000 facial images that include individuals of all ages, ranging from 0 to 116 years old. It is widely used as a benchmark dataset for various tasks, such as face detection and AE [33]. These images capture various poses, facial expressions, lighting conditions, obstructions, resolutions, and other variables. The dataset presents an ordinal nature by default but can be decomposed into different hierarchical levels based on age ranges. Global classes were defined by balancing age ranges and the number of images for each class, as reported in Table II in the supplementary material. According to our formulation, the overall problem was decomposed into three different hierarchical levels. Local classes of the first level consist of $Y_L^1 = \{y_1^1, y_2^1, y_3^1\} = \{0\text{–}26, 27\text{–}44, 45+\}$, while for the second level, $Y_L^2 = \{y_1^2, y_2^2, y_3^2, y_4^2, y_5^2, y_6^2, y_7^2, y_8^2\} = \{0\text{–}11, 12\text{–}22, 23\text{–}26, 27\text{–}37, 38\text{–}44, 45\text{–}64, 65\text{–}79, 80+\}$. The global classes consist of the set $Y_G = \{y_1^3, y_2^3, y_3^3, y_4^3, y_5^3, y_3^2, y_6^3, y_7^3, y_5^2, y_8^3, y_9^3, y_{10}^3, y_{11}^3, y_{12}^3\} = \{0\text{–}1, 2\text{–}5, 6\text{–}11, 12\text{–}18, 19\text{–}22, 23\text{–}26, 27\text{–}29, 30\text{–}37, 38\text{–}44, 45\text{–}54, 55\text{–}64, 65\text{–}72, 73\text{–}79, 80+\}$ [see Fig. 3(c)].

D. CCR Dataset

Credit ratings are evaluations of company credit worthiness provided by specialized agencies. These ratings are essential financial indicators for potential investors, as they help to understand better the risk associated with a company's investment returns. Most credit rating agencies use a unique rating scale with a discrete ordinal structure. The employed dataset was collected by the S&P rating agency, which uses a grading scale with 23 different levels ranging from AAA, the most favorable rating, to D, the riskiest [34]. However, due to the few samples present for classes C and D (261 in total), we discarded CCC^{+}, CCC^{c}, CCC^{-}, CC^{+}, CC^{c}, C^{c}, and D classes, resulting in a final 16-class grading task, as shown in Table III in the supplementary material. In addition, considering the highly unbalanced distribution of classes, an undersampling strategy directly proportional to class frequencies was adopted based on training samples for each seed. Concerning our $Y_L^1 = \{y_1^1, y_2^1\} = \{\mathrm{A}^*, \mathrm{B}^*\}$, where $A^* = \{AAA, AA, A\}$ and $B^* = \{BBB, BB, B\}$, while for the second level, $Y_L^2 = \{y_1^2, y_2^2, y_3^2, y_4^2, y_5^2, y_6^2\} = \{\mathrm{AAA, AA, A, BBB, BB, B}\}$. The global classes consist on the set $Y_G = \{y_1^2, y_1^3, y_2^3, y_3^3, y_4^3, y_5^3, y_3^2, y_6^3, y_7^3, y_5^2, y_8^3, y_9^3, y_{10}^3, y_{11}^3, y_{12}^3, y_{13}^3, y_{14}^3, y_{15}^3\} = \{$AAA, AA^{+},AA^{c}, AA^{-}, A^{+}, A^{c}, A^{-}, BBB^{+}, BBB^{c}, BBB^{-}, BB^{+}, BB^{c}, BB^{-}, B^{+}, B^{c}, B$^{-}\}$ [see Fig. 3(d)].

V. Experimental Procedure

In this section, we describe the experimental procedure, starting from the employed architectures (see Section V-A) and also including the description of the performed experimental comparisons (see Section V-B) and the evaluation metrics (see Section V-C).

A. Architectures

The structure of the predictive model does not require any particular assumption; thus, different model structures are equally acceptable.

The hierarchical constraint is modulated on the networks' top fully connected (FC) layers, by forming a multioutput classification head to achieve local- and global-ordinal optimizations. This classification head comprises $H - 1$ local outputs and one global output. The main flow includes H FC layers with ReLU activation to which local submodules are connected. Its own FC layer characterizes each local submodule before the local output. Our approach ensures that each local submodule learns the ordinal constraint from a given hierarchical level. In our architecture, batch normalization (BN) [35] was inserted to accelerate the networks' convergence and improve the training stability.

As regards the proposed HCLM approach, each local output and the global one present only one neuron, which provides the model projection in a 1-D space. This value is used to classify the sample into the corresponding class according to the CLM with the logit link function. Accordingly, for the proposed HOBD approach, each local submodule FC layer is decomposed into a set of $|Y_L^h| - 1$ FC blocks (with the same dimension). Each block consists of an FC layer, a leaky ReLU activation function, and a dropout layer. Each final output layer with a sigmoid activation function solves an individual binary classification subproblem. The global output also assumes the same decomposition, presenting $|Y_G| - 1$ binary outputs.

Considering the different types of data related to the classification tasks we aim to solve, we employed CNNs as feature extractors to solve the AQC and AE tasks on rifle stocks images and facial images, while we adopted standard multilayer perceptron (MLP) architectures for solving the VPA and CCR tasks associated with tabular data.

For the AQC classification, the VGG16 model [36] was used as architecture, maintaining the pretrained ImageNet weights for the convolutional part of the model according to a transfer learning approach. The choice of VGG16 architecture as a baseline model for the AQC task is related to a previous work [31], where VGG16 achieved the best results among other state-of-the-art classification models in standard nominal classification. A dropout regularization layer was inserted in the first FC layer of the classification head. The rate of dropout and the size of all dense layers are selected within the hyperparameters optimization procedure. On the other hand, the choice of ResNet50 for the AE dataset relies on additional performed tests, which highlights the superiority of this model to other standard architectures (VGG-16, EfficientNet, MobileNet, and so on). In this way, using different networks allows us to evaluate whether the approach can be generalized to different architectures.

In the VPA task, *monitoring* and *pathologies* information is annotated irregularly over time, leading to sparse observations. Therefore, standard MLP architecture achieved competitive state-of-the-art results to deal with this setting by modeling spatial relationship [37]. Since the CCR dataset originated from tabular data with no temporal information encoded, the MLP was also employed for solving this task. The MLP networks consist of two hidden layers with a ReLU activation function, each followed by a dropout layer. Also, in this case, the size of hidden and top FC layers and dropout rate are chosen within hyperparameters selection. Adam [38] is adopted as an optimizer, and the best learning rate, batch size, and dropout rate were selected as hyperparameters. All hyperparameters were tuned in the separate validation set using a grid-search approach (see Table I).

HCLM and HOBD methodologies were evaluated following a stratified (over rifles for AQC dataset, over GP for VPA dataset, and over companies for CCR dataset) holdout procedure: the dataset was split by maintaining 80% of the whole set for the training phase and the remaining 20% for the test set. From the training set, another 15% of the samples were taken for the validation set. For the AQC dataset, experiments with an on-the-fly data augmentation strategy were performed

TABLE I

MODEL HYPERPARAMETERS EXPLORED FOR EACH BACKBONE ARCHITECTURE RELATED TO AQC, VPA, AE, AND CCR DATASETS. ALL HYPERPARAMETERS WERE TUNED IN THE SEPARATE VALIDATION SET USING A GRID-SEARCH APPROACH. FC: FULLY CONNECTED LAYERS. LR: LEARNING RATE

Backbone architecture	FC neurons	Dropout rate	Batch size	LR	Epochs
VGG16 (AQC task)	{2048, 4096}	{0.1, 0.3, 0.5}	{8, 16, 32, 64}	$\{10^{-4}, 10^{-3}, 10^{-2}\}$	50
MLP (VPA task)	{256, 512}	{0.1, 0.3, 0.5}	{16, 32, 64, 128}	$\{10^{-4}, 10^{-3}, 10^{-2}\}$	50
ResNet50 (AE task)	{1024, 2048}	{0.1, 0.3, 0.5}	{8, 16, 32, 64}	$\{10^{-4}, 10^{-3}, 10^{-2}\}$	50
MLP (CCR task)	{256, 512}	{0.3, 0.5, 0.8}	{16, 32, 64, 128}	$\{10^{-4}, 10^{-3}, 10^{-2}\}$	200

for balancing global classes, randomly applying a horizontal flip to the training samples. Moreover, we also evaluated the contribution of the local- and global-ordinal losses exploring the following values of β: $\{0.1, 0.2, 0.5, 0.8, 0.9\}$. The early stopping strategy was adopted for all the experiments with a patience value of 15 epochs monitoring validation loss. To achieve robust results from a statistical perspective, all the experiments were performed 30 times using different seeds to create the data partitions and initialize the model parameters. All the experiments were run using TensorFlow 2.0 and Keras 2.3.1 frameworks on an Intel Core i7-4790 CPU 3.60 GHz with 16 GB of RAM and NVIDIA GeForce GTX 970. All the experiments are reproducible and the Python code used in the experiments will be available in a public repository.[1]

B. Experimental Comparisons

Unlike other state-of-the-art work, our methodologies are conceived for learning ordinal and hierarchical dependencies. For that reason, we decided to compare the proposed HCLM and HOBD with respect to other hierarchical and ordinal formulations widely employed in the ML literature (see Section II). The state-of-the-art comparisons include the following.

1) *Global (GLB) Approach [22].* This approach maps a hierarchical problem into a standard classification problem that fully embeds the parent-level information. The nominal global approach (GLB-NOM) ignores the class hierarchy predicting only leaf node classes as a standard multiclass classification [28] with categorical cross-entropy (CCE) loss. The ordinal variant of this approach was implemented by integrating ordinal relationship through OBD [5] (GLB-OBD) and CLM [39] (GLB-CLM).
2) *Local Classifier per Parent Node (LCPN) [25], [26], [27] Approaches:* Models are trained for solving each local task (i.e., a separate model for each parent node) using nominal (LCPN-NOM) or ordinal classifiers. The ordinal relationship was encoded with an OBD (LCPN-OBD) and a CLM (LCPN-CLM) layer by considering, respectively, MAE and QWK as loss functions. Binary subproblems were treated with a sigmoid activation function on the output neuron and binary cross-entropy (BCE) loss.
3) For the AQC task, we performed additional experiments decomposing the hierarchical constraints in two different tasks using the multitask learning (MTL) approach [40]. This strategy is viable when the global label can be handled as a combination of labels from two distinct tasks (i.e., macro and micro tasks as reported in Table I in the supplementary material) for the AQC dataset). This approach is not viable for the VPA, AE, and CCR tasks, as the global classes are not fully decomposable. In the MTL-CLM formulation, we computed the QWK macro and micro losses related to the macro $\{1, 2, 3, 4\}$ and micro $\{+, c, -\}$ classes, respectively. We also extended this comparison (MTL-CLM$_{\text{loc}}$) to include a hierarchical constraint by minimizing the micro loss locally for each macro class. Since class 1 has no child labels, a further postprocessing step was necessary to put this hierarchical constraint. It is worth noting that this additional step is not required in our approach. In MTL-OBD and MTL-OBD$_{\text{loc}}$, the OBD approach is applied to both decomposable tasks and MAE losses are minimized.
4) *HMCN [29]:* This method uses a multilabel binary encoding strategy to minimize the BCE loss for each hierarchical level. In this case, the output is a binary class vector (expected output) containing all classes in the hierarchy. In contrast with our approach, in this case, the model may lead to the prediction of nonadmissible paths, thus requiring a further postprocessing stage (i.e., violation constraint) to avoid inconsistent global classes.

C. Evaluation Metrics

Although the proposed HCLM and HOBD approach can be used for predicting local and global classes, we measured the performance with respect to state-of-the-art methodologies in terms of the prediction obtained for the global classes. The main objective for all the considered datasets is to provide a reliable global class prediction rather than a good performance for local classes. Moreover, the global results are the only ones that can be compared directly with those of nonhierarchical state-of-the-art approaches. However, for the sake of completion, we further extracted the local class performance of the proposed approach for the AQC datasets.

Considering the ordinal nature of global and local classes, ordinal metrics were chosen for evaluating our hierarchical–ordinal problem as they properly reflect the deviation of a misclassification error from the actual class.

The following metrics were considered.

1) QWK, to be maximized [14].
2) MAE, to be minimized, which reflects the average absolute deviation of the predicted class from the true class (i.e., average absolute deviation in the number of categories of the ordinal scale) [41].
3) 1-off accuracy, to be maximized, indicates the predicted class as correct when it is off at most by one adjacent class from the ground-truth one.

[1] Python code used in the experiments will be made available on the GitHub repository.

4) Accuracy, to be maximized, included as the standard nominal metric.

VI. Results

In this section, we report the results of the proposed HOBD and HCLM approach for AQC (see Section VI-A), VPA (see Section VI-B), AE (see Section VI-C), and CCR (see Section VI-D) datasets. Afterward, we provide the statistical test criteria for evaluating the results and the performed statistical analysis (see Section VI-E) to highlight the possible significant performance improvement of the proposed approach to state-of-the-art models.

A. Aesthtetic Quality Control Dataset

Table II compares the proposed approaches to GLB, LCPN, MTL, and HMCN competitors for the AQC dataset. With QWK = 0.921 (0.009) and MAE = 0.635 (0.048), the HOBD approach outperforms all the baseline algorithms. Moreover, the averaged ranking in terms of QWK [R_{QWK} = 2.100 (1.213)] and MAE [R_{MAE} = 2.000 (1.203)] for HOBD are lower than those reported by all state-of-the-art competitors. The HCLM discloses lower performance than HOBD and comparable results to state-of-the-art methods. The proposed HOBD approach is consistent with this task's imbalanced setting. Indeed, adopting the data augmentation procedure reflects no improvement in QWK and an improvement of 0.6% in terms of MAE.

B. VPA Dataset

Table II compares the proposed approaches to GLB, LCPN, and HMCN competitors for the VPA dataset. With QWK = 0.942 (0.013) and MAE = 0.960 (0.091), the HOBD approach outperforms all the baseline algorithms. Moreover, the averaged ranking in terms of QWK [R_{QWK} = 2.133 (1.525)] and MAE [R_{MAE} = 1.433 (0.626)] for HOBD are lower than those reported by all state-of-the-art competitors. Similar to the AQC dataset, the HCLM discloses lower performance than HOBD and comparable results to state-of-the-art methods.

C. AE Dataset

Table II displays how the proposed method compares to its competitors, GLB, LCPN, and HMCN, for the AE dataset. In the QWK metric, the proposed approaches HOBD [QWK = 0.889 (0.004)] and HCLM [QWK = 0.889 (0.007)] outperform all the state-of-the-art algorithms. This is confirmed by the QWK averaged ranking where these methods achieve the best results, in particular with R_{QWK} = 1.708 (0.999) for HCLM. At the same time, MAE results are comparable between HOBD [MAE = 0.978 (0.019)] and standard GLB-OBD [MAE = 0.978 (0.020)], but the MAE averaged ranking for HOBD [R_{MAE} = 1.458 (0.509)] is lower than those of all competitors. However, the extracted results demonstrated how the proposed method is more stable in terms of R_{MAE} and R_{QWK} for solving the specific task.

D. CCR Dataset

Table II displays how the proposed method compares to its competitors, GLB, LCPN, and HMCN, for the CCR dataset. With MAE = 2.175 (0.129), the HOBD method outperforms all the other approaches. Moreover, the averaged ranking in terms of MAE [R_{MAE} = 1.467 (0.507)] for HOBD is lower than the state-of-the-art competitors. For what concerns QWK, the performance between HOBD [MAE = 0.611 (0.037)] and GLB-CLM [MAE = 0.610 (0.035)] is comparable. The QWK averaged ranking for HOBD [R_{QWK} = 2.400 (1.221)] is higher than GLB-CLM [R_{QWK} = 2.167 (1.555)]. The higher performance of GLB-CLM in terms of QWK can be justified by the distribution of rating classes (see Table III in the supplementary material). A greater number of samples in the middle classes may negatively influence the GLB-CLM that systematically predicts always those central classes, thus providing an increase of QWK, which, however, does not correlate with a reduction of MAE and an increase of accuracy. Differently, our HOBD leads to achieving the best tradeoff regarding Acc, 1-off, MAE, and QWK.

E. Statistical Analysis

We evaluated the statistical significance of the best-performing approach compared to other competitors. First, we performed an Anderson–Darling test [42] to test that the values of the metrics follow a normal distribution considering $\alpha = 0.05$. In this way, the QWK ($p = 0.697$) and MAE ($p = 0.894$) scores for the proposed HOBD approach were found to follow a normal distribution for the AQC dataset. Accordingly, the QWK ($p = 0.266$) and MAE ($p = 0.231$) scores for the proposed HOBD approach were also found to follow a normal distribution according to the Anderson–Darling test for the VPA dataset. Hence, a paired-sample one-sided t-test ($\alpha = 0.05$) was performed to compare the QWK and MAE of the best-performing HOBD with respect to the best-performing state-of-the-art methodologies.

For the AQC dataset, the QWK scores were found to be significantly ($\alpha = 0.01$) higher for HOBD than all GLB, LCPN, MTL, and HMCN models. Accordingly, we also found the MAE scores to be significantly lower ($\alpha = 0.01$) for HOBD than all GLB, LCPN, MTL, and HMCN models.

Moreover, for the VPA dataset, the QWK scores were found to be significantly higher for HOBD than GLB-NOM, GLB-CLM, LCPN, and HMCN, with $\alpha = 0.01$, and GLB-OBD with $\alpha = 0.05$. Accordingly, we also found the MAE scores significantly lower ($\alpha = 0.01$) for HOBD than GLB, LCPN, and HMCN.

The results of the AE dataset indicate that HOBD outperformed GLB-NOM, GLB-CLM, LCPN, and HMCN in terms of QWK scores, with statistical significance ($\alpha = 0.01$). We also found that HOBD had significantly ($\alpha = 0.01$) lower MAE scores than GLB-NOM, GLB-CLM, LCPN, and HMCN.

For the CCR dataset, the QWK scores were found to be significantly higher for HOBD than GLB-NOM, LCPN, and HMCN with $\alpha = 0.01$ and GLB-OBD with $\alpha = 0.05$. Accordingly, we also found the MAE scores to be significantly ($\alpha = 0.01$) lower for HOBD than GLB-NOM, GLB-CLM, LCPN, and HMCN.

TABLE II

PERFORMANCE EVALUATION ON AQC, VPA, AE, AND CCR DATASETS. EXPERIMENTAL RESULTS: AVERAGE OVER 30 EXECUTIONS EXPRESSED WITH MEAN (STD). THE PERFORMANCE OF EACH MODEL IS MEASURED ON THE TEST SET. THE BEST RESULTS ARE IN BOLD. STARS INDICATE WHETHER THE BEST-PERFORMING ALGORITHM IS SIGNIFICANTLY BETTER THAN STATE-OF-THE-ART APPROACHES (**: $\alpha = 0.05$). DA: DATA AUGMENTATION. ACT: ACTIVATION FUNCTION. CCE: CATEGORICAL CROSS ENTROPY. BCE: BINARY CROSS ENTROPY. R_{QWK}: AVERAGED RANK FOR QWK METRIC. R_{MAE}: AVERAGED RANK FOR THE MAE METRIC

Dataset	Model	DA	Act	Loss	β	Acc↑	1-off↑	QWK↑	MAE↓	R_{QWK} ↓	R_{MAE} ↓
AQC	GLB-NOM [22]	yes	Softmax	CCE		0.504 (0.038)	0.803 (0.029)	0.870 (0.047) **	0.809 (0.119)**	16.400 (6.015)	21.200 (4.374)
	GLB-NOM [22]	no	Softmax	CCE		0.528 (0.033)	0.817 (0.023)	0.882 (0.022) **	0.756 (0.069)**	11.933 (6.017)	18.567 (5.456)
	GLB-OBD [5]	yes	Sigmoid	MAE		0.410 (0.024)	0.783 (0.023)	0.791 (0.041) **	0.976 (0.074)**	25.167 (0.531)	25.233 (0.430)
	GLB-OBD [5]	no	Sigmoid	MAE		0.413 (0.023)	0.772 (0.024)	0.768 (0.048) **	1.017 (0.079)**	25.333 (1.295)	25.667 (0.606)
	GLB-CLM [39]	yes	Logit	QWK		0.426 (0.028)	0.834 (0.022)	0.902 (0.011) **	0.790 (0.049)**	16.500 (4.424)	10.600 (5.090)
	GLB-CLM [39]	no	Logit	QWK		0.432 (0.030)	0.835 (0.020)	0.904 (0.011) **	0.785 (0.051)**	15.833 (5.240)	9.467 (4.688)
	LCPN-NOM [25], [26], [27]	yes	Softmax	CCE		0.529 (0.029)	0.835 (0.021)	0.898 (0.011) **	0.709 (0.052)**	7.233 (4.408)	13.600 (5.934)
	LCPN-NOM [25], [26], [27]	no	Softmax	CCE		**0.538** (0.029)	0.840 (0.020)	0.903 (0.010) **	0.688 (0.048)**	5.167 (3.343)	9.367 (5.034)
	LCPN-OBD	yes	Sigmoid	MAE		0.529 (0.032)	0.845 (0.026)	0.908 (0.016) **	0.684 (0.066) **	5.533 (3.803)	7.633 (6.278)
	LCPN-OBD	no	Sigmoid	MAE		0.524 (0.041)	0.847 (0.024)	0.906 (0.015) **	0.690 (0.070) **	6.500 (5.316)	8.667 (6.557)
	LCPN-CLM	yes	Logit	QWK		0.446 (0.045)	0.837 (0.030)	0.896 (0.019) **	0.782 (0.082) **	15.100 (6.970)	13.567 (6.334)
	LCPN-CLM	no	Logit	QWK		0.446 (0.044)	0.826 (0.030)	0.891 (0.017) **	0.802 (0.081) **	16.867 (5.835)	16.133 (5.551)
	MTL-OBD [40]	yes	Sigmoid	MAE	0.8	0.475 (0.023)	0.821 (0.023)	0.897 (0.012) **	0.781 (0.048) **	15.333 (4.147)	13.633 (4.165)
	MTL-OBD [40]	no	Sigmoid	MAE	0.8	0.490 (0.023)	0.828 (0.026)	0.902 (0.014) **	0.755 (0.054) **	11.500 (4.224)	10.567 (5.793)
	MTL-CLM [40]	yes	Logit	QWK	0.5	0.460 (0.025)	0.820 (0.022)	0.892 (0.012) **	0.801 (0.052) **	17.733 (4.968)	16.800 (4.781)
	MTL-CLM [40]	no	Logit	QWK	0.5	0.469 (0.023)	0.817 (0.023)	0.893 (0.012) **	0.794 (0.047) **	16.233 (4.688)	15.800 (4.649)
	MTL-OBD$_{loc}$	yes	Sigmoid	MAE	0.8	0.474 (0.026)	0.819 (0.026)	0.896 (0.014) **	0.786 (0.058)**	15.633 (4.295)	14.233 (3.812)
	MTL-OBD$_{loc}$	no	Sigmoid	MAE	0.8	0.491 (0.025)	0.823 (0.022)	0.901 (0.013) **	0.757 (0.053) **	11.600 (3.616)	10.067 (4.009)
	MTL-CLM$_{loc}$	yes	Logit	QWK	0.5	0.456 (0.030)	0.808 (0.023)	0.888 (0.014) **	0.820 (0.051) **	19.700 (3.395)	18.533 (3.560)
	MTL-CLM$_{loc}$	no	Logit	QWK	0.5	0.463 (0.031)	0.811 (0.024)	0.890 (0.016) **	0.803 (0.062) **	17.467 (5.342)	17.500 (4.762)
	HMCN [29]	yes	Sigmoid	BCE	0.5	0.524 (0.034)	0.823 (0.022)	0.890 (0.016) **	0.737 (0.058) **	10.633 (5.720)	16.233 (6.388)
	HMCN [29]	no	Sigmoid	BCE	0.5	0.532 (0.030)	0.824 (0.023)	0.887 (0.020) **	0.733 (0.0691) **	9.500 (5.722)	16.667 (6.194)
	HOBD	yes	Sigmoid	MAE	0.5	0.534 (0.028)	**0.868** (0.200)	**0.921** (0.009)	**0.635** (0.048)	**2.100** (1.213)	**2.000** (1.203)
	HOBD	no	Sigmoid	MAE	0.5	0.536 (0.028)	0.864 (0.018)	0.921 (0.008)	0.639 (0.044)	2.533 (1.676)	2.200 (1.448)
	HCLM	yes	Logit	QWK	0.5	0.426 (0.036)	0.833 (0.029)	0.905 (0.011)	0.795 (0.065)	16.367 (6.049)	8.267 (4.017)
	HCLM	no	Logit	QWK	0.5	0.433 (0.032)	0.834 (0.021)	0.905 (0.011)	0.783 (0.055)	14.667 (5.274)	8.700 (5.167)
VPA	GLB-NOM [22]		Softmax	CCE		**0.688** (0.020)	0.752 (0.023)	0.928 (0.016)**	1.067 (0.108) **	5.667 (1.709)	3.500 (1.042)
	GLB-OBD [5]		Sigmoid	MAE		0.654 (0.020)	0.765 (0.019)	0.939 (0.016)*	0.998 (0.102)**	2.733 (1.617)	2.267 (0.944)
	GLB-CLM [39]		Logit	QWK		0.458 (0.029)	0.685 (0.021)	0.936 (0.014)**	1.288 (0.078) **	3.667 (1.749)	7.900 (0.960)
	LCPN-NOM [25], [26], [27]		Softmax	CCE		0.659 (0.020)	0.719 (0.021)	0.921 (0.021)**	1.179 (0.109)**	7.533 (1.358)	6.200 (1.243)
	LCPN-OBD		Sigmoid	MAE		0.657 (0.027)	0.723 (0.026)	0.922 (0.018)**	1.166 (0.132) **	6.567 (1.501)	5.467 (2.013)
	LCPN-CLM		Logit	QWK		0.611 (0.087)	0.677 (0.056)	0.921 (0.014)**	1.268 (0.114) **	7.600 (2.027)	7.267 (1.874)
	HMCN [29]		Sigmoid	BCE	0.5	0.685 (0.023)	0.753 (0.025)	0.929 (0.019)**	1.061 (0.126)**	5.533 (1.925)	3.667 (1.583)
	HOBD		Sigmoid	MAE	0.8	0.684 (0.020)	**0.766** (0.019)	**0.942** (0.013)	**0.960** (0.091)	**2.133** (1.525)	**1.433** (0.626)
	HCLM		Logit	QWK	0.5	0.458 (0.059)	0.689 (0.017)	0.939 (0.013)	1.265 (0.071)	2.967 (1.426)	7.267 (0.907)
AE	GLB-NOM [22]		Softmax	CCE		0.369 (0.007)	0.663 (0.012)	0.821 (0.012)	1.211 (0.032)	8.333 (0.917)	6.833 (1.090)
	GLB-OBD [5]		Sigmoid	MAE		0.382 (0.011)	**0.755** (0.010)	0.888 (0.005)	**0.978** (0.020)	2.250 (0.794)	1.625 (0.647)
	GLB-CLM [39]		Logit	QWK		0.288 (0.083)	0.669 (0.006)	0.862 (0.003)	1.207 (0.014)	5.167 (0.565)	6.750 (0.944)
	LCPN-NOM [25], [26], [27]		Softmax	CCE		0.324 (0.013)	0.687 (0.014)	0.833 (0.009)	1.212 (0.037)	7.083 (0.0.776)	6.792 (1.141)
	LCPN-OBD		Sigmoid	MAE		0.390 (0.008)	0.710 (0.008)	0.868 (0.007)	1.049 (0.024)	4.250 (0.608)	3.917 (0.584)
	LCPN-CLM		Logit	QWK		0.299 (0.010)	0.672 (0.009)	0.824 (0.007)	1.268 (0.029)	8.042 (0.751)	8.667 (0.637)
	HMCN [29]		Sigmoid	BCE	0.5	0.353 (0.034)	0.680 (0.037)	0.844 (0.032)	1.171 (0.013)	6.083 (1.472)	5.625 (1.861)
	HOBD		Sigmoid	MAE	0.2	**0.387** (0.011)	0.750 (0.010)	**0.889** (0.004)	**0.978** (0.019)	2.083 (0.654)	**1.458** (0.509)
	HCLM		Logit	QWK	0.1	0.341 (0.017)	0.728 (0.018)	**0.889** (0.007)	1.036 (0.045)	**1.708** (0.999)	3.333 (0.868)
CCR	GLB-NOM [22]		Softmax	CCE		**0.153** (0.020)	0.403 (0.029)	0.567 (0.043)	2.425 (0.136)	5.533 (1.456)	4.367 (1.217)
	GLB-OBD [5]		Sigmoid	MAE		0.151 (0.014)	0.433 (0.024)	0.593 (0.045)	2.183 (0.132)	3.433 (1.278)	1.567 (0.568)
	GLB-CLM [39]		Logit	QWK		0.140 (0.020)	0.383 (0.032)	0.610 (0.035)	2.513 (0.198)	**2.167** (1.555)	5.000 (1.554)
	LCPN-NOM [25], [26], [27]		Softmax	CCE		0.093 (0.013)	0.243 (0.029)	0.054 (0.027)	3.822 (0.305)	8.900 (0.305)	8.267 (0.450)
	LCPN-OBD		Sigmoid	MAE		0.144 (0.013)	0.392 (0.027)	0.559 (0.041)	2.450 (0.146)	6.100 (1.296)	4.700 (1.264)
	LCPN-CLM		Logit	QWK		0.073 (0.012)	0.246 (0.027)	0.097 (0.034)	4.092 (0.311)	8.100 (0.305)	8.733 (0.450)
	HMCN [29]		Sigmoid	BCE	0.5	0.147 (0.017)	0.392 (0.029)	0.573 (0.044)	2.439 (0.160)	5.100 (1.213)	4.700 (1.418)
	HOBD		Sigmoid	MAE	0.1	**0.153** (0.016)	**0.435** (0.029)	**0.611** (0.037)	**2.175** (0.129)	2.400 (1.221)	**1.467** (0.507)
	HCLM		Logit	QWK	0.2	0.135 (0.013)	0.375 (0.025)	0.600 (0.038)	2.584 (0.155)	3.167 (1.663)	6.167 (1.020)

F. Local Performance Analysis

For the sake of completion, the method's performance when considering the local classes has been evaluated on the AQC dataset, and the metrics' results can be checked in Table III. As expected, when comparing these results with those in Table II, the overall performance is significantly improved due to reducing the number of classes with respect to the global setting. Depending on the real scenario, the performance of

TABLE III

PERFORMANCE EVALUATION ON LOCAL CLASSES (Y_L^1) OF OUR BEST PERFORMING METHOD (HOBD, WITH DATA AUGMENTATION AND $\beta = 0.5$) ON THE AQC DATASET

Y_L^1	Acc↑	1-off↑	QWK↑	MAE↓
{1, 2, 3, 4 }	0.830 (0.027)	0.996 (0.005)	0.891 (0.022)	0.175 (0.030)

TABLE IV

PERFORMANCE EVALUATION ON CCR. ABLATION ANALYSIS WAS PERFORMED BY REMOVING THE MID-HIERARCHICAL LEVEL ON THE CCR DATASET

Model	Levels	Acc↑	1-off↑	QWK↑	MAE↓
GLB-NOM	/	0.153 (0.020)	0.403 (0.029)	0.567 (0.043)	2.425 (0.136)
GLB-OBD	/	0.151 (0.014)	0.433 (0.024)	0.593 (0.045)	2.183 (0.132)
GLB-CLM	/	0.140 (0.020)	0.383 (0.032)	0.610 (0.035)	2.513 (0.198)
HOBD	2	**0.156** (0.012)	0.431 (0.024)	0.597 (0.036)	2.180 (0.112)
HOBD	3	0.153 (0.016)	**0.435** (0.029)	**0.611** (0.037)	**2.175** (0.129)

local classes can also be an important factor in selecting the predictor to be implemented.

G. Hierarchical-Level Ablation Analysis

We further analyzed the potential gain of the proposed approach to model multiple hierarchical constraints by performing an ablation analysis. Thus, from the three levels of the hierarchy of the CCR dataset [see Fig. 3(d)], we excluded the second level and compared our best-performing approach's performance to the best state-of-the-art competitors (see Table IV). Also, in this case, as expected, we found a drop in the performance related to ordinal metrics when the middle level was removed. This result justifies the effectiveness of the proposed method, which is able to increasingly improve the overall global performance when the information available about the hierarchy constraints is higher (three levels instead of two).

VII. CONCLUSION

This article proposes a DL methodology for learning ordinal–hierarchical constraints. Our formulation breaks down the hierarchical–ordinal problem into local- and global-ordinal tasks that can potentially share common patterns. Differently from state-of-the-art DL approaches that tried to model only the hierarchical or the ordinal dependencies independently, we proposed two compact hierarchical DL strategies based on OBD (HOBD) and CLM (HCLM) to embrace hierarchical–ordinal constraints of the classification problem. We tested our approach on four different real-world hierarchical–ordinal datasets. The proposed HOBD proved effective in dealing with these tasks by overcoming the other state-of-the-art nominal, hierarchical, and ordinal approaches. Thus, the proposed approaches are suited for solving other real-world classification tasks with hierarchical and ordinal properties.

A. Limitations and Future Work

Supported by numerous experiments conducted across various datasets, the proposed methodology demonstrates the capacity to automatically encode diverse hierarchical structures. This is feasible as long as comprehensive knowledge about the hierarchy (including local classes belonging to specific hierarchical levels) and the corresponding ordinal constraints at each hierarchical level are available. Although the experiments were performed using two and three hierarchical levels, the value of h can be increased to deal with more complex hierarchies. The proposed methodology is conceived to deal with a fixed number of hierarchical levels specified by the problem. Given that the proposed ordinal–hierarchical approach improved the state-of-the-art methodologies, in future works, this proposal can be extended to deal with tasks where the hierarchical and ordinal label structure is not given a priori but can be inferred from the characteristics of the data. Thus, our approach can be extended via a meta-learning formulation to customize and preserve ordinal and hierarchical task knowledge simultaneously [43]. Moreover, the proposed approach needs to consider the presence of sparse or missing labels. In different application scenarios ranging from recommendation systems to clinical decision support systems, some classes in the hierarchy may be missing or unavailable. For example, in the cross-domain recommendation, providing reliable recommendations to newly joined users (so-called cold-start users) is a challenging task. In this context, unlabeled data are abundant and easily accessible, while the collection of labeled data is a major task. Future work may be handled to generalize the proposed methodology to weakly supervised and semi-supervised settings scenarios, using self-learning [44], context-based learning [45], and incremental learning approaches [46].

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no competing interests.

REFERENCES

[1] R. A. Stein, P. A. Jaques, and J. F. Valiati, "An analysis of hierarchical text classification using word embeddings," *Inf. Sci.*, vol. 471, pp. 216–232, Jan. 2019.

[2] X. Tan, L. Zhang, D. Xiong, and G. Zhou, "Hierarchical modeling of global context for document-level neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 1576–1585.

[3] K. Tran, A. Bisazza, and C. Monz, "The importance of being recurrent for modeling hierarchical structure," 2018, *arXiv:1803.03585*.

[4] C. Wan and A. A. Freitas, "An empirical evaluation of hierarchical feature selection methods for classification in bioinformatics datasets with gene ontology-based features," *Artif. Intell. Rev.*, vol. 50, no. 2, pp. 201–240, Aug. 2018.

[5] J. Barbero-Gómez, P.-A. Gutiérrez, V.-M. Vargas, J.-A. Vallejo-Casas, and C. Hervás-Martínez, "An ordinal CNN approach for the assessment of neurological damage in Parkinson's disease patients," *Exp. Syst. Appl.*, vol. 182, Nov. 2021, Art. no. 115271.

[6] A. M. Durán-Rosal et al., "Ordinal classification of the affectation level of 3D-images in Parkinson diseases," *Sci. Rep.*, vol. 11, no. 1, pp. 1–13, Mar. 2021.

[7] T. Albuquerque, R. Cruz, and J. S. Cardoso, "Ordinal losses for classification of cervical cancer risk," *PeerJ Comput. Sci.*, vol. 7, p. e457, Apr. 2021.

[8] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2002–2011.

[9] H. Zhu et al., "Convolutional ordinal regression forest for image ordinal estimation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 4084–4095, Aug. 2022.

[10] L. Kook, L. Herzog, T. Hothorn, O. Dürr, and B. Sick, "Deep and interpretable regression models for ordinal outcomes," *Pattern Recognit.*, vol. 122, Feb. 2022, Art. no. 108263.

[11] F. Tan, X. Hou, J. Zhang, Z. Wei, and Z. Yan, "A deep learning approach to competing risks representation in peer-to-peer lending," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1565–1574, May 2019.

[12] R. Rosati, L. Romeo, V. M. Vargas, P. A. Gutiérrez, C. Hervás-Martínez, and E. Frontoni, "A novel deep ordinal classification approach for aesthetic quality control classification," *Neural Comput. Appl.*, vol. 34, no. 14, pp. 11625–11639, Jul. 2022.

[13] V. M. Vargas, P. A. Gutiérrez, and C. Hervás, "Deep ordinal classification based on the proportional odds model," in *Proc. Int. Work-Conf. Interplay Between Natural Artif. Comput.* Cham, Switzerland: Springer, 2019, pp. 441–451.

[14] J. de la Torre, D. Puig, and A. Valls, "Weighted Kappa loss function for multi-class classification of ordinal data in deep learning," *Pattern Recognit. Lett.*, vol. 105, pp. 144–154, Apr. 2018.

[15] P. A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernández-Navarro, and C. Hervás-Martínez, "Ordinal regression methods: Survey and experimental study," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 127–146, Jan. 2016.

[16] M. Alali, N. M. Sharef, H. Hamdan, M. A. A. Murad, and N. A. Husin, "Multi-layers convolutional neural network for Twitter sentiment ordinal scale classification," in *Proc. Int. Conf. Soft Comput. Data Mining*. Cham, Switzerland: Springer, 2018, pp. 446–454.

[17] S. Chen, C. Zhang, M. Dong, J. Le, and M. Rao, "Using ranking-CNN for age estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5183–5192.

[18] J.-C. Xie and C.-M. Pun, "Deep and ordinal ensemble learning for human age estimation from facial images," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2361–2374, 2020.

[19] L. Jin, K. Li, Z. Li, F. Xiao, G.-J. Qi, and J. Tang, "Deep semantic-preserving ordinal hashing for cross-modal similarity search," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1429–1440, May 2018.

[20] C. K. Goh, Y. Liu, and A. W. K. Kong, "A constrained deep neural network for ordinal regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 831–839.

[21] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, "Decision trees for hierarchical multi-label classification," *Mach. Learn.*, vol. 73, no. 2, pp. 185–214, Nov. 2008.

[22] L. Masera and E. Blanzieri, "AWX: An integrated approach to hierarchical-multilabel classification," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, 2018, pp. 322–336.

[23] Y. Li et al., "DEEPre: Sequence-based enzyme EC number prediction by deep learning," *Bioinformatics*, vol. 34, no. 5, pp. 760–769, Mar. 2018.

[24] E. Giunchiglia and T. Lukasiewicz, "Coherent hierarchical multi-label classification networks," in *Proc. NIPS*, vol. 33, 2020, pp. 9662–9673.

[25] M. Kulmanov, M. A. Khan, and R. Hoehndorf, "DeepGO: Predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, no. 4, pp. 660–668, Feb. 2018.

[26] C. Xu and X. Geng, "Hierarchical classification based on label distribution learning," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 5533–5540.

[27] R. M. Pereira, Y. M. G. Costa, and C. N. Silla, "Handling imbalance in hierarchical classification problems using local classifiers approaches," *Data Mining Knowl. Discovery*, vol. 35, no. 4, pp. 1564–1621, Jul. 2021.

[28] C. N. Silla Jr. and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Min. Knowl. Discov.*, vol. 22, no. 1, pp. 31–72, 2011.

[29] J. Wehrmann, R. Cerri, and R. Barros, "Hierarchical multi-label classification networks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5075–5084.

[30] A. Agresti, *Analysis of Ordinal Categorical Data*, vol. 656. Hoboken, NJ, USA: Wiley, 2010.

[31] R. Rosati et al., "Bias from the wild Industry 4.0: Are we really classifying the quality or shotgun series?" in *Proc. Int. Conf. Pattern Recognit.* Cham, Switzerland: Springer, 2021, pp. 637–649.

[32] L. Romeo and E. Frontoni, "A unified hierarchical XGBoost model for classifying priorities for COVID-19 vaccination campaign," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108197.

[33] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4352–4360.

[34] P. Hajek, V. Olej, and O. Prochazka, "Predicting corporate credit ratings using content analysis of annual reports—A Naïve Bayesian network approach," in *Enterprise Applications, Markets and Services in the Finance Industry*. Frankfurt, Germany: Springer, 2017, pp. 47–61.

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[37] M. Bernardini, L. Romeo, P. Misericordia, and E. Frontoni, "Discovering the type 2 diabetes in electronic health records using the sparse balanced support vector machine," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 1, pp. 235–246, Jan. 2020.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[39] V. M. Vargas, P. A. Gutiérrez, and C. Hervás-Martínez, "Cumulative link models for deep ordinal classification," *Neurocomputing*, vol. 401, pp. 48–58, Aug. 2020.

[40] V. Sanh, T. Wolf, and S. Ruder, "A hierarchical multi-task approach for learning embeddings from semantic tasks," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 6949–6956.

[41] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, and P. A. Gutiérrez, "Metrics to guide a multi-objective evolutionary algorithm for ordinal classification," *Neurocomputing*, vol. 135, pp. 21–31, Jul. 2014.

[42] F. Scholz and M. Stephens, "K-sample Anderson–Darling tests," *J. Amer. Stat. Assoc.*, vol. 82, no. 399, pp. 918–924, Sep. 1987.

[43] H. Yao, Y. Wei, J. Huang, and Z. Li, "Hierarchically structured meta-learning," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds. Jun. 2019, pp. 7045–7054.

[44] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S^{4}L: Self-supervised semi-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1476–1485.

[45] Z. Li, Y. Sun, L. Zhang, and J. Tang, "CTNet: Context-based tandem network for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9904–9917, Dec. 2022.

[46] H. Chen, Y. Jia, J. Ge, and B. Gu, "Incremental learning algorithm for large-scale semi-supervised ordinal regression," *Neural Netw.*, vol. 149, pp. 124–136, May 2022.

Riccardo Rosati was born in Cagli, Italy. He received the Ph.D. degree in information engineering, computer science curriculum, from Università Politecnica delle Marche, Ancona, Italy, in 2023.

He is currently a Post-Doctoral Research Fellow and an Adjunct Professor at the Department of Information Engineering (DII), Università Politecnica delle Marche. He is also a member of the VRAI Laboratory, Università Politecnica delle Marche. His research interests include machine learning, deep learning, and extended reality systems applied to the biomedical field and Industry 4.0 landscape.

Luca Romeo received the Ph.D. degree in computer science from the Department of Information Engineering, Università Politecnica delle Marche, Ancona, Italy, in 2018. His Ph.D. thesis was on "applied machine learning for human motion analysis and affective computing."

He is currently a Tenure-Track Assistant Professor in computer science with the Department of Economics and Law, University of Macerata, Macerata, Italy. He is also with the Computational Statistics and Machine Learning Unit, Fondazione Istituto Italiano di Tecnologia Genova, Genoa, Italy. His research interests include the design of novel machine learning algorithms for solving relevant challenges in different real-world domains.

Víctor Manuel Vargas (Member, IEEE) was born in Córdoba, Spain. He received the B.Sc. degree in computer engineering from the University of Córdoba, Córdoba, Spain, in 2018, the M.Eng. degree in artificial intelligence research from Menéndez Pelayo International University, Madrid, Spain, in 2019, and the Ph.D. degree in artificial intelligence from the University of Córdoba in 2023.

He is currently an Interim Assistant Professor and a member of the AYRNA Research Group, Department of Computer Science and Numerical Analysis, University of Córdoba. His current research interests include a wide range of topics, including deep learning, ordinal classification, time series classification, and applications related to renewable energies, biomedicine, and industry.

Pedro Antonio Gutiérrez (Senior Member, IEEE) received the B.S. degree in computer science from the University of Seville, Seville, Spain, in 2006, and the Ph.D. degree in computer science and artificial intelligence from the University of Granada, Granada, Spain, in 2009.

He is currently a Professor with the Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain. His research interests are in the areas of supervised learning, evolutionary artificial neural networks, and ordinal classification for both shallow and deep learning models.

Dr. Gutiérrez serves on the Editorial Board for the journal *IEEE Transaction on Neural Networks and Learning Systems* and on the organization/program committees for several computational intelligence conferences.

Emanuele Frontoni (Senior Member, IEEE) was born in Fermo, Italy.

He is currently a Full Professor of computer science at the University of Macerata, Macerata, Italy, and the Co-Director of the VRAI Laboratory, Department of Information Engineering (DII), Università Politecnica delle Marche, Ancona, Italy. He conducts his research in artificial intelligence and computer vision, human behavior analysis, augmented reality and sensitive spaces, and digital humanities. He has authored over 230 international articles and collaborates with numerous national and international companies in technology transfer and innovation activities.

Mr. Frontoni is a member of the European Association for Artificial Intelligence, the European AI Alliance, and the International Association for Pattern Recognition.

César Hervás-Martínez (Senior Member, IEEE) was born in Cuenca, Spain. He received the B.S. degree in statistics and operations research from the Universidad Complutense de Madrid, Madrid, Spain, in 1978, and the Ph.D. degree in mathematics from the University of Seville, Seville, Spain, in 1986.

He is currently a Computer Science and Artificial Intelligence Professor at the Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain. His current research interests include neural networks, evolutionary computation, and natural systems modeling.