Hierarchical Semantic Risk Minimization for Large-Scale Classification

Yu Wang, Zhou Wang, *Fellow, IEEE*, Qinghua Hu, *Senior Member, IEEE*, Yucan Zhou, and Honglei Su, *Member, IEEE*,

***Abstract*—Hierarchical structures of labels usually exist in large-scale classification tasks, where labels can be organized into a tree-shaped structure. The nodes near the root stand for coarser labels, while the nodes close to leaves mean the finer labels. We label unseen samples from the root node to a leaf node, and obtain multigranularity predictions in the hierarchical classification. Sometimes, we cannot obtain a leaf decision due to uncertainty or incomplete information. In this case, we should stop at an internal node, rather than going ahead rashly. However, most existing hierarchical classification models aim at maximizing the percentage of correct predictions, and do not take the risk of misclassifications into account. Such risk is critically important in some real-world applications, and can be measured by the distance between the ground truth and the predicted classes in the class hierarchy. In this work, we utilize the semantic hierarchy to define the classification risk and design an optimization technique to reduce such risk. By defining the conservative risk and the precipitant risk as two competing risk factors, we construct the balanced conservative/precipitant semantic (BCPS) risk matrix across all nodes in the semantic hierarchy with user-defined weights to adjust the tradeoff between two kinds of risks. We then model the classification process on the semantic hierarchy as a sequential decision-making task. We design an algorithm to derive the risk-minimized predictions. There are two modules in this model: 1) multitask hierarchical learning and 2) deep reinforce multigranularity learning. The first one learns classification confidence scores of multiple levels. These scores are then fed into deep reinforced multigranularity learning for obtaining a global risk-minimized prediction with flexible granularity. Experimental results show that the proposed model outperforms state-of-the-art methods on seven large-scale classification datasets with the semantic tree.**

***Index Terms*—Deep *Q*-network (DQN), granular computing, hierarchical classification, multigranularity learning, risk minimization.**

Manuscript received 23 January 2020; revised 23 November 2020; accepted 1 February 2021. Date of publication 17 March 2021; date of current version 18 August 2022. This work was supported in part by the National Key Research and Development Project under Grant 2019YFB2101901; in part by the National Natural Science Foundation of China under Grant 61925602, Grant 61732011, Grant 62076179 and Grant 62006221; and in part by the Tianjin Science and Technology Plan Project under Grant 19ZXZNGX00050. This article was recommended by Associate Editor S. Ozawa. *(Corresponding authors: Qinghua Hu; Yucan Zhou.)*

Yu Wang and Qinghua Hu are with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: armstrong_wangyu@tju.edu.cn; huqinghua@tju.edu.cn).

Zhou Wang is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: zhou.wang@uwaterloo.ca).

Yucan Zhou is with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100089, China (e-mail: zhouyucan@iie.ac.cn).

Honglei Su is with the School of Electronic Information, Qingdao University, Qingdao 266071, China (e-mail: suhonglei@qdu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3059631.

Digital Object Identifier 10.1109/TCYB.2021.3059631

I. Introduction

With the rapid growth of the Internet and Internet of Things, many large-scale classification tasks have been emerging in real-world applications. Usually, there exist hierarchical structures in such tasks, where labels are often organized into a tree-shaped structure. The nodes near the root stand for coarser labels, while the nodes close to leaves mean the finer labels. For example, the well-known ImageNet dataset uses a semantic hierarchy WordNet to collect more than 1 million images with over 22 000 classes [1], and Amazon designs the large hierarchies to organize billions of extensive products [2]. Leveraging the hierarchical structure of classes, namely, hierarchical classification, is effective and efficient for large-scale classification tasks [3]–[5], and has been widely used in various real-world applications [6]–[8].

In the hierarchical classification scenario, we label unseen samples from the root node to a leaf node, and obtain multigranularity predictions. Sometimes, we cannot obtain a leaf decision due to uncertainty or incomplete information. In this case, we should stop at an internal node, rather than going ahead rashly. However, most existing hierarchical classification models aim at maximizing the percentage of correct predictions, and do not take the risk of misclassifications into account [1], [3], [9]–[12]. Such risk is critically important in some real-world scenarios [13]–[15]. For example, some types of dogs are visually similar to wolves, which results in difficulties in identifying them. In a ranch, classifying a dog as a wolf is a high-risk mistake, because a wolf will kill all farmyard animals and lead to a disaster at the ranch. In this scenario, it is reasonable to reduce the risk by stopping the uncertain sample at an internal node and reporting a coarse-grained result *carnivore* for further checking (see Fig. 1).

In fact, the aforementioned classification risk describes the risk degree that a misclassification would take, and this is highly related to the characteristics of various tasks. Conventionally, classification risk is designed manually by human experts [16], [17] or is defined by simple statistical information, for example, the degree of imbalance between each class [18], [19]. However, the ways of defining risk are either cost expensive or inadequate for describing risk. Semantics is the relation between linguistic expressions and their meanings, which reflects the knowledge of humans [20]. The semantic hierarchy can properly describe multigranularity relations of different classes, and is easy to obtain and simple to use [21]. Therefore, it can be used to measure the classification risk. Such risk can be measured by the distance between the ground truth and the predicted classes in the class

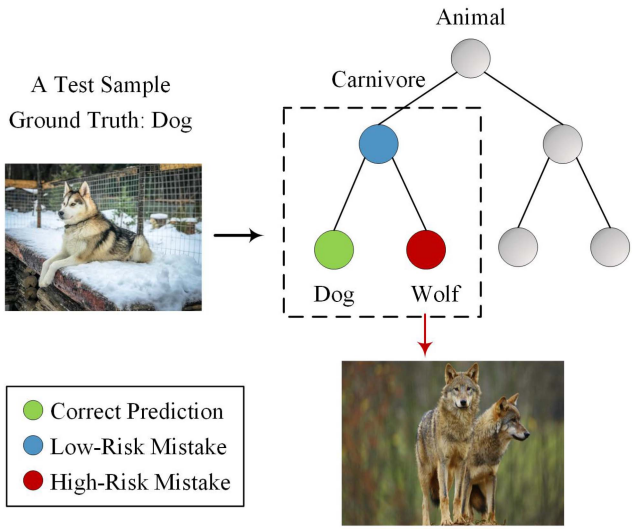


Fig. 1. In a ranch, classifying a dog as a wolf results in a high risk. Given a dog visually similar to a wolf, it is reasonable to stop the sample from going down rashly and report it as a carnivore for further checking, instead of making a high-risk mistake as a wolf.

hierarchy. Therefore, the key to this problem is how to make models learn to stop the various samples at different levels of the hierarchy to minimize the classification risk.

Several works have been dedicated to stopping uncertain samples at the proper nodes in the semantic hierarchy, and they generally use three kinds of strategies. The first kind offers conservative predictions to avoid misclassification by predicting the uncertain samples to the root node. Usually, the models learn a global threshold and directly assign the samples to the root node where probabilities of correct predictions are lower than the threshold [19], [22], [23]. Unfortunately, this strategy is often too conservative in that it loses information provided by lower levels. The second kind encourages samples toward the leaf nodes to make more informative predictions [9]–[11] by optimizing the flat evaluation metrics, for example, the F measure, on the leaf nodes. However, they tend to be too precipitant to make wrong predictions [24]. Aiming at overcoming the problems of the above two types, the third kind takes a balance into account to achieve a tradeoff between predicting conservatively and informatively based on the semantic tree structure. Ceci and Malerba [25] utilized the tree induced error (TIE) to measure the degree of error predictions, attempting to stop the samples at the proper nodes in the tree by setting a threshold on each node, and a predefined candidate set is used, making it difficult to make optimal predictions in the large-scale data. Wang *et al.* [24] tried to improve by optimizing TIE in a global view. They used the generic algorithm (GA) and achieved a better result. Deng *et al.* [1] tried to maximize the information gain while ensuring the given hierarchical accuracy to obtain the predictions with appropriate granularity. Recently, Lee *et al.* [12] used the confidence level of the classifier to determine whether the classification process should be stopped at the current node. They jointly learn the classifier and the confidence thresholds and show good performance in the zero-shot learning.

Despite the progress made by recent works, two problems remain. On the one hand, there is no simple mechanisms to define the risks for different real-world application scenarios. Some methods make use of the semantic hierarchy to report results of flexible granularity, but most of them are designed to avoid classification errors [9]–[11] or seek the informativeness of correct predictions [1], [12]. Moreover, all the methods overlook the fact that risks should be adjusted for various applications. On the other hand, effective optimization algorithms and machine learning methods for risk minimization are still open problems. Specifically, the optimization of hierarchical metrics is a nonconvex and underivative problem, which has not been well addressed in hierarchical risk minimization scenarios [24], [25].

In this work, we aim to solve the aforementioned problems. First, we design a new risk matrix utilizing the semantic tree structure to dynamically measure the risk degree in different scenarios. Inspired by [1], we regard the prediction errors as two types of risks: 1) the conservative risk and 2) the precipitant risk. Predicting a sample as a more abstract and coarse-grained label avoids misclassification but takes conservative risk of losing information provided by the lower levels. In contrast, seeking an informative prediction too aggressively takes precipitant risk of misclassification. We propose to jointly consider both types of risks as competing factors and define a balanced conservative/precipitant semantic (BCPS) risk, where the weights of the risk factors are adjustable. Second, we design a new model to effectively obtain risk-minimized predictions, with two modules of multitask hierarchical learning and deep reinforced multigranularity learning. The multitask hierarchical learning module jointly learns the classification confidence in multiple levels, which are fed into the deep reinforced multigranularity learning to obtain a global risk-minimized prediction with flexible granularity by the deep Q-network (DQN). By using deep reinforcement learning, the prediction considers the risk of the current local decision as well as the low-level decisions under the guidance of the long-term reward. We evaluate the proposed hierarchical semantic risk minimization (HSRM) model on seven datasets with semantic hierarchy, and demonstrate that it achieves state-of-the-art performance compared with existing hierarchical models and is flexible to produce multigranularity outputs.

The contributions of this article are summarized as follows.

1) We propose a new hierarchical classification model which makes use of the semantic hierarchy to define and minimize the risk in the large-scale classification task.
2) A BCPS risk is defined by leveraging the semantic hierarchy to dynamically measure the degree of risk in different scenarios, consisting of conservative risk and precipitant risk serving as two competing factors.
3) We model the classification process as a sequential decision-making task, and make use of deep reinforcement learning to solve the nonconvex and underivative optimization problem of the BCPS risk and other hierarchical metrics.

The remainder of this article is organized as follows. Section II elaborates the definition of the problem. Section III describes the details of the proposed HSRM model. Experimental results on various dimensions are presented in Section IV. Finally, we conclude our work in Section V.

II. Problem Definition

Hierarchical classification is the classification task which is performed based on a class hierarchy that organizes classes into a hierarchical structure where the granularity ranges from coarse grained to fine grained. There are two kinds of structures in class hierarchy: 1) tree and 2) directed acyclic graph (DAG). We focus on tree in this work since the tree structure is the most common and widely used in semantics.

A tree hierarchy organizes class labels into a tree-like structure to represent a kind of "IS-A" relationship between labels [26]. Specifically, Aris *et al.* [27] pointed out that the properties of the IS-A relationship can be described as asymmetry, anti-reflexivity, and transitivity. We define a tree as a triplet $(\mathcal{V}, E, \pi)$ with a group of edges E between nodes in different levels, "parent–child" relationship π, and a set of nodes $\mathcal{V} = \{\mathcal{V}_N, \mathcal{V}_L\}$, where $\mathcal{V}_N$ denotes nonleaf nodes and $\mathcal{V}_L$ the leaf nodes. Generally, there are several types of nodes in a tree. For a given node v, its parent node is denoted by P_v; its child nodes are denoted by C_v, its leaf nodes are denoted by $\mathcal{L}_v$, and $|\mathcal{L}_v|$ is the number of leaf nodes of v. $\mathcal{V}_N$ is the nonleaf nodes of the tree, $\mathcal{V}_L$ denotes the leaf nodes of the tree, and $|\mathcal{V}_L|$ denotes the number of all leaf nodes.

Given training samples $\{\boldsymbol{X}_t^k\}_{k=1}^K$ with labels $\{Y_t^k\}_{k=1}^K$, a semantic tree structure $(\mathcal{V}, E, \pi)$, our goal is to first construct a risk matrix ξ which contains risks of all the possible predictions $\mathcal{V}$ based on the semantic tree structure $(\mathcal{V}, E, \pi)$, and subsequently learn a classifier which predicts classes by minimizing the risks according to ξ for the training samples $\{\boldsymbol{X}_t^k\}_{k=1}^K$ within set $\mathcal{V}$, that is, not only the leaf nodes $\mathcal{V}_N$, but the internal nodes $\mathcal{V}_L$ as well. The propagation process of the samples starts from the root node, for example, node #1 in Fig. 2, to its child, that is, nodes #2 or #3, according to the probabilities and terminates until the risk-minimal nodes are reached.[1]

III. Model and Solution

Before minimizing the risk of a hierarchical classification task, we require a clear definition of the risk, a classifier to model the multigranularity hierarchical classification process, and an effective algorithm to solve the optimization problem of the risk based on the performance of the classifier. In this section, we first define a BCPS risk to measure the risk of hierarchical classification (Section III-A), then model the hierarchical classification process as a sequential decision-making task between stopping at the current node or going down to the lower level. A multitask hierarchical learning model is designed to jointly predict the classification confidence scores of multiple levels (Section III-B). Finally, the scores are fed into a subsequent DQN to learn to obtain a global optimal prediction by minimizing the BCPS risk (Section III-C). The framework of the proposed method is shown in Fig. 3.

[1]Such a sample-propagation manner is called the top-down process, and another possible manner is the bottom-up process, where the sample starts from the leaf nodes and propagates to its parent nodes. In this article, we focus on the former, because it is the most common case of hierarchical classification and can reflect the coarse-to-fine way of handling a complex problem.

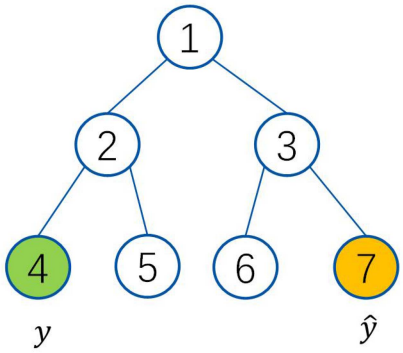


Fig. 2. Toy example for hierarchical metrics. y is the ground-truth node and $\hat{y}$ is the prediction node.

A. Balanced Conservative/Precipitant Semantic Risk Matrix

Conventionally, the cost matrix in cost-sensitive learning records the cost of misclassifying a sample to a class other than the ground-truth class [13], [18]. But the cost matrix is constructed by experts of domain knowledge for the specific task, which is hard to obtain, or by using statistic information, for example, data distribution, which is often difficult to apply in more general real-world applications. Moreover, the conventional cost matrix only contains the corresponding costs between the ground-truth class and the candidate classes in the sense that the predictions are only correct or wrong. Inspired by the setting of the hierarchical classification, we believe that it is reasonable to report a more abstract internal class of a semantic tree without having to make a wrong decision. Meanwhile, this coarse-grained prediction loses information provided by the lower levels, which also has risk in the prediction. Thus, we extend the conventional cost matrix to a risk matrix which contains both the leaf and internal classes based on the semantic tree.

Definition 1: Given a sample set $\{\boldsymbol{X}_t^k\}_{k=1}^K$ with label set $\{\boldsymbol{Y}_t^k\}_{k=1}^K$ and a semantic tree structure $(\mathcal{V}, E, \pi)$ with node set $\mathcal{V} = \{\mathcal{V}_N, \mathcal{V}_L\}$, the risk matrix ξ has dimensions of $|\mathcal{V}| \times |\mathcal{V}|$, and the element $\xi(i, j)$ in the risk matrix denotes the risk of classifying class i to class j.

For the risk matrix ξ, the problem of computing each element is transformed to a risk measurement between different nodes of the tree structure. There are two existing metrics, TIE [28] and hierarchical F measure (HF) [27], that describe the error degree according to the tree structure. In hierarchical evaluation metrics, the ground-truth class and the prediction class are extended to the augmented ground-truth class set and the augmented prediction class set by adding all the parent nodes of each corresponding class

$$\begin{aligned} Y_{\text{aug}} &= y \cup \pi(y) \cup \pi(\pi(y)) \cup \cdots \cup \Gamma \\ \hat{Y}_{\text{aug}} &= \hat{y} \cup \pi(\hat{y}) \cup \pi(\pi(\hat{y})) \cup \cdots \cup \Gamma \end{aligned} \tag{1}$$

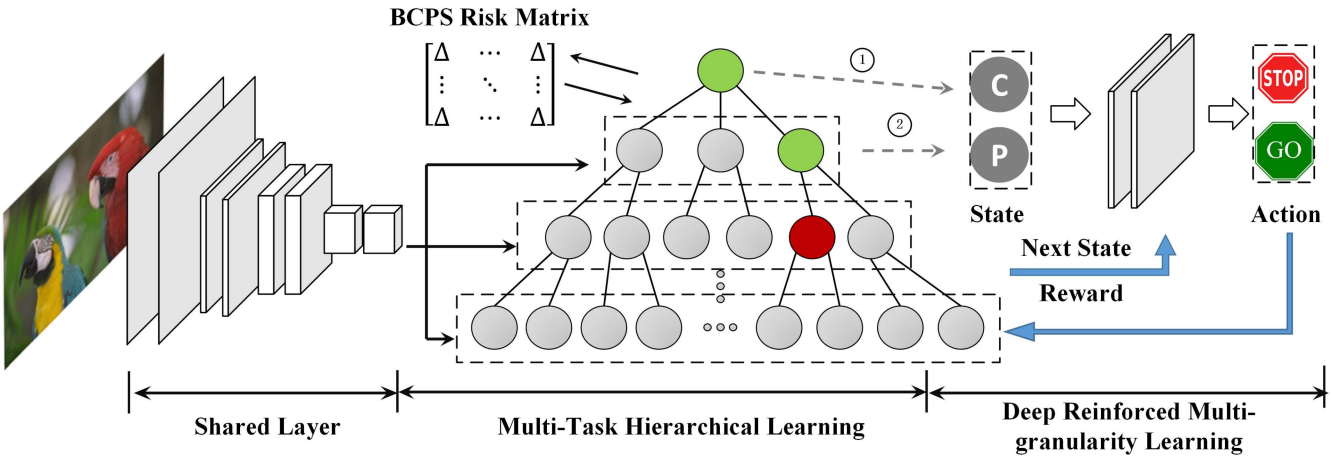


Fig. 3. Framework of proposed model HSRM. It constructs a BCPS risk matrix with adjustable emphasis on the conservative risk and the precipitant risk, and then jointly trains a hierarchical multitask classifier. By considering ① the information loss by stopping at the current node (gray circle "C") and ② uncertainty of going down to the lower level (gray circle "P"), the DQN makes predictions with the minimum risk according to the BCPS risk matrix.

where y denotes the ground-truth class in the tree, $\hat{y}$ denotes the prediction class in the tree, $\pi(\cdot)$ is the parent relation, and Γ is the root node of the tree. Based on the augmented true and prediction class in (1), the TIE and HF are defined.

Definition 2: Given a tree structure $(\mathcal{V}, E, \pi)$, TIE equals the number of edges in the link from one node to another

$$\begin{aligned}\Phi(y,\hat{y}) &= \left(|Y_{\text{aug}}| - \left|Y_{\text{aug}} \cap \hat{Y}_{\text{aug}}\right|\right) \\ &\quad + \left(\left|\hat{Y}_{\text{aug}}\right| - \left|Y_{\text{aug}} \cap \hat{Y}_{\text{aug}}\right|\right) - 1 \\ &= |Y_{\text{aug}}| + \left|\hat{Y}_{\text{aug}}\right| - 2\left|Y_{\text{aug}} \cap \hat{Y}_{\text{aug}}\right| - 1 \end{aligned} \tag{2}$$

where $|\cdot|$ is the cardinality of the set.

TIE can appropriately describe the error degree in the tree in that the severeness of the wrong prediction is proportional to the distance between the prediction node and the ground-truth node. A toy example is shown in Fig. 2. Suppose node 4 is the ground-truth node and node 7 is the prediction node. Basically, $Y_{\text{aug}} = \{4, 2, 1\}$, $\hat{Y}_{\text{aug}} = \{7, 3, 1\}$, and this leads to $|Y_{\text{aug}} \cap \hat{Y}_{\text{aug}}| = |\{1\}| = 1$, $|Y_{\text{aug}}| = |\hat{Y}_{\text{aug}}| = 3$, so TIE $= 4$. However, it overlooks the fact that the percentage of correct proportion of the path in the tree may be different even if the absolute "distance" from the ground truth is the same. Another metric named HF overcomes this problem by adding normalization factors, and it extends the F measure in flat classification into a hierarchical one.

Definition 3: Given a tree structure $(\mathcal{V}, E, \pi)$, the hierarchical precision, recall, and F measure are defined as

$$\begin{aligned} P_H(y,\hat{y}) &= \frac{\left|\hat{Y}_{\text{aug}} \cap Y_{\text{aug}}\right|}{\left|\hat{Y}_{\text{aug}}\right|} \\ R_H(y,\hat{y}) &= \frac{\left|\hat{Y}_{\text{aug}} \cap Y_{\text{aug}}\right|}{|Y_{\text{aug}}|} \\ \text{HF}(y,\hat{y}) &= \frac{2 \cdot P_H R_H}{P_H + R_H} \end{aligned} \tag{3}$$

where $|\cdot|$ is the cardinality of the set.

Although HF solves the scaling problem via set normalization, the combination of P_H and R_H for HF is somewhat arbitrary and does not provide a mechanism to adjust the balance between the two factors for specific applications. The predictions in the location of the parent nodes of the ground truth avoid wrong predictions but lose information provided in the lower levels, which results in the *conservative risk* for the conservative predictions. On the other hand, those predictions not located at any parent node of the ground truth, seek more informative predictions at the price of possibly going to the wrong path, resulting in *precipitant risk* for being too precipitant in decisions. We define the two types of risks formally as follows.

Definition 4: Given a tree structure $(\mathcal{V}, E, \pi)$, the conservative risk R_C and the precipitant risk R_P are defined as

$$\begin{aligned} R_C(y,\hat{y}) &= 1 - \frac{\left|\hat{Y}_{\text{aug}} \cap Y_{\text{aug}}\right|}{|Y_{\text{aug}}|} \\ R_P(y,\hat{y}) &= 1 - \frac{\left|\hat{Y}_{\text{aug}} \cap Y_{\text{aug}}\right|}{\left|\hat{Y}_{\text{aug}}\right|}. \end{aligned} \tag{4}$$

We differentiate various kinds of risks by judging whether the prediction belongs to the augmented set of the ground truth Y_{aug}, and add a normalization term like HF to take into account the depths of the ground truth and prediction classes in risk assessment. A toy example is shown in Fig. 2. For the ground-truth node 4 and prediction node 7, $R_C(y,\hat{y}) = R_P(y,\hat{y})1 - (1/3) = (2/3)$. Furthermore, the two kinds of risks should achieve a balance depending on real-world applications, so we introduce a tradeoff parameter in calculating the total risk as follows.

Definition 5: The balanced conservative/precipitant (BCPS) risk, or the element (i, j) of the BCPS risk matrix ξ is defined as

$$\begin{aligned}\xi(i,j) &= \lambda R_C(i,j) + (1-\lambda) R_P(i,j) \\ &= 1 - \left(\lambda \frac{\left|\hat{Y}_{\text{aug}} \cap Y_{\text{aug}}\right|}{|Y_{\text{aug}}|} + (1-\lambda)\frac{\left|\hat{Y}_{\text{aug}} \cap Y_{\text{aug}}\right|}{\left|\hat{Y}_{\text{aug}}\right|}\right) \end{aligned} \tag{5}$$

where $\xi(i, j)$ denotes the risk of predicting class i as class j, and λ is a tradeoff parameter between [0, 1]. For predictions, which report correct but abstract results, the second term equals zero, which means that only the conservative risk exists in these predictions. For those which report incorrect results, the first term encourages the predictions to include more correct nodes, and the second term penalizes too precipitant predictions. Such a risk matrix cannot only address the issue of the risk of reporting a more abstract result but put adjustable emphasis on conservative and precipitant risks for different applications. The properties of the BCPS risk matrix are listed as follows.

Properties of the Proposed BCPS Risk Matrix ξ:

Lemma 1: ξ is non-negative, that is, $\xi \geq 0$.

Lemma 2: The value of the BCPS risk is between 0 and 1, that is, $\forall \xi(i, j) \in [0, 1]$.

Lemma 3: ξ is asymmetric if $\lambda \neq 0.5$ and the tree $(\mathcal{V}, E, \pi)$ is unbalanced.

Lemma 4: An arbitrary TIE $\Phi(y, \hat{y})$ is an element without normalization in ξ if the tree $(\mathcal{V}, E, \pi)$ is balanced.

Lemma 5: The HF is the special case for the proposed risk matrix ξ with arbitrary structure of the tree $(\mathcal{V}, E, \pi)$.

B. Multitask Hierarchical Classification

In the conventional hierarchical classification process, a local classifier is trained using the extracted features on each nonleaf node of the tree structure, and the sample starts from the root node and is assigned to the child node with maximum confidence score recursively until it reaches a leaf node [9], [29]. The method requires buiilding a large number of local classifiers after the features are extracted in advance. Recently, a few studies address the problem and use a convolutional neural network (CNN) to jointly learn the features and level-wise classifier in a multitask way [3]. However, they force the samples to reach a leaf node, so a hyperparameter has to be put in each task as a measurement of influence for the current level on the leaf node level, which is difficult to use especially for those tree structures with many levels, for example, ImageNet.

In this work, we discard the hyperparameter, and leverage the idea of multitask learning to simultaneously train classifiers for different levels. Specifically, we denote the input training mini-batch samples $\{\boldsymbol{X}_t^k, \boldsymbol{Y}_t^k\}_{k=1}^K$, where t in X_t^k or Y_t^k represents the samples of the training set, $\boldsymbol{X}_t^k$ is the kth raw samples, and $\boldsymbol{Y}_t^k$ is the corresponding ground-truth labels. Assume the semantic tree structure $(\mathcal{V}, E, \pi)$ has H levels, the ground-truth label $\boldsymbol{Y}_t^k$ of sample $\boldsymbol{X}_t^k$ can be extended to $\boldsymbol{Y}_{H_t}^k$, which contains H labels of all the levels. Inspired by the network architecture proposed by Ma *et al.* [30], we modify the current CNN model by sharing the convolutional features for all H tasks to enhance the ability of more discriminative feature learning for the CNN model. In each task, we use the same fully connected layers and softmax layer. In optimization, the loss function in each layer is set as empirical cross entropy loss

$$l_h\left(\left\{\boldsymbol{X}_t^k\right\}_{k=1}^K; \boldsymbol{W}_h\right) = -\sum_{k=1}^{K}\sum_{i=1}^{C} p_k^i \log \hat{p}_k^i\left(\left\{\boldsymbol{X}_t^k\right\}_{k=1}^K; \boldsymbol{W}_h\right) \tag{6}$$

where $\boldsymbol{W}$ is the model parameters. The overall loss function of the multitask hierarchical classification is defined as

$$l\left(\left\{\boldsymbol{X}_t^k\right\}_{k=1}^K; \boldsymbol{W}\right) = \sum_{h=1}^{H} \zeta_h l_h \tag{7}$$

where ζ_h is the balance weight to account for the scale difference between the two terms.

C. Deep Reinforced Multigranularity Learning

Given the confidence scores of different levels obtained by multitask hierarchical classification, the model should make the risk-minimized predictions with flexible granularity by minimizing the BCPS risk. We consider the prediction process as a sequence of stopping or going down decisions from the root node toward the leaf nodes. At each step, we measure the local uncertainty, including information loss by stopping at the current level and uncertainty of going down to the lower level. Since the BCPS risk is underivative, we leverage the deep reinforcement learning method, specifically DQN [31] in this work, to approach the global optimum solution and take into account the lower level decisions for the current decision.

1) Measuring the Local Uncertainty:

*a) Information loss by stopping:*Recall that the conservative risk defined in Definition 4 is the information loss due to more abstract and coarse-grained predictions, because choosing to stop at the current level will lose information provided by the next level. To measure the information loss, we develop from the concept of information gain introduced by Deng *et al.* [1], which is described as the decrease in number of leaf nodes when taking an action in comparison to staying at the root node. Specifically, for an arbitrary node v, its information gain of node v corresponding to the root node is calculated as

$$I(v) = \log|\mathcal{V}_L| - \log|\mathcal{L}_v| \tag{8}$$

where $|\mathcal{V}_L|$ is the number of leaf nodes in the tree and $|\mathcal{L}_v|$ is the number of leaf nodes corresponding to node v. In our case, the information loss by stopping is defined as the information gain from the current node to its child node.

Definition 6: Given a tree structure $(\mathcal{V}, E, \pi)$ with a set of nodes $\mathcal{V}$, let $\mathcal{V}_L$ be the set of leaf nodes, $\mathcal{L}_v$ be the set of leaf nodes corresponding to the current nonleaf node v_N, $\mathcal{V}_N = \mathcal{V} - \mathcal{V}_L$ be the set of nonleaf nodes, and $C(v_N)$ be the child nodes of node $v_N \in \mathcal{V}_N$. The information loss by stopping from the level of v_N to the level of ith child nodes $C(v_N)_i$ is

$$\begin{aligned} R_{v_N}^{\mathrm{LC}} &= I(C(v_N)) - I(v_N) \\ &= \log\left|\mathcal{L}_{v_N}\right| - \log\left|\mathcal{L}_{C(v_N)_i}\right|. \end{aligned} \tag{9}$$

Equations (8) and (9) show that deeper prediction nodes contain more information gain than the shallow ones. Predictions at the root node have no information gain, while the information gain reaches the maximum at the leaf nodes.

*b) Uncertainty of going down:*We can encourage all the samples to go down to the leaf node if only the information loss by stopping is taken into consideration. Unfortunately, misclassification may occur at some nodes, especially if the uncertainty in the lower level is large. Previous works [12]

and [24] show that errors usually occur when the classifier is not sure about which node to assign in the lower level. In this article, we not only include this factor to model the uncertainty but also take into account if the classifier is reliable to produce the correct result.

We leverage the information entropy to account for how confident the classifier is for its prediction. Given a discrete variable M with possible values $\{m_1, m_2, \ldots, m_n\}$, the information entropy is explicitly written as

$$H(M) = \sum_{i=1}^{n} -p_i \log p_i$$

where p_i is the probability of the value m_i. At each nonleaf node v_N, it is straightforward to measure the risk of misclassification by using the confidence of the multitask output for all the child nodes $C(v_N)$

$$P^c_{v_N}(\boldsymbol{x}) = \sum_{i=1}^{|C(v_N)|} -p_{c_i}(\boldsymbol{x}) \log p_{c_i}(\boldsymbol{x}) \tag{10}$$

where v_N is an arbitrary nonleaf node, $|C(v_N)|$ is the number of child nodes of v_N and $p_{c_i}(x)$ is the confidence score of assigning sample $\boldsymbol{x}$ to the ith child node of v_N.

Moreover, we split a subvalidation set of samples $\{\boldsymbol{X}^q_{sv}\}^Q_{q=1}$ to model if the classifier (multitask output) is reliable to produce the correct result using

$$P^r_l = \exp\Big(1 - \epsilon_l\Big(\big\{\boldsymbol{X}^q_{sv}\big\}^Q_{q=1}, \boldsymbol{W}\Big)\Big) \tag{11}$$

where l is the level, ϵ_l is the generalized accuracy in lth level. Then, we define the uncertainty of going down as follows.

Definition 7: Given the sample $\boldsymbol{x}$ at node v, the trained multitask hierarchical classifier $\boldsymbol{W}$, the uncertainty of going down R_P at node v is defined as

$$\begin{aligned} R^P_v(\boldsymbol{x}) &= P^c_v \cdot P^r_{l(v)} \\ &= \left(\sum_{i=1}^{|C(v_N)|} -p_{c_i}(x) \log p_{c_i}(x), \right) \\ &\quad \times \exp\Big(1 - \epsilon_l\Big(\big\{\boldsymbol{X}^q_{sv}\big\}^Q_{q=1}, \boldsymbol{W}\Big)\Big) \end{aligned} \tag{12}$$

where $l(v)$ is the level of the node v in the semantic tree. It is worth noting that there are two main properties of (12).

Lemma 6: Given a sample $\boldsymbol{x}$ at node v, $R^P_v(\boldsymbol{x})$ is decided by the first term P^c_v if the value of the second term $P^r_{l(v)}$ is small.

Lemma 7: Given a sample $\boldsymbol{x}$ at node v, $R^P_v(\boldsymbol{x})$ is large if the value of the second term $P^r_{l(v)}$ is large.

2) Risk Minimization With Deep Q-Network: A local decision may be made to reduce the BCPS risk by weighting information loss by stopping and uncertainty of going down. However, optimal local decisions may not necessarily lead to global optimum in terms of total BCPS risks, for which we should consider the interdependent relations between different local decisions, where a certain local decision should take the results of later local decisions into account. For example, given a certain sample, the local decision at the first level is not sure about whether to stop or go down, but the local decision at the second level is confident to go down to the third level. In this scenario, the low-level decision helps the high level to make a decision of going down instead of stopping at the first level by its own local decision. In this regard, any local decisions at a certain level should consider information from lower levels, aiming for a global total rewards as measured by the total BCPS risks. The effect has been verified by our experiment, and an example is given in Fig. 4. Since this is analogous to a reinforcement learning problem, we resort to the DQN [31] to lay out the framework and to train the decision-making NN for global risk minimization.

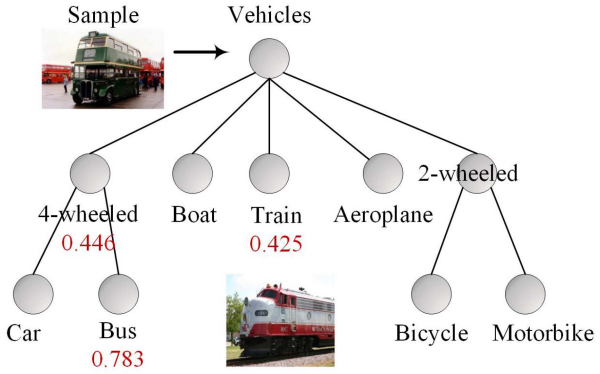


Fig. 4. Sample in PASCAL VOC experiment. The confidence scores given by multitask hierarchical model are difficult to differentiate between *4-wheeled* and *Train*, but can be easily recognized as *Bus* in the lower level. DQN is capable of capturing this long-term interdependent task relationship to predict in a global view.

*a) Action:*The action space $\mathcal{A}$ is a set of all possible actions that the agent could make. In our case, the actions contain stopping at the current node or going down to the child node with maximum confidence score.

*b) State:*The state $\mathcal{S}$ contains information that the agent could observe. Learning a good and stable policy is challenging in general [32]. Mao *et al.* [33] used high-dimensional text features for hierarchical text classification, but have to apply supervised pretraining to help learn a good policy. We generalize the state as a 2-D input with the information loss by stopping and uncertainty of going down corresponding to various samples at different nodes, which significantly helps learn a good policy for DQN.

*c) Rewards:*The reward $\mathcal{R}(s, a)$ given by the environment guides the agent and trains it by learning to maximize the cumulative reward. With the designed risk matrix ξ computed in (5), the proposed model can capture the reward with various predictions on all nodes of the semantic tree. Motivated by the idea of Mao *et al.* [34], we give intermediate rewards to the local decision of each step to help improve the learning process. The intermediate reward is set as the difference between the current step and the last step, which encourages the sample to go down if it is positive or stop if it is negative serving as an indicator. On the other hand, the cumulative reward from the current step to the end of an episode would cancel the intermediate rewards and thus reflect whether the current action improves the overall prediction process [34]. Specifically, given a sample $\boldsymbol{x}$ at node v, we set the intermediate reward $R^{it}_v(\boldsymbol{x}) = R_v - R_{\pi(v)}$, where $R_v = [1/(\xi(y_{\boldsymbol{x}}, v))]$ ($y_{\boldsymbol{x}}$ is the ground-truth class of $\boldsymbol{x}$). The

episode is ended if a stopping decision is made or a leaf node is reached. As a result, the agent can learn to make risk-minimized predictions with a long term and global view.

d) Training: In DQN, upon taking an action a at the current state s, the agent receives a reward $R(s, a)$ and reaches a new state s', determined from the probability distribution $P(s'|s, a)$. A policy π specifies for each state which action the agent will take. The goal of the agent is to find the policy π mapping states to actions that maximizes the expected discounted cumulative reward over the agent's lifetime. The value $Q(s, a)$ of a given state-action pair (s, a) is an estimate of the expected future reward that can be obtained from (s, a) when following policy π. The optimal value function $Q^*(s', a')$ provides maximal values in all states and is determined by solving the Bellman equation

$$Q^*(s,a) = \mathbb{E}\left[R(s,a) + \gamma \sum_{s'} P\big(s'|s,a\big) \max Q^*\big(s',a'\big)|s,a\right]. \tag{13}$$

The optimal policy π is then $\pi(s) = \arg\max_{a\in\mathcal{A}} Q^*(s, a)$. In our scenario, assuming the sample $\boldsymbol{x}$ on node v, $s = [R_v^{\mathrm{LC}}(\boldsymbol{x}), R_v^P(\boldsymbol{x})]$, $a = [0, 1]$, $s' = [R_{Ch(v)}^{\mathrm{LC}}(\boldsymbol{x}), R_{Ch(v)}^P(\boldsymbol{x})]$, where $Ch(v)$ is the child node of v with them maximal confidence score.

During the learning, we follow the method in [31] for Q-learning update. First, mini-batches of experience are uniformly drawn from the pool of stored samples. At each iteration i, the loss function used is

$$L_i(\theta_i) = \mathbb{E}_w\left[\left(R(s,a) + \gamma \max_{a'} Q\big(s',a';\theta_i^-\big) - Q(s,a;\theta_i)\right)^2\right] \tag{14}$$

where $w = (s, a, r, s') \sim U(D)$ denotes a sample of experience with a tuple of state, action, reward and the next state drawn uniformly from the experience pool; γ is the discount factor; θ_i are the parameters of the Q-network at iteration i and θ_i^- are the parameters used to compute the target at iteration i. In the process of learning, the target network parameters θ_i^- are updated every G steps with the Q-network parameters θ_i. The optimal target values $R(s, a) + \gamma \max Q^*(s', a')$ are substituted with the approximate target values $R(s, a) + \gamma \max_{a'} Q(s', a'; \theta_i^-)$ for the well-defined optimization, which has been proved to be stable and efficient [35].

With different discount factor γs, the agent will take various emphases on the future rewards into account. In our case, the global risk-minimized problem is degenerated to a local supervised learning problem by only making local decisions of stopping or going down based on the local risks ($\lambda = 0$). Our experiment shows the global risk minimization performs better than the local one, by considering information of lower levels.

We formulate the above process in Algorithm 1. First, we build the BCPS risk matrix with adjustable emphasis on the conservative risk and precipitant risk. Then, a multitask hierarchical learning classifier is trained without hyperparameter on weighted effect on the leaf node level. Subsequently, the DQN takes information loss by stopping and uncertainty of going down as input state to make an action with the rewards based on the BCPS risk matrix, and finally, we obtain global risk-minimized predictions.

Algorithm 1: HSRM

Input: Training samples $\{\boldsymbol{X}_t^k, \boldsymbol{Y}_t^k\}_{k=1}^K$, subvalidation samples $\{\boldsymbol{X}_{sv}^q, \boldsymbol{Y}_{sv}^q\}_{q=1}^Q$, validation samples $\{\boldsymbol{X}_v^m, \boldsymbol{Y}_v^m\}_{m=1}^M$, and semantic tree structure $(\mathcal{V}, E, \pi)$.

Output: Multitask hierarchical classifier $\boldsymbol{W}$, deep Q-network $\boldsymbol{W}_R$.

$R_C(y, \hat{y}) \leftarrow 1 - \frac{|\hat{Y}_{\mathrm{aug}} \cap Y_{\mathrm{aug}}|}{|Y_{\mathrm{aug}}|}$, $R_P(y, \hat{y}) \leftarrow 1 - \frac{|\hat{Y}_{\mathrm{aug}} \cap Y_{\mathrm{aug}}|}{|\hat{Y}_{\mathrm{aug}}|}$;

$\xi(i, j) \leftarrow \lambda R_C(i, j) + (1 - \lambda) R_P(i, j)$;

for *mini-batch* $b = 1{:}B_t$ **do**

  $\boldsymbol{W} \leftarrow -\sum_{h=1}^H \sum_{k=1}^K \sum_{i=1}^C \zeta_h p_k^i \log \hat{p}_k^i(\{\boldsymbol{X}_t^k(b)\}_{k=1}^K; \boldsymbol{W}_h)$;

$P_l^r \leftarrow \exp(1 - \epsilon_l(\{\boldsymbol{X}_{sv}^q\}_{q=1}^Q, \boldsymbol{W}))$;

for *mini-batch* $b = 1{:}B_v$ **do**

  while $v \notin \mathcal{V}_L$ && *stop* == *False* **do**

    $R_{v_N}^{LC} \leftarrow \log|\mathcal{L}_{v_N}| - \log|\mathcal{L}_{C(v_N)_i}|$,

    $P_{v_N}^c(\boldsymbol{X}_v^m(b)) \leftarrow \sum_{i=1}^{|C(v_N)|} -p_{v_c^i}(\boldsymbol{x}) \log p_{v_c^i}(\boldsymbol{X}_v^m(b))$,

    $R_v^P(\boldsymbol{X}_v^m(b))) \leftarrow \left(\sum_{i=1}^{|C(v_N)|} -p_{v_c^i}(x) \log p_{v_c^i}(x)\right) \cdot \exp(1 - \epsilon_l(\{\boldsymbol{X}_{sv}^q\}_{q=1}^Q, \boldsymbol{W}))$,

    $\boldsymbol{W}_R \leftarrow (R_{v_N}^{LC}, R_v^P(\boldsymbol{X}_v^m(b)), \xi)$,

    if $\boldsymbol{W}_R(\boldsymbol{X}_v^m(b))$==*stop* **then**

      stop = *True*;

    else

      $v \leftarrow C(v)_{\max}$;

      stop = *False*;

return $\boldsymbol{W}$, $\boldsymbol{W}_R$.

The computations of the proposed hierarchical classification system are mainly on the multitask hierarchical learning module, so the analysis of computational complexity focuses on this part. Concretely, the computational complexity of this module is $O(M_i \cdot N_i \cdot \mathcal{D}^2 \cdot K_H \cdot K_W + H \cdot M_f \cdot D_{f_{\mathrm{in}}} \cdot D_{f_{\mathrm{out}}})$, where M_i is the spatial width of the input map; N_i is the spatial height of the input map; $\mathcal{D}$ is the depth of the previous and current layer; K_W is the width of the kernel; K_H is the height of the kernel; $D_{f_{\mathrm{in}}}$ is the number of input of the fully connected layer; $D_{f_{\mathrm{out}}}$ is the number of output of the fully connected layer, and M_f is the number of the fully connected layers. The proposed model only has several more fully connected layers according to number of the hierarchy, so its computational complexity is comparable with traditional CNN models, and hence can be scaled well to various hierarchies with different levels.

IV. Experiments

A. Experimental Setups

1) Implementation Details: In the training phase of both multitask learning and DQN, the Adam optimization algorithm is applied with a mini-batch of 128. The learning rate used is

TABLE I
DATASET DESCRIPTION

datasets	#Sample	#Leaf	#Node	Depth
PASCAL VOC	34828	20	30	5
Stanford Cars	16185	196	206	3
ILSVRC 65	23546	60	65	4
Cifar 100	60000	100	121	3
Caltech256	30607	256	277	3
SUN	90212	324	343	4
ImageNet 1K	1321167	1000	1860	19

$\alpha = 10^{-4}$ or $\alpha = 10^{-5}$ with different datasets. Other parameters in Adam optimization are set by default. The balance parameters ζ are set following the setting in [3] which are various in different datasets. All the datasets are split into training set, subvalidation set, validation set, and test set by 50%, 10%, 20%, and 20%, respectively. The training set is used to train multitask hierarchical classifier, the subvalidation set is applied to obtain the accuracy score in the reliability part (11), the validation set is utilized to train the DQN, while the test set is applied to obtain the test results. All the results shown are the average of ten trails of random splitting.

2) Datasets: We perform experiments on seven datasets with the semantic tree structure (see Table I), which are commonly used in hierarchical classification problems, including PASCAL VOC [36], ILSVRC65 [1], Stanford Cars [37], Cifar-100 [38], Caltech256 [39], SUN [40], and ImageNet 1K [21]. For the SUN dataset, we modify it by leaving out the categories that have more than one parent labels and samples with multiple labels. Finally, the SUN dataset turns into 324 classes with at least 100 images per category. It is worth mentioning that these datasets can test all the models in various perspectives. For example, the PASCAL VOC dataset has some images of multiple objects but with a single label. For example, a person is leading a horse in an image, but the ground truth given of this image is only *Person*. The Stanford Cars dataset consists of various cars in fine-grained classes, which is challenging to differentiate. The SUN and ImageNet datasets have massive labels, and the latter has a very complicated semantic tree structure with 19 levels. We use all these datasets to provide a comprehensive testing.

3) Evaluation Metrics: To evaluate the performance, flat accuracy is a common used metric in classification tasks. But it calculates at the leaf nodes and does not take into account the varying degrees of classification risks. Therefore, we use classic hierarchical evaluation metrics TIE [28], HF [27], and the proposed BCPS risk to measure the performance for intuitive understanding and fair comparison.

B. Results on Classification Tasks

1) Experimental Settings: We compare the proposed model with classic and state-of-the-art algorithms of hierarchical methods which assign samples to arbitrary nodes of the semantic tree, including TSS [25], DARTS [1], TDKL [12], and RMGA [24]. We also compare the cost-sensitive learning method [18], which is the conventional way for the risk minimization classification. We extend it to the hierarchical classification version by adding all the internal nodes and using the BCPS risk building the cost matrix. Moreover, the flat VGG net, which is used in this article in the multitask hierarchical classification, is carried out in the experiment as a flat baseline against the hierarchical classification models. To comprehensively compare the performance of all the models, we show the results of TIE, HF, and BCPS risk with different λ values on seven datasets (see Table II and Fig. 5). Furthermore, TSS, DARTS, and RMGA are developed under the condition that a local classifier is trained on each nonleaf node in the semantic tree, while TDKL can jointly optimize the tree classifier and the confidence thresholds which are used to stop the samples at the internal nodes. For fair comparison, we use multitask hierarchical output for the first three models to replace the local classifiers in the tree, and utilize the features generated in the last convolutional layer in the multitask

TABLE II
RESULTS OF ALL ALGORITHMS ON SEVEN DATASETS IN TERMS OF TIE AND HF (%). (a) CIFAR 100. (b) PASCAL VOC. (c) ILSVRC 65. (d) STANFORD CARS. (e) CALTECH 256. (f) SUN. (g) IMAGENET

(a)

	TSS	DARTS	TKDL	CSMSE	RMGA	VGG	HSRM
TIE↓	1.853	1.804	1.784	1.758	1.808	1.866	**1.728**
HF↑	78.47	80.22	81.28	84.33	81.97	78.38	**86.10**
Rank	6	5	4	2	3	7	1

(b)

	TSS	DARTS	TKDL	CSMSE	RMGA	VGG	HSRM
TIE↓	1.975	1.898	1.884	1.826	1.891	1.992	**1.798**
HF↑	78.78	81.51	82.29	85.25	82.38	78.66	**87.11**
Rank	6	5	4	2	3	7	1

(c)

	TSS	DARTS	TKDL	CSMSE	RMGA	VGG	HSRM
TIE↓	1.275	1.218	1.187	1.152	1.202	1.298	**1.097**
HF↑	88.69	91.32	92.30	93.91	91.87	88.17	**94.82**
Rank	6	5	3	2	4	7	1

(d)

	TSS	DARTS	TKDL	CSMSE	RMGA	VGG	HSRM
TIE↓	2.895	2.838	2.763	2.884	2.802	2.907	**2.708**
HF↑	70.47	72.28	74.12	71.88	73.07	70.11	**76.92**
Rank	6	4	2	5	3	7	1

(e)

	TSS	DARTS	TKDL	CSMSE	RMGA	VGG	HSRM
TIE↓	1.612	1.578	1.535	1.605	1.519	1.596	**1.475**
HF↑	83.77	86.28	87.41	85.18	88.84	85.49	**90.52**
Rank	7	4	3	6	2	5	1

(f)

	TSS	DARTS	TKDL	CSMSE	RMGA	VGG	HSRM
TIE↓	1.832	1.617	1.591	1.712	1.628	1.692	**1.446**
HF↑	82.87	83.90	84.19	83.15	85.49	83.55	**87.93**
Rank	7	4	3	6	2	5	1

(g)

	TSS	DARTS	TKDL	CSMSE	RMGA	VGG	HSRM
TIE↓	3.287	2.937	2.846	3.051	2.825	2.998	**2.798**
HF↑	71.97	73.42	74.85	71.92	75.28	73.05	**77.40**
Rank	7	4	3	6	2	5	1

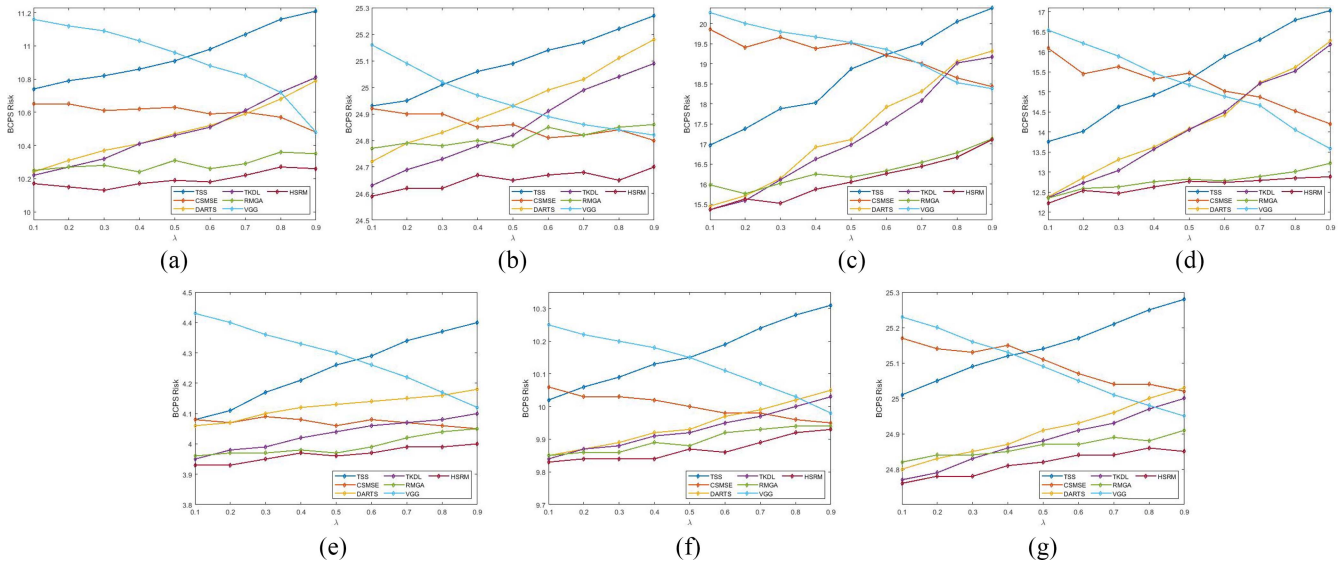


Fig. 5. Results of BCPS risk (%) curve on seven datasets. (a) Caltech 256. (b) ImageNet. (c) PASCAL VOC. (d) Cifar 100. (e) ILSVRC 65. (f) SUN. (g) Stanford Cars.

hierarchical learning module for TDKL. For the cost-sensitive method CSMSE [18], we implement the end-to-end version by using the mean-square error (MSE) loss in the article.

2) Experimental Results: Table II shows the results on all the datasets in terms of TIE and HF, respectively. The proposed HSRM model outperforms others in all the datasets, which demonstrates the effectiveness of the proposed model. Cost-sensitive method CSMSE obtains the second best performance in small scale datasets, such as PASCAL VOC, CIFAR 100, and ILSVRC 65, but it does not perform well comparing with hierarchical methods in larger scale datasets with more classes or deeper tree structure, such as Caltech 256, SUN, and ImageNet. Hierarchical methods, TKDL, RMGA, and DARTS, perform better than CSMSE, and the reasons come from three aspects. First, cost-sensitive learning performs good in data with a few classes but not good in large-scale tasks because of the instability of optimizing the cost matrix [18]. Second, hierarchical classification is good at handling tasks with many classes by dividing a hard problem into several easier subproblems [5]. It is difficult for cost-sensitive training to handle a large number of classes and a deep semantic tree. For example, the ImageNet dataset has 1000 classes originally, but it expands to 1860 classes if the nonleaf nodes of the semantic tree are considered. It also has a 19-layer semantic tree, much deeper than the other datasets. On the other hand, HSRM outperforms other hierarchical methods. State-of-the-art models DARTS and TDKL focus on making predictions more informative provided the predictions are correct. The overall performance is influenced by the level of risk of wrong predictions but not the informativeness of correct samples. TSS and RMGA aim to optimize TIE. TSS selects a threshold at each node based on a limited and static candidate set, where finding the proper threshold is difficult. RMGA optimizes the TIE in a global view using GA which is sensitive to initial guesses of the parameters. By leveraging the DQN, HSRM quickly obtains the solution in a global view, and achieves better performance. Fig. 5 shows the performance of all the models on BCPS risk with various emphases on the conservative and precipitant risks, where the lower curve represents the better performance. We set the λ value to be $\{0.1, 0.2, \ldots, 0.9\}$, and obtain the BCPS risk curve of all the methods. The results demonstrate that HSRM achieves the best or competitive performance across all seven datasets and all λ values. The conventional flat classification model VGG only classifies samples to the leaf nodes whose predictions have high precipitant risk (the value of λ is small), while the hierarchical methods have relatively higher conservative risk than the flat one, so their curves increase in general when λ increases. It is worth noting that HSRM and RMGA, CSMSE are the only three models which can optimize BCPS risk with different λs, so their curves are not monotonically increasing or decreasing. CSMSE treats the hierarchical problem as a flat one and have less power than hierarchical methods to predict samples at the proper internal nodes.

To further explore whether the observed differences are statistically significant, we carry out the Friedman test for multiple comparisons together with the Bonferroni–Dunn post hoc test to identify pairwise differences [41]. In Friedman test, given k compared algorithms and N datasets, let r_i^j be the rank of the jth model on the ith dataset, and $R_i = (1/N)\sum_{i=1}^{N} r_i^j$ be the average rank of model i among all datasets. The null hypothesis of Friedman test is that all the models are equivalent in terms of both of the hierarchical metrics. Under null hypothesis, the Friedman statistic is distributed according to χ_F^2 with $k-1$ degrees of freedom

$$\chi_F^2 = \frac{12N}{k(k+1)}\left(\sum_{i=1}^{k} R_i^2 - \frac{k(k+1)^2}{4}\right)$$
$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \tag{15}$$

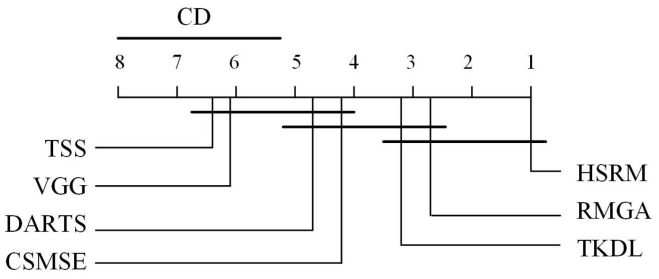


Fig. 6. Hierarchical metrics comparison of different models with the Bonferroni–Dunn test.

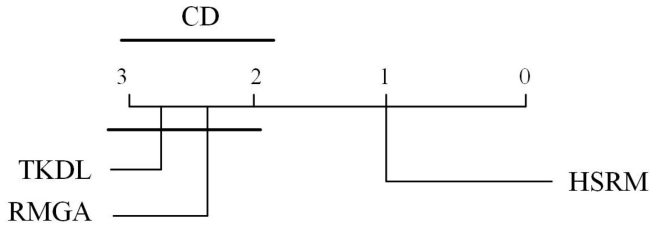


Fig. 7. Hierarchical metrics comparison of three models with the Bonferroni–Dunn test.

where F_F follows a Fisher distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom. The average rank is calculated by the rank across all the datasets in Table II, and the value $F_F = 16.057$ is computed according to (15). With seven models and seven datasets, the critical value for $\alpha = 0.05$ of $F((7-1), (7-1) \times (7-1))$ is 2.3638, so the null hypothesis is rejected. Thus, all the models are not equivalent in terms of both of the hierarchical metrics, and there exist significant differences between them.

The Bonferroni–Dunn post-hoc test is leveraged to detect if the proposed model is better than the existing ones. Specifically, the performance of the two models are significantly different if the distance between the average ranks exceeds the critical distance (CD): $\mathrm{CD}_\alpha = q_\alpha\sqrt{[(k(k+1))/6N]}$, where q_α is given in [41, Table 5]. Note that $q_{0.1} = 2.394$ with $k = 7$, so $\mathrm{CD}_{0.1} = q_{0.1}\sqrt{[(7\times 8)/(6\times 7)]} = 2.750$. Fig. 6 visually shows the CD diagrams in terms of the hierarchical metrics, and the lowest (best) ranks on the axis are to the right. We can conclude that in terms of both of the hierarchical metrics, *HSRM*, *RMGA*, and *TKDL* perform statistically better than the others, but there is no consistent evidence to indicate statistical differences between these three models. We further conduct another Bonferroni–Dunn post-hoc test for the three models on all the datasets. In this case, $\mathrm{CD}_{0.1} = 1.96\sqrt{[(3\times 4)/(6\times 7)]} = 1.048$ with $k = 3$, the results are shown in Fig. 7. The test results vary with the averaged rank of each model across all the datasets. The results of such averaged ranks are the same under both metrics, so the results of Bonferroni–Dunn test, that is, Figs. 6 and 7, are the same. We can see that HSRM is statistically better than the other two models, which verifies the effectiveness of the proposed model.

To gain more insights of the behaviors of the proposed model against other models, we perform a test using samples that are of challenge to typical classification methods.

TABLE III
COMPARISON OF DIFFERENT OPTIMIZATION METHODS. GA IS THE GA APPLIED IN [24], WHILE DL IS THE DEEP REINFORCEMENT LEARNING METHOD USED IN THIS ARTICLE. THE NAME OF THE DATASETS IS SHORTEN FOR SPACE. (a) TIE ↓. (b) HF (%) ↑. (c) WEIGHTED BCPS (%) ↓

(a)

	CIFAR	VOC	I65	Cars	Caltech	SUN	ImgNet
GA	1.786	0.877	1.179	2.765	1.501	1.591	2.821
DL	**1.728**	**1.798**	**1.097**	**2.708**	**1.475**	**1.446**	**2.798**

(b)

	CIFAR	VOC	I65	Cars	Caltech	SUN	ImgNet
GA	82.88	84.43	92.01	73.95	88.97	85.91	75.71
DL	**86.10**	**87.11**	**94.82**	**76.92**	**90.52**	**87.93**	**77.40**

(c)

	CIFAR	VOC	I65	Cars	Caltech	SUN	ImgNet
GA	14.72	18.28	4.69	25.21	10.59	10.15	25.01
DL	**11.08**	**15.14**	**3.95**	**24.65**	**10.08**	**9.81**	**24.60**

TABLE IV
COMPARISON OF DIFFERENT RISK MEASUREMENT. THE NAME OF THE DATASETS IS SHORTEN FOR SPACE. (a) TIE ↓. (b) HF (%) ↑. (c) WEIGHTED BCPS (%) ↓

(a)

	CIFAR	VOC	I65	Cars	Caltech	SUN	ImgNet
TKDL	1.746	1.829	1.130	2.755	1.508	1.533	2.822
HSRM	**1.728**	**1.798**	**1.097**	**2.708**	**1.475**	**1.446**	**2.798**

(b)

	CIFAR	VOC	I65	Cars	Caltech	SUN	ImgNet
TKDL	82.89	84.97	92.77	74.69	88.47	85.56	75.48
HSRM	**86.10**	**87.11**	**94.82**	**76.92**	**90.52**	**87.93**	**77.40**

(c)

	CIFAR	VOC	I65	Cars	Caltech	SUN	ImgNet
TKDL	15.77	18.37	4.38	25.19	10.97	10.28	25.17
HSRM	**11.08**	**15.14**	**3.95**	**24.65**	**10.08**	**9.81**	**24.40**

All the models are trained on the PASCAL VOC dataset. All test samples are classified based on the semantic tree structure of PASCAL VOC. Fig. 8 shows the results by all models. It appears that HSRM provides more risk-minimized predictions with better balance between mistaken and too abstract predictions. Fig. 9 shows pairs of samples of high similarities but belonging to different classes. It is difficult for any classifiers to make accurate predictions with high confidence.

It is thus interesting to observe the behaviors of different risk minimization algorithms on such cases. For example, in the first sample pair, the top one is a bird while the bottom one is an aeroplane that imitates the shape of a bird. HSRM reports them as the least common ancestor node *Objects* to avoid misclassification. The samples in the second pair are not the bicycles and motorbikes we see often, and even humans cannot easily predict them correctly. HSRM gives abstract answer of $2-$ *Wheeled* vehicles. Samples in the last pair look very similar in shape, but the one on the top is a locomotive that is not seen separated from train wagons in the training samples. HSRM predicts it as *Vehicles* to avoid the mistakes.

3) Ablation Study: To explore the influence of different modules, we perform ablation test on the proposed model, in terms of the optimization method and the risk measurement. In all the tests, we use the TIE, HF and the Weighted BCPS

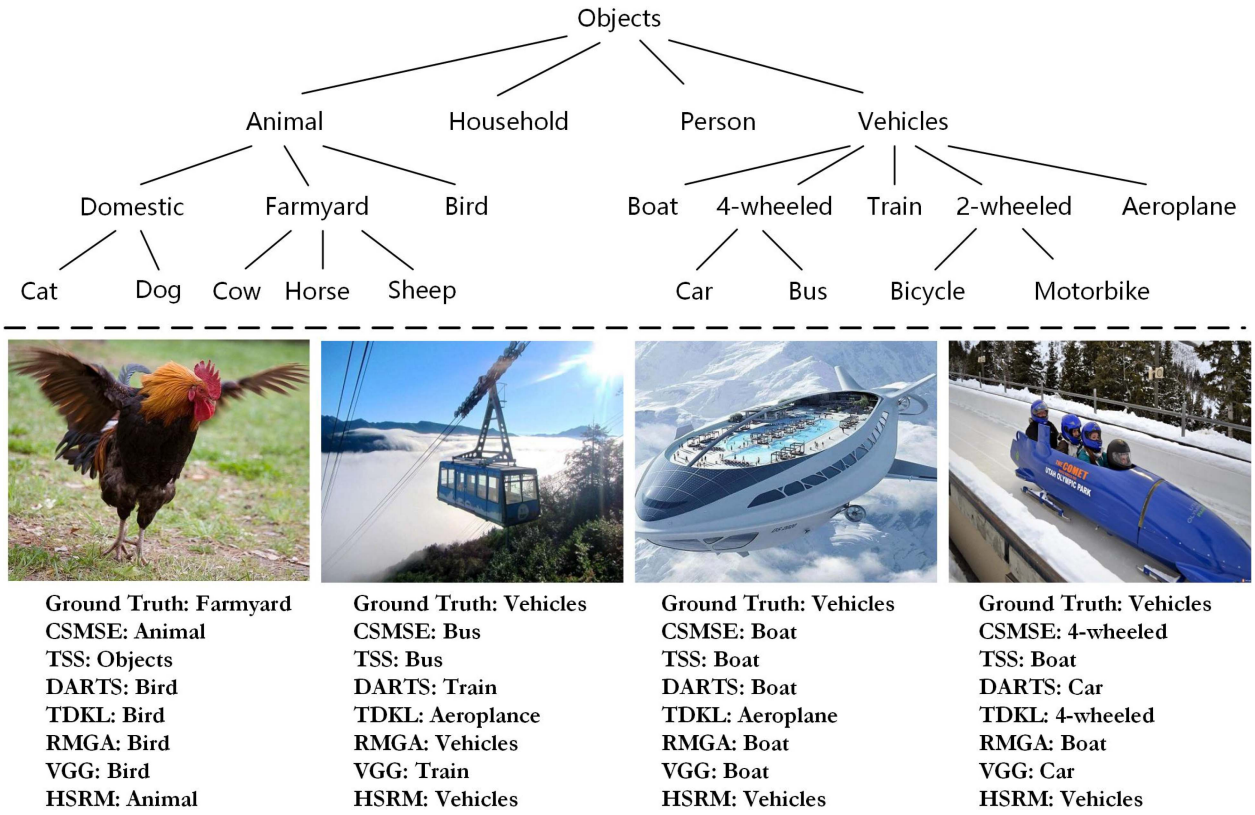


Fig. 8. Predictions on some challenging samples using semantic tree of the PASCAL VOC dataset.

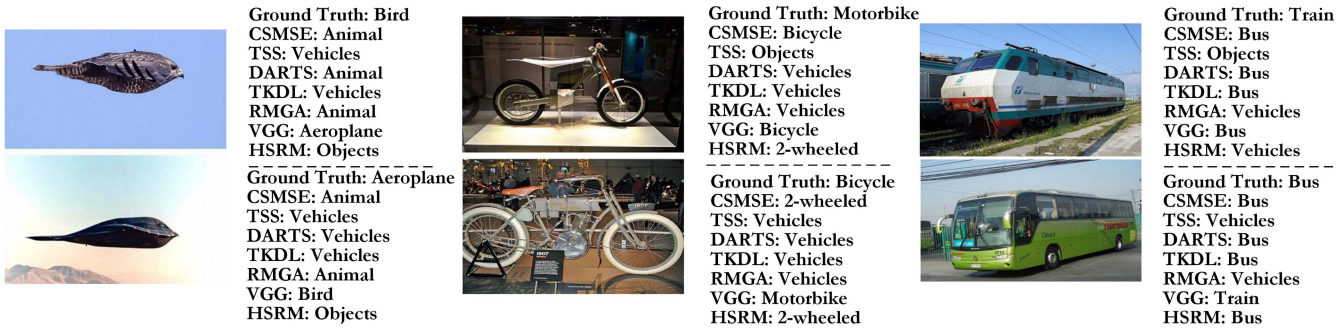


Fig. 9. Predictions on sample pairs of high similarities but belonging to different classes from the PASCAL VOC dataset.

as evaluation metrics, where the Weighted BCPS is the average results of different λ values of $\{0.1, 0.2, \ldots, 0.9\}$ in (5). In the test on optimization methods, we compare the deep reinforcement learning based method (DL) in this article with the GA used in [24] by optimizing the same output of the network, that is, the output in the hierarchical multitask module. The results are shown in Table III and demonstrate that the DL method is better than the GA method over all the datasets. The main reason is that the GA method requires a good initialization to obtain a approximate local optimum [24], while the DL method searches in the solution space more efficiently. In addition, the DL method considers the influence of the low-level decisions by leveraging long-term reward, which is important in the hierarchical classification problem [25]. In the test on risk measurement, we compare the proposed conservative/precipitant risk with the confidence KL divergence of [12], which regards the output distribution of predictions similar to the uniform distribution as high risk predictions. The results can be seen in Table IV and demonstrate that the proposed risk measurement is better than the confidence KL divergence one over all the datasets. The main reason is that the proposed conservative/precipitant risk considers not only the output distribution but also the reliability of the classifier in the sense that the output of an unreliable classifier is not a solid evidence to tell the uncertainty of the prediction.

V. Conclusion

We propose a novel hierarchical classification model to define and minimize the risk in large-scale classification tasks. First, we leverage the semantic hierarchy to define the balanced conservative/precipitant semantic risk. The BCPS risk adjusts the balance between the competing factors of conservative and precipitant risks. Second, we use deep reinforcement learning to solve the nonconvex and underivative optimization

problem of BCPS risk and other hierarchical metrics. We consider the hierarchical classification as a sequence of stopping and going down decisions. Hierarchical multitask learning is designed to obtain the confidence scores of different granularity, which gets rid of the hyperparameters in previous works. Then, these scores are fed to the deep reinforced multigranularity learning network to obtain a global risk-minimized prediction with flexible granularity. By considering various uncertainty factors, the information loss by stopping and uncertainty of going down are computed at each step. By taking into account the long-term rewards, the proposed method demonstrates clear advantages in minimizing the global risks. Experimental results on seven datasets with the semantic tree structure show that the proposed HSRM method achieves superior performance based on all evaluation criteria.

The proposed method have two limitations. For one thing, it currently cannot well addressed multilabel classification scenarios. Although thresholding the output to make the model output multiple predictions is straightforward to handle the multilabel classification problem, the number of predictions varies from different samples, and this makes it difficult to make predictions with a proper number. In the future, a possible way is to learn a powerful deep reinforced agent that can learn to not only make multigranularity prediction but also predict an adequate number of labels. For another, the proposed method cannot handle the DAG structure. Although the semantic risk can be measured by simply computing the risk as the shortest or longest path between the prediction node and ground-truth node for DAG, there is no sensible explanations for some prediction nodes with multiple parents. Also, it is promising to extend the current method to classification problems with other relation structures like graph, where the class relationship could be bidirectional [42].

REFERENCES

[1] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei, "Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 3450–3457.

[2] S. Gopal and Y. Yang, "Hierarchical Bayesian inference and recursive regularization for large-scale classification," *ACM Trans. Knowl. Discov. Data*, vol. 9, no. 3, p. 18, 2015.

[3] T. Zhao *et al.*, "Embedding visual hierarchy with deep networks for large-scale visual recognition," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 4740–4755, Oct. 2018.

[4] H. Zhao, Q. Hu, P. Zhu, Y. Wang, and P. Wang, "A recursive regularization based feature selection framework for hierarchical classification," *IEEE Trans. Knowl. Data Eng.*, early access, Dec. 23, 2019, doi: 10.1109/TKDE.2019.2960251.

[5] C. Freeman, D. Kulić, and O. Basir, "Feature-selected tree-based classification," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1990–2004, Dec. 2013.

[6] H. Zhao, P. Zhu, P. Wang, and Q. Hu, "Hierarchical feature selection with recursive regularization," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, vol. 2017, 2017, pp. 3483–3489.

[7] X. Zhu, X. Li, and S. Zhang, "Block-row sparse multiview multilabel learning for image classification," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 450–461, Feb. 2016.

[8] F. Azhar and C.-T. Li, "Hierarchical relaxed partitioning system for activity recognition," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 784–795, Mar. 2017.

[9] A. Sun, E.-P. Lim, W.-K. Ng, and J. Srivastava, "Blocking reduction strategies in hierarchical text classification," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 10, pp. 1305–1308, Oct. 2004.

[10] M. Ceci and D. Malerba, "Hierarchical classification of HTML documents with WebClassII," in *Proc. Eur. Conf. Inf. Retrieval*, 2003, pp. 57–72.

[11] S. D'Alessio, K. Murray, R. Schiaffino, and A. Kershenbaum, "The effect of using hierarchical classifiers in text categorization," in *Content-Based Multimedia Information Access-Volume 1*, 2000. Seattle, WA, USA: Association for the Advance of Artificial Intelligence, pp. 302–313.

[12] K. Lee, K. Lee, K. Min, Y. Zhang, J. Shin, and H. Lee, "Hierarchical novelty detection for visual object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 1034–1042.

[13] M. Lázaro and A. R. Figueiras-Vidal, "A bayes risk minimization machine for example-dependent cost classification," *IEEE Trans. Cybern.*, early access, May 13, 2019, doi: 10.1109/TCYB.2019.2913572.

[14] G. Elsayed, D. Krishnan, H. Mobahi, K. Regan, and S. Bengio, "Large margin deep networks for classification," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2018, pp. 850–860.

[15] W. Ouyang, H. Zhou, H. Li, Q. Li, J. Yan, and X. Wang, "Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1874–1887, Aug. 2018.

[16] A. Bernstein, F. Provost, and S. Hill, "Toward intelligent assistance for a data mining process: An ontology-based approach for cost-sensitive classification," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 503–518, Apr. 2005.

[17] P. Domingos, "MetaCost: A general method for making classifiers cost-sensitive," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining.*, 1999, pp. 155–164.

[18] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, Aug. 2018.

[19] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. 17th Int. Joint Conf. Artif. Intell.*, vol. 2, 2001, pp. 973–978.

[20] C. Fellbaum, "WordNet: An electronic lexical database," in *Language*, vol. 76. Cambridge, MA, USA: MIT Press, 2000, p. 706.

[21] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[22] M. Yuan and M. Wegkamp, "Classification methods with reject option based on convex risk minimization," *J. Mach. Learn. Res.*, vol. 11, no. 5, pp. 111–130, 2010.

[23] B. Hanczar and E. R. Dougherty, "Classification with reject option in gene expression data," *Bioinformatics*, vol. 24, no. 17, pp. 1889–1895, 2008.

[24] Y. Wang, Q. Hu, Y. Zhou, H. Zhao, Y. Qian, and J. Liang, "Local bayes risk minimization based stopping strategy for hierarchical classification," in *Proc. IEEE Int. Conf. Data Mining*, New Orleans, LA, USA, 2017, pp. 515–524.

[25] M. Ceci and D. Malerba, "Classifying Web documents in a hierarchy of categories: A comprehensive study," *J. Intell. Inf. Syst.*, vol. 28, no. 1, pp. 37–78, 2007.

[26] C. N. S. Jr and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining. Knowl. Discov.*, vol. 22, nos. 1–2, pp. 31–72, 2011.

[27] K. Aris, P. Ioannis, G. Eric, G. Paliouras, and I. Androutsopoulos, "Evaluation measures for hierarchical classification: A unified view and novel approaches," *Data Mining. Knowl. Discov.*, vol. 29, no. 3, pp. 820–865, 2015.

[28] O. Dekel, J. Keshet, and Y. Singer, "Large margin hierarchical classification," in *Proc. Int. Conf. Mach. Learn.*, 2004, pp. 27–35.

[29] D. Koller and M. Sahami, "Hierarchically classifying documents using very few words," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, 1997, pp. 170–178.

[30] K. Ma, W. Liu, K. Zhang, Z. Duanmu, Z. Wang, and W. Zuo, "End-to-end blind image quality assessment using deep neural networks," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1202–1213, Mar. 2018.

[31] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[32] E. Conti, V. Madhavan, F. P. Such, J. Lehman, K. O. Stanley, and J. Clune, "Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2018, pp. 5032–5043.

[33] Y. Mao, J. Tian, J. Han, and X. Ren, *End-to-End Hierarchical Text Classification With Label Assignment Policy*," OpenReview, Mountain View, CA, USA, 2018.
[34] Y. Mao, X. Ren, J. Shen, X. Gu, and J. Han, "End-to-end reinforcement learning for automatic taxonomy induction," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguist.*, 2018, pp. 2462–2472.
[35] T. Hester *et al.*, "Deep Q-learning from demonstrations," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3223–3230.
[36] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
[37] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Sydney, NSW, Australia, 2014, pp. 554–561.
[38] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.
[39] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," Division of Biology and Biological Engineering, California Inst. Technol., Pasadena, CA, USA, Rep. 7694, 2007.
[40] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 3485–3492.
[41] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 1–30, 2006.
[42] R. Jin, Y. Dou, Y. Wang, and X. Niu, "Confusion graph: Detecting confusion communities in large scale image classification," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 1980–1986.

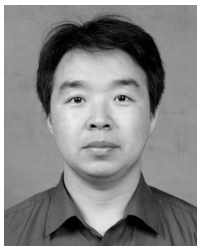
Yu Wang received the B.S. degree in communication engineering, the M.S. degree in software engineering, and the Ph.D. degree in computer applications and techniques from Tianjin University, Tianjin, China, in 2013, 2016, and 2021, respectively.

He is currently an Assistant Professor with Tianjin University. He was a Visitor Scholar with the University of Waterloo, Waterloo, ON, Canada, in 2019. He has published many peer-reviewed papers in world-class conferences and journals, such as the IEEE International Conference on Data Mining, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and IEEE TRANSACTIONS ON FUZZY SYSTEMS. His research interests focus on hierarchical learning, open-set recognition, incremental learning, and data mining and machine learning on various computer vision and industrial applications.

Zhou Wang (Fellow, IEEE) received the Ph.D. degree from the University of Texas at Austin, Austin, TX, USA, in 2001.

He is currently a Professor and the University Research Chair with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He has more than 200 publications in these fields with over 50 000 citations (Google Scholar). His research interests include image and video processing and coding, visual quality assessment and optimization, computational vision and pattern analysis, multimedia communications, and biomedical signal processing.

Prof. Wang is a recipient of the 2017 Faculty of Engineering Research Excellence Award at the University of Waterloo, the 2016 IEEE Signal Processing Society Sustained Impact Paper Award, the 2015 Primetime Engineering Emmy Award, the 2014 NSERC E.W.R. Steacie Memorial Fellowship Award, the 2013 *IEEE Signal Processing Magazine* Best Paper Award, and the 2009 IEEE Signal Processing Society Best Paper Award. He has been serving as a Senior Area Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING since 2015, and an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY since 2016. He is a Fellow of the Canadian Academy of Engineering, a Elected Fellow of the Royal Society of Canada-Academy of Science, and a Tier 1 Canada Research Chair.



Qinghua Hu (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively.

He was a Postdoctoral Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, from 2009 to 2011. He is currently the Dean of the School of Artificial Intelligence, the Vice Chairman of the Tianjin Branch of China Computer Federation, the Vice Director of the SIG Granular Computing and Knowledge Discovery, and the Chinese Association of Artificial Intelligence. He is currently supported by the Key Program, National Natural Science Foundation of China. He has published over 200 peer-reviewed papers. His current research is focused on uncertainty modeling in big data, machine learning with multimodality data, and intelligent unmanned systems.

Dr. Hu is an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Acta Automatica Sinica*, and *Energies*.

Yucan Zhou received the Ph.D. degree from the College of Artificial Intelligence, Tianjin University, Tianjin, China, in 2019.

She is currently an Assistant Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. Her research interests include artificial intelligence, deep learning, long-tail distribution learning, and hierarchical classification.

Honglei Su (Member, IEEE) received the B.A.Sc. degree from the Shandong University of Science and Technology, Qingdao, China, in 2008, and the Ph.D. degree from Xidian University, Xi'an, China, in 2014.

Since 2014, he has been working as an Assistant Professor with the School of Electronic Information, Qingdao University, Qingdao. From 2018 to 2019, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include perceptual image processing, immersive media processing, and computer vision.