Label Hierarchy Transition: Modeling
Class Hierarchies to Enhance Deep Classifiers

Renzhen Wang
Xi’an Jiaotong University
wrzhen@stu.xjtu.edu.cn

De Cai
Tencent
daviddecai@tencent.com

Kaiwen Xiao
Tencent
loktarxiao@tencent.com

Xixi Jia
Xidian University
hsijiaxidian@gmail.com

Xiao Han*
Tencent
haroldhan@tencent.com

Deyu Meng*
Xi’an Jiaotong University
dymeng@mail.xjtu.edu.cn

Abstract

Hierarchical classification aims to sort the object into a hierarchy of categories. For example, a bird can be categorized according to a three-level hierarchy of order, family, and species. Existing methods commonly address hierarchical classification by decoupling it into several multi-class classification tasks. However, such a multi-task learning strategy fails to fully exploit the correlation among various categories across different hierarchies. In this paper, we propose Label Hierarchy Transition, a unified probabilistic framework based on deep learning, to address hierarchical classification. Specifically, we explicitly learn the label hierarchy transition matrices, whose column vectors represent the conditional label distributions of classes between two adjacent hierarchies and could be capable of encoding the correlation embedded in class hierarchies. We further propose a confusion loss, which encourages the classification network to learn the correlation across different label hierarchies during training. The proposed framework can be adapted to any existing deep network with only minor modifications. We experiment with three public benchmark datasets with various class hierarchies, and the results demonstrate the superiority of our approach beyond the prior arts. Source code will be made publicly available.

1. Introduction

Image classification is a classic computer vision task that aims to predict a category label for each image. Recent years have witnessed the tremendous success of image classification due to the rapid development of deep learning [20], especially for deep classification networks [17, 19, 30]. Conventionally, a classification network is trained with cross entropy loss on one-hot labels, such that it predicts almost orthogonal category labels for images in different classes. Such a scheme focuses on the differences of categories, while it fails to capture inter-class similarity [26, 33]. Hierarchical classification, contrast to classic image classification, attempts to classify the object into a hierarchy of categories, with each hierarchy represents one specific level of concept abstraction. For example, birds can be categorized into a three-level hierarchy of order, family and species according to their taxonomic characters, as shown in Fig. 1. In hierarchical classification learning, the labels can be organized according to the domain knowledge (e.g, WordNet database [25]) or automatically generated by algorithms [1, 21, 24, 31]. Compared to traditional multi-class classification, hierarchical classification pays more attention to exploit rich semantic correlation among categories [8], in order to improve the performance at different hierarchies and mitigate the severity of prediction mistakes, i.e., making the incorrectly classified samples fall within semantically related categories [3, 37]. Take the example of making diagnosis for one malignant tumor, mistaking it as another malignant one in the same subtype is less serious than mistaking it as one benign tumor, as such a mistake would have crucial implications in terms of the follow-up

Figure 1. A typical hierarchical classification scenario, where a bird can be categorized into a three-level hierarchy of species, family, and order. The objects within the same hierarchy commonly share rich semantic correlation.

*Corresponding author
Hierarchical classification has been proved effective in many visual scenarios, which can be traced back to traditional linear classifiers [13, 22, 40, 41]. Recently, hierarchical classification has attracted vast attention in deep learning [9]. Most of the existing works [4–6, 37, 38] commonly fall into a multi-task learning manner. For example, a deep network was proposed in [38] to classify coarse- and fine-grained categories through two different branches. In [37], a granular network was proposed for food image classification, where a single network backbone was shared by multiple fully-connected layers with each one responsible for the label prediction within one hierarchy. Chen et al. [6] likewise employed a multi-head architecture to level-wisely output the prediction, but instead introduced an attention mechanism to guide label prediction of fine-grained hierarchy through that of the coarse-grained hierarchy. Furthermore, Alsallakh et al. [4] used a standard deep architecture to fit the finest-grained labels, and added branches at the intermediate layers to fit the coarser-grained labels. All the above methods adopt a divide-and-conquer strategy to model the class hierarchies, in which the semantic correlation of labels are implicitly captured by the gradient back propagation of the multi-task loss, e.g., the sum of cross entropy losses of each hierarchy. Recently, Chang et al. [5] pointed out that fine-grained features benefit coarse-grained classifier learning, yet coarse-grained label prediction is harmful to fine-grained feature learning. They in turn proposed to disentangle coarse- and fine-level features by leveraging level-specific classification heads, and further force finer-grained features (with stopped gradient) to participate in coarser-grained label predictions. Such a simple method has achieved significant improvement in fine-grained image recognition, which, however, does not explicitly make use of the class hierarchies. Moreover, the cross entropy loss with one-hot labels used in these existing methods [4–6, 37, 38] incline to produce orthogonal predictions at each hierarchy and fail to depict the correlation among different categories.

In this paper, we develop a deep hierarchical classification method that explicitly captures the semantic correlation embedded in class hierarchies. Inspired by the hierarchical labeling process of human [5] that an accurate annotation usually depends on the image itself and the label at the adjacent hierarchy, we propose Label Hierarchy Transition (LHT), a unified probabilistic approach, to recursively predict the labels from fine- to coarse-grained class hierarchies. Concretely, the LHT model contains two components: (1) an arbitrary existing classification network, which is used to predict the labels for the finest-grained hierarchy; (2) a transition network, which generates a label hierarchy transition matrix for each hierarchy (except for the finest-grained one). Each element of the transition matrix denotes the conditional probability of one class at the corresponding hierarchy given the other class of the adjacent finer-grained hierarchy. For each coarser-grained hierarchy, the label prediction can be formulated as the product of its corresponding transition matrix and the prediction score of its adjacent finer-grained hierarchy. To address the over-confident prediction problem introduced by the cross entropy loss, we further propose a confusion loss that explicitly regularizes the label hierarchy transition matrices, and encourages the network to learn semantic correlation from all class hierarchies during training. The proposed framework can be adapted to any existing deep networks with only minor modifications. To summary, the main contributions are three-fold:

- We propose a unified deep classification framework to address the hierarchical classification problem. Contrast to predicting the labels for each hierarchy, we propose to learn label hierarchy transition matrices, which encode the likelihood of similarity between classes across different label hierarchies. To the best of our knowledge, we are the first to leverage deep networks to explicitly learn the semantic correlation embedded in class hierarchies.

- We propose a confusion loss to directly regularize the negative entropy of the column space of label hierarchy transition matrices, which further encourages the network to learn the correlation among class hierarchies, and alleviates the over-confident prediction arisen by cross entropy loss.

- Our LHT method can be applied to any existing classification network, and the resulting model can be trained in an end-to-end manner. Extensive experiments on three public datasets demonstrate the superiority of our approach beyond the prior arts.

2. Related work

Exploiting hierarchical structure in classification task has been extensively explored [29, 34], which is mainly related to three lines of research: hierarchical architecture, hierarchical loss function and label embedding based methods. For hierarchical architecture, we have detailed the works based on deep learning, and only traditional classifiers [2, 11, 14, 15, 22] are discussed here. These methods mainly focus on the problem that the linear classifiers suffer a heavy computational burden when the number of recognition categories become huge. The core idea was to create a classifier tree whose leaf nodes were used for the prediction of the given categories and internal nodes for the cluster of categories. Contrary to these works, our LHT model predicts labels for each hierarchy by deep networks.
A range of literature on hierarchical classification resorts to incorporating the class hierarchies in the loss function. In [35], the metrics associated with each node of the taxonomy tree were learned based on a probabilistic nearest-neighbour classification framework. In [32], the class hierarchies with a tree structure were used to impose prior on the parameters of the classification layer for knowledge transfer between classes, and similar strategy was proposed by Zhao et al. [40] in the traditional linear classifier. In [9], various label relations (e.g., mutual exclusion and subsumption) were encoded as conditional random field to pair-wisely capture the correlation between classes. More recently, Bertinetto et al. [12] proposed a hierarchical loss that could be viewed as a weighted version of the plain cross entropy loss, and the weights were defined according to the depth of the label tree. Our LHT model introduces a novel loss that combines the traditional cross entropy loss with the proposed confusion loss. Different from the aforementioned methods, we aim to regularize the label hierarchy transition matrices output by our LHT model, such that class correlation from all hierarchies can be fully used to improve the classification.

Label embedding aims to encode the correlation information among classes into the traditional one-hot labels, which has been widely explored in multi-class classification [7, 16, 26, 33]. In hierarchical classification, Bertinetto et al. [3] proposed a soft-label-based method to soften the one-hot labels according to the metric defined by the label tree. Dhall et al. [12] employed entailment cones to learn order preserving embedding, which can be embedded in both Euclidean and hyperbolic geometry enforces transitivity among class hierarchies. Although label embedding is a potential strategy to capture the correlation across class hierarchies, in this paper, we attempt to develop a unified deep classification framework for hierarchical classification.

3. Methodology

In hierarchical classification tasks, a given image $x$ is usually annotated with a chain of $K$ labels $(y^1, y^2, \ldots, y^K)$, where $K$ is the number of label hierarchies, and each $y^k \in \{1, 2, \ldots, C_k\}$ is the label of $k$-th hierarchy and $C_k$ indicates the number of categories within this hierarchy. We denote $\mathbf{y}^k$ is the one-hot vector of $y^k$. Without loss of generality, we assume that the label granularity becomes coarse as $k$ increases, i.e., $C_k > C_{k+1}$.

In this paper, our goal is to propose a unified probabilistic framework to improve the accuracy of label prediction at different class hierarchies. In the following subsections, we first formulate our LHT framework that explicitly encodes the semantic correlation among hierarchies, and implement it with a deep architecture, as shown in Fig. 2. We then introduce the proposed confusion loss to encourage correlation learning across hierarchies.

3.1. Label Hierarchy Transition

The problem of interest is to directly learn the unknown classification distribution $p(y^1, \ldots, y^K | x)$. We first revisit the traditional multi-task based methods in hierarchical classification.
classification. Existing methods commonly decouple it into several multi-class classification tasks, where the label of each hierarchy is predicted by typical loss functions (e.g., cross-entropy). This is equivalent to solve the negative log-likelihood of the following equation:\footnote{Please refer to the supplementary materials (Sec. A) for detailed derivation.}:

\[
p(y^1, \cdots, y^K | x) = \prod_{k=1}^{K} p(y^k | x).
\]

(1)

This equation implies that all class hierarchies are statistically independent to each other, which ignores the correlation across different class hierarchies. Moreover, the cross entropy loss with one-hot labels used in Eq. (9) inclines to overemphasize on the differences of different categories but fail to depict the inter-class similarities in a specific hierarchy. Thus, the semantic correlation within the same hierarchy are easily ignored.

To address the aforementioned problem, we propose to explicitly encode the correlation across hierarchies into the existing multi-task learning framework. We assume that one label hierarchy \(y^k\) is statistically dependent on \(y^{k-1}\) when \(x\) is given, and the distribution \(p(y^k | x)\) can be rewritten as:

\[
p(y^k | x) = \sum_{y^{k-1}=1}^{C_{k-1}} p(y^k | y^{k-1}, x) \cdot p(y^{k-1} | x), k = 2, \cdots, K,
\]

(2)

where \(p(y^{k-1} | x)\) is the label distribution of \((k-1)\)-th hierarchy, and \(p(y^k | y^{k-1}, x)\) describes the transition probability from \((k-1)\)-th to \(k\)-th hierarchy, which can be viewed as the hierarchical labeling process for \(k\)-th hierarchy according to the context of image itself and the granularity information of labeling expertise. In fact, with Eq. (2), we readily define a fine-to-coarse learning curriculum, from which the coarser-level labels are gradually learned conditioned on the input image and its adjacent fine-grained level of labels. On this count, the finest-level labels should be firstly fit, which can be achieved by any existing multi-class classifier. For simplicity, we here denote \(t^k_{ij}(x) := p(y^k = i | y^{k-1} = j; x)\), and then we have a \(C_k \times C_{k-1}\) matrix \(\mathbb{T}^k(x) = (t^k_{ij}(x))\). We call this matrix the label hierarchy transition matrix, as it defines the probability of a fine-grained label clustered to a coarse-grained label and we can get a glimpse of the transition status between the adjacent class hierarchies.

In this paper, the core of our Label Hierarchy Transition model is to jointly learn the two kinds of conditional probability involved in Eq. (2) for hierarchical classification, using a unified deep learning framework. As shown in Fig. 2, the framework mainly consists of two components: (i) Classification Network. A plain deep classification network \(\mathbb{P}^k_x(x)\) with parameters \(w\), e.g., ResNet-50, which is used to predict the finest-level label (one-hot vector of \(y^1\)) of the image \(x\); (ii) Transition Network. A lightweight network \(T^k(x)\) with parameters \(\theta\) that generates the label hierarchy transition matrices \(\{\mathbb{T}^k(x)\}_{k=2}^{K}\) from the high-level embedding of the input \(x\) to approximate the inherent transition matrices \(\{\mathbb{T}^k(x)\}_{k=2}^{K}\). It is natural and practical to learn the matrices from the high-level embedding of \(x\), as the high-level embedding is the cue to abstract \(y^{k-1}\). Specifically, the network contains \((K - 1)\) parallel fully-connected layers, where the \((k - 1)\)-th layer outputs a \((C_k \times C_{k-1})\)-dim vector. The resulting vector is reshaped to a \(C_k \times C_{k-1}\) matrix, and imposed a column-wise normalization (via softmax activation function) for satisfying \(\sum_{i=1}^{C_k} T^k_{ij} = 1\).

With the learned \(\mathbb{P}_x^k\) and \(T^k(x)\), the predictions of \(k\)-th label hierarchy are computed by

\[
\hat{p}^k = \mathbb{P}^k(x)T^k(x)p^{k-1}, k = 2, \cdots, K.
\]

(3)

Remark 1: From the given hierarchical labels, we can obtain a set of naive transition matrices \(\mathbb{T}^k\), where only one element is 1 in each column, corresponding to the deterministic state from fine to coarse hierarchy, i.e., \(\mathbb{T}^k_{i,j} = 1\) if category \(j\) at hierarchy \(k - 1\) is a sub-class of category \(i\) at hierarchy \(k\), otherwise \(\mathbb{T}^k_{i,j} = 0\). For example, in Fig. 1, \(\mathbb{T}^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}\). Instead, in Eq. (2), we aims to model \(p(y^k | y^{k-1}, x)\), which is a conditional label distribution considering the uncertainty from image context and hierarchy prior. In fact, if we replace \(\mathbb{T}^k\) with \(\hat{T}^k\) in Eq. (3), the model will degenerate into only fitting the finest-grained labels and it ignores the semantic correlation implied into class hierarchies, as the following theorem states.

**Theorem 1.** Let \(\mathbb{T}^k\) be the naive transition matrix with element \(\mathbb{T}^k_{i,j} = 1\) if category \(j\) at hierarchy \(k - 1\) is a sub-class of category \(i\) at hierarchy \(k\) else \(\mathbb{T}^k_{i,j} = 0\), such that \(\mathbb{T}^k y^{k-1} = y^k\), and the hierarchical cross-entropy loss

\[
\mathcal{L}_{CE} = \sum_{i=1}^{K} \sum_{k=1}^{K} \text{CE}(\hat{p}^k_i, y^k_i),
\]

(4)

where \(\hat{p}^k = \overline{\mathbb{T}^k} \hat{p}^{k-1}, k = 2, \cdots, K\), then we have that minimizing the loss function in Eq. (4) results in a trivial solution. To be more specific, the problem (4) has exactly the same solution as the following optimization problem

\[
\min_w \sum_{i=1}^{N} \text{CE}(\hat{p}^1_i(x_i), y^1_i).
\]

(5)

The proof of the theorem is provided in the supplementary materials, Sec. B.
Remark 2: In Eq. (2), we construct the transition probability in a fine-to-coarse order by first fitting the finest-level labels (dubbed LHT_{f2c}). Similarly, we can also learn the transition probability in a coarse-to-fine order, by which the coarsest-level labels should be firstly predicted by a multi-class classification network (dubbed LHT_{c2f}). We will explore the pros and cons of the two strategies in the experiment section.

3.2. Confusion Loss

We herein describe the optimization of the parameters involved in our approach. From Eq. (3), we get the predictions of the labels at each hierarchy, such that the parameters involved in classification network and transition network, i.e., \( w \) and \( \theta \) can be jointly optimized by the cross-entropy loss imposed at each hierarchy. Given a batch of training data \( \{x_i, y_i^1, y_i^2, \ldots, y_i^K\}_{i=1}^N \), where \( N \) is batch-size, the resulting loss function can be formulated as

\[
L_{CE} = \sum_{i=1}^{N} \sum_{k=1}^{K} CE(\hat{p}_i^k, y_i^k)
\]

\[
= \sum_{i=1}^{N} \left( CE(\hat{p}_i^1(x_i), y_i^1) + \sum_{k=2}^{K} CE(\hat{T}^k(x_i)\hat{p}_i^{k-1}, y_i^k) \right)
\]

As previously discussed, the proposed transition matrices play the roles to recursively predict the label distribution in a fine-to-coarse order, which to some extent imposes the correlation between class hierarchies. However, the network training by minimizing Eq. (11) remains the following challenges: (i) The manually annotated labels within each hierarchy are one-hot, and such a cross entropy loss encourages the network to move all categories toward orthogonality at each hierarchy, counteracting the inter-class similarity introduced by class hierarchies; (ii) From the perspective of numerical optimization, there are infinite combinations of \( \mathcal{T}_G(x) \) and Classification Network \( \hat{p}_i^1(x) \), such that \( \hat{p}_i^2 \) computed by Eq. (3) can perfectly match the given label distribution \( y_i^2 \) for any given image \( x \).

To address the aforementioned challenges, we further introduce an additional regularization loss on transition matrices \( \{\mathcal{T}^k(x)\}_{k=2}^K \), which we call the confusion loss \( L_{Conf} \). We herein revisit the label hierarchy transition matrices involved in our LHT model. The column vector (e.g., the \( j \)-th one) of transition matrix \( \mathcal{T}^k(x) \) represents the label distribution of \( k \)-th hierarchy when the sample is labeled as class \( j \) at \( (k-1) \)-th hierarchy. Each element of this vector denotes the likelihood of similarity, or confusion, between two classes across two adjacent hierarchies. The proposed confusion loss aims to encourage such class confusion to counter the over-confident prediction brought about by the cross entropy loss. Mathematically, the confusion loss is defined as the negative entropy of the column space spanned by the transition matrices, i.e.,

\[
L_{Conf} = \sum_{i=1}^{N} \sum_{k=2}^{K} \sum_{j=1}^{C_{k-1}} \frac{1}{C_{k-1}} \left( \mathcal{T}^j_k(x_i), \log(\hat{T}^k_j(x_i)) \right)
\]

where \( \mathcal{T}^j_k(x_i) \) denotes \( j \)-th column of \( \mathcal{T}^k(x_i) \), \( \log(\cdot) \) is element-wise logarithm operation for vectors, and \( \langle \cdot, \cdot \rangle \) is the inner product operation of vectors. Through minimization of Eq. (7), the conditional distribution represented by each column of \( \mathcal{T}^k(x_i) \) is encouraged to have a higher entropy (i.e., this distribution inclines to uniform distribution). In other words, this introduces confusion in clustering a fine-grained label to a coarse-grained one during training, and eventually softens the prediction scores obtained by Eq. (3). Thus, the confusion loss can be used to alleviate the over-confident prediction brought about by cross entropy loss.

Combined Eq. (11) and Eq. (7), the overall training loss can be formulated as

\[
L = L_{CE} + \lambda L_{Conf},
\]

where \( \lambda \) is hyper-parameter for trade-off between the cross entropy term \( L_{CE} \) and the confusion regulation term \( L_{Conf} \). All the parameters involved in our LHT can be trained in an end-to-end manner. The training algorithm is summarized in Algorithm 1.

4. Experimental Results and Analysis

4.1. Datasets and Evaluation Criterion

In our experiments, we evaluate our method on three public benchmark datasets: CUB-200-2011 [36], FGVC-Aircraft [23] and Stanford Cars [18], where the label hierarchies are publicly available or constructed by Chang et al. [5] according to their lexical relationships in Wikipedia. (i) CUB-200-2011 is a dataset of bird images that has mainly...
been introduced for the problem of fine-grained visual categorisation methods, and contains 11,877 images belonging to 200 bird species. For hierarchical classification evaluation, the labels are re-organized into three-level hierarchy with 13 orders, 38 families and 200 species by tracing their biological taxonomy in [5]. For example, the original label “Brewer Blackbird” is spanned into (“Passeriformes”, “Icteridae”, “Brewer Blackbird”) in the order of (order, family, species). (ii) FGVC-Aircraft is a set of 10,000 images across 100 airplane classes, where the data ratio between training and test set is about 2:1. As a three-level label hierarchy with 30 makers (e.g., “Boeing”), 70 families (e.g., “Boeing 767”) and 100 models (e.g., “767-200”) are accessible in this dataset, it can readily be used in our experiments without modifications. (iii) Stanford Cars contains 8,144 training and 8,041 test images across 196 car classes, and two-level label hierarchy, i.e., 9 car types and 196 car classes, is introduced by Chang et al. [5]. For example, the class label “BMW X6 SUV 2012” is spanned into (“SUV”, “BMW X6 SUV 2012”). We use the standard public train/test splits, and not any bounding box/part annotations are involved in all our experiments.

We adopt the same evaluation metrics as Chang et al. [5] to evaluate our method. Specifically, acc (the percentage of images whose labels are correctly classified) is reported at each class hierarchy, and then acc_avg (the mean of acc across all class hierarchies) is calculated to measure the classification performance at a hierarchical level. Note that the acc results are reported in the form of mean and standard deviation over five runs of each experiment.

### 4.3. Quantitative Results

We compare the proposed method with the following baselines and state-of-the-art methods: (i) Vanilla: it adopts the baseline method with one head to predict the label at the finest-grained hierarchy, and backprop through the class hierarchies to obtain other coarse-grained categories. (ii) Vanilla_single: the baseline model containing one single shared network backbone and multiple classification heads for per-level label prediction. (iii) Vanilla_multi: except for independent classification heads, each level of label has its own network backbone. (iv) FGoN: the current art proposed in [5] where level-specific classification heads are employed to disentangle different levels of features and finer-grained features (with stopped gradients) are allowed to participate in coarser-grained label predictions.

Table 1 reports the results of various methods on CUB-200-2011 and FGVC-Aircraft datasets with three-level label hierarchies. It can be observed that: (i) For each class hierarchy, the proposed method consistently attains the best performance among all comparison methods. Especially compared with the Vanilla model (only trained with the finest-grained hierarchy of labels), we achieve a significant improvement, which demonstrates that our LHT can be effective to solve multi-class recognition by leveraging the prior of class hierarchies. (ii) Vanilla_single that adopts a plain multi-task learning strategy to address hierarchical classification, results in a performance drop of 0.46% in

| Methods          | CUB-200-2011 | FGVC-Aircraft |
|------------------|--------------|---------------|
|                  | order_acc    | family_acc    | species_acc | avg_acc   | maker_acc | family_acc | model_acc | avg_acc   |
| Vanilla          | 95.54±0.13   | 88.21±0.20    | 74.54±0.44  | 86.10     | 92.99±0.20| 89.72±0.21| 85.41±0.35| 89.37     |
| Vanilla_single   | 94.92±0.80   | 87.58±1.11    | 74.42±0.60  | 85.64     | 93.84±0.18| 91.68±0.16| 86.36±0.50| 90.63     |
| Vanilla_multi    | 96.44±0.35   | 89.51±0.15    | 77.42±0.39  | 87.79     | 93.65±0.29| 92.46±0.15| 87.62±0.25| 91.24     |
| FGoN             | 96.28±0.24   | 90.16±0.28    | 77.98±0.19  | 88.14     | 94.05±0.29| 91.92±0.34| 88.07±0.32| 91.35     |
| FGoN*            | 96.37±0.16   | 90.39±0.15    | 77.95±0.04  | 88.24     | 93.04±0.25| 90.73±0.19| 88.35±0.18| 90.71     |
| LHT_c2f (ours)   | 97.49±0.12   | 91.70±0.15    | 78.13±0.29  | 89.11     | 95.42±0.30| 92.80±0.29| 88.52±0.36| 92.25     |
| LHT_f2c (ours)   | 98.19±0.10   | 92.92±0.12    | 79.29±0.18  | 90.13     | 95.73±0.20| 92.89±0.19| 88.56±0.29| 92.39     |

Table 1. Comparison results (mean±std of five runs) on CUB-200-2011 and FGVC-Aircraft datasets. The best results are highlighted in bold. * indicates the results reported in the original paper.
Figure 3. Per class performance comparison between our LHT and the state-of-the-art method FGoN within different class hierarchies on the CUB-200-2011 dataset. X-axis: class index. Y-axis: per class classification accuracy improvement relative to the plain baseline. Ours significantly improves the performance with an increase of the label granularity.

| Methods       | maker_acc | model_acc | avg_acc |
|---------------|-----------|-----------|---------|
| Vanilla       | 94.60±0.09 | 86.49±0.25 | 90.54   |
| Vanilla_single| 95.28±0.24 | 89.13±0.47 | 92.21   |
| Vanilla_multi | 93.28±0.31 | 89.94±0.22 | 91.61   |
| FGoN          | 95.37±0.16 | 89.83±0.16 | 92.60   |
| FGoN*         | 95.58±0.06 | 89.66±0.16 | 92.02   |
| LHT_c2f (ours)| 95.95±0.10 | 89.08±0.18 | 92.52   |
| LHT_f2c (ours)| 96.74±0.09 | 89.67±0.24 | 93.21   |

Table 2. Comparison results (mean±std of five runs) on Stanford Cars dataset. The best results are highlighted in **bold**. * indicates the results reported in the original paper.

We conduct an ablation study on CUB-200-2011 dataset, and the results are summarized in Tab. 3.

The role of label hierarchy: We first study the effect of the class hierarchies on the finest-grained label fitting. First, we remove one level of class hierarchy, and the resulting models trained with two-level hierarchies are denoted as w/o Family and w/o Order. For w/o Family, the family hierarchy is removed during training, and the prediction scores of family hierarchy are backtracked through the hierarchy. The result shows that an evident performance drop over avg_acc is mainly due to the performance degradation of family hierarchy. The similar findings can be found for w/o Order. Note that both the two models perform better than Vanilla model, where no class hierarchies are exploited. This tells...
Table 3. The results (mean±std of five runs) of ablation analysis on CUB-200-2011 dataset.

| Methods        | order_acc | family_acc | specie_acc | avg_acc |
|----------------|-----------|------------|------------|---------|
| Vanilla        | 95.54±0.13| 88.21±0.20 | 74.54±0.44 | 86.10   |
| w/o Family     | 97.71±0.15| 92.15±0.19 | 79.28±0.14 | 89.71   |
| w/o Order      | 97.34±0.13| 92.25±0.20 | 79.17±0.32 | 89.59   |
| w/ Random      | 89.17±0.65| 80.42±0.27 | 78.73±0.24 | 82.77   |
| Ours           | 98.19±0.10| 92.92±0.12 | 79.29±0.18 | 90.13   |
| w/o $L_{conf}$ | 97.62±0.13| 91.60±0.24 | 77.53±0.25 | 88.92   |
| FGoN (input-size=448) | 98.60±0.04| 95.53±0.25 | 86.37±0.23 | 93.50   |
| Ours (input-size=448) | 99.04±0.05| 96.25±0.08 | 86.74±0.19 | 94.01   |

Figure 4. Visualization of the activation obtained by three network branches, with each one corresponding to a specific hierarchy. The columns from left to right are the original image, order, family, and species, respectively.

The role of confusion loss: We further conduct ablation analysis on the proposed confusion loss. Once we remove it from the Eq. (8), the resulting model (w/o $L_{conf}$) consistently drops at all hierarchies yet still outperforms the Vanilla model. This, on the one hand, verifies the positive effect of the confusion loss on capturing the semantic correlation within class hierarchies, and on the other hand verifies the effectiveness of predicting the labels at all hierarchies through the proposed LHT approach.

The role of image size of input: Following the setup in [5], we study the effectiveness of our method on the traditional fine-grained image recognition setting, i.e., just resize the image size of input as 448 × 448 and keep other implementation settings unchanged. The results in Tab. 3 show that a large image size can efficiently benefit fine-grained image recognition.

Visualization of activation maps: In Fig. 4, we further employ Grad-Cam [28] to visualize the activation maps for three typical examples. These objects belong to the same order (i.e., Suliforme), two families (i.e., Phalacrocoracid and Fregatida), and three species (i.e., Brandt’s Cormorant, Pelagic Cormorant, and Frigatebird). It can be observed that (i) the order branch aims to activate the entire bodies of the three birds; (ii) the family branch seems to activate similar semantic regions for the two birds within the same family yet different regions for birds belong to different families; (iii) the species branch activates different discriminative regions for distinguishing all three species of birds. This further verifies that our LHT model can capture semantic correlation across different hierarchies.

5. Conclusion

In this work we proposed a probabilistic approach to address the hierarchical classification problem. Our key insight is to directly encode the correlation between two adjacent class hierarchies as conditional probabilities, such that the category labels across different hierarchies can be recursively predicted in a fine-to-coarse order. Beyond that, we designed a confusion loss to further encourage the deep classifiers to learn the hierarchical correlation. Experimental results on three benchmark datasets demonstrate the superiority of our method in hierarchical classification. In future, we will investigate our method in other problems, e.g.,
computer-aided diagnosis, where making a diagnosis for tumors commonly is a typical hierarchical classification task.

**Broad impacts.** In this paper we develop a deep hierarchical classification framework that can help the network mitigate the severity of prediction mistakes. The model has great potential for vision tasks requiring reliable recognition, e.g., autonomous vehicle and disease diagnosis. On the flip side, the model needs a hierarchy of semantic annotations for hierarchical classification, which may be not easy to obtain. As a general deep learning method, this model may suffer from the common problems, such as the safety risk caused by potential adversarial attacks.

**References**

[1] Hichem Bannour and Céline Hudelot. Hierarchical image annotation using semantic hierarchies. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 2431–2434, 2012.

[2] Samy Bengio, Jason Weston, and David Grangier. Label embedding trees for large multi-class tasks. In *NeurIPS*, pages 163–171, 2010.

[3] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangoei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *CVPR*, pages 12506–12515, 2020.

[4] Alsallakh Bilal, Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. Do convolutional neural networks learn class hierarchy? *IEEE TVCG*, 24(1):152–162, 2017.

[5] Dongliang Chang, Kaitue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Your “flaming” is my “bird”: Fine-grained, or not. In *CVPR*, pages 11476–11485, 2021.

[6] Tianshui Chen, Wexi Wu, Yuefang Gao, Le Dong, Xiaonan Luo, and Liang Lin. Fine-grained representation learning and recognition by exploiting hierarchical semantic embedding. In *ACM MM*, pages 2023–2031, 2018.

[7] Jan Chorowski and Navdeep Jaitly. Towards better decoding and language model integration in sequence to sequence models. *arXiv preprint arXiv:1612.02695*, 2016.

[8] Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In *ECCV*, pages 71–84, 2010.

[9] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *ECCV*, pages 48–64, 2014.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[11] Jia Deng, Sanjeev Satheesh, Alexander C Berg, and Fei-Fei Li. Fast and balanced: Efficient label tree learning for large scale object recognition. In *NeurIPS*, volume 24, pages 567–575, 2011.

[12] Ankit Dhall, Anastasia Makarova, Octavian Ganea, Dario Pavllo, Michael Greff, and Andreas Krause. Hierarchical image classification using entailment cone embeddings. In *Conference on Computer Vision and Pattern Recognition Workshops*, pages 836–837, 2020.

[13] Rob Fergus, Hector Bernal, Yair Weiss, and Antonio Torralba. Semantic label sharing for learning with many categories. In *ECCV*, pages 762–775, 2010.

[14] Tianshi Gao and Daphne Koller. Discriminative learning of relaxed hierarchy for large-scale visual recognition. In *ICCV*, pages 2072–2079, 2011.

[15] Gregory Griffin and Pietro Perona. Learning and using taxonomies for fast visual categorization. In *CVPR*, pages 1–8, 2008.

[16] Biyang Guo, Songqiao Han, Xiao Han, Hailiang Huang, and Ting Lu. Label confusion learning to enhance text classification models. In *AAAI*, volume 35, pages 12929–12936, 2021.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[18] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *International Conference on Computer Vision Workshops*, pages 554–561, 2013.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *NeurIPS*, 25:1097–1105, 2012.

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(753):343–444, 2015.

[21] Li-Jia Li, Chong Wang, Yongwen Liu, David M Blei, and Li Fei-Fei. Building and using a semantivisual image hierarchy. In *CVPR*, pages 3336–3343, 2010.

[22] Baoyuan Liu, Fereshteh Sadeghi, Marshall Tappen, Ohad Shamir, and Ce Liu. Probabilistic label trees for efficient large scale image classification. In *CVPR*, pages 843–850, 2013.

[23] Subhansu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[24] Marcin Marszałek and Cordelia Schmid. Constructing category hierarchies for visual recognition. In *ECCV*, pages 479–491, 2008.

[25] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990.

[26] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *NeurIPS*, 32:4694–4703, 2019.

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32:8026–8037, 2019.

[28] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *CVPR*, pages 618–626, 2017.
[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 1

[31] Josef Sivic, Bryan C Russell, Andrew Zisserman, William T Freeman, and Alexei A Efros. Unsupervised discovery of visual object class hierarchies. In *CVPR*, pages 1–8, 2008. 1

[32] Nitish Srivastava and Ruslan Salakhutdinov. Discriminative transfer learning with tree-based priors. In *NeurIPS*, pages 2094–2102, 2013. 3

[33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 1, 3

[34] Anne-Marie Tousch, Stéphane Herbin, and Jean-Yves Audibert. Semantic hierarchies for image annotation: A survey. *PR*, 45(1):333–345, 2012. 2

[35] Nakul Verma, Dhruv Mahajan, Sundararajan Sellamanickam, and Vinod Nair. Learning hierarchical similarity metrics. In *CVPR*, pages 2280–2287, 2012. 3

[36] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5

[37] Hui Wu, Michele Merler, Rosario Uceda-Sosa, and John R Smith. Learning to make better mistakes: Semantics-aware visual food recognition. In *ACM MM*, pages 172–176, 2016. 1, 2

[38] Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. Hdcnn: hierarchical deep convolutional neural networks for large scale visual recognition. In *ICCV*, pages 2740–2748, 2015. 2

[39] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *ICLR*, 2017. 8

[40] Bin Zhao, Li Fei-Fei, and Eric P Xing. Large-scale category structure aware image categorization. In *NeurIPS*, pages 1251–1259, 2011. 2, 3

[41] Alon Zweig and Daphna Weinshall. Exploiting object hierarchy: Combining models from different category levels. In *ICCV*, pages 1–8, 2007. 2
A. Derivation of Equation 1

In this section, we provide a detailed derivation of the equivalence between hierarchical cross-entropy loss (i.e., the sum of cross entropy loss for each hierarchy) and negative log-likelihood of the following equation (Eq. (1) in the main text): \[
p(y^1, \cdots, y^K | x) = \prod_{k=1}^{K} p(y^k | x). \quad (9)
\]

We first denote \( y^k \) the one-hot vector of \( y^k \), \( p^k \) the corresponding prediction of \( y^k \), and \( v[i] \) the \( i \)-th element of vector \( v \). We then have
\[
- \log p(y^1, \cdots, y^K | x) = - \log \prod_{k=1}^{K} p(y^k | x)
\]
\[
= - \sum_{k=1}^{K} \log p(y^k | x)
\]
\[
= - \sum_{k=1}^{K} \sum_{i=1}^{C_k} p(y^k[i]) y^k[i] \log p^k[i]
\]
\[
= \sum_{k=1}^{K} CE(y^k, p^k).
\]

Note that it implies that the prediction at each hierarchy follows a categorical distribution to the right hand side of the third equal sign.

B. Proof of Theorem 1

Proof. The cross-entropy loss in Eq. (5) is bounded below by zero. If there exits \( w^* \) such that the zero loss value in Eq. (5) can be achieved \(^2\), then we have \( p^1(x_i) = y^1_i \) and correspondingly \( p^1 = y^1 \). Meanwhile we have \( \mathbf{T}^2 y^1 = y^2 \), which implies \( \mathbf{T}^2 p^1 = y^2 \). By analogy, we have \( \mathbf{T}^k p^{k-1} = y^k \) for \( k \in \{2, \cdots, K\} \), and the loss value in Eq. (4) becomes zero, then the optimal solution \( w^* \) of problem Eq. (5) is also the optimal solution of Eq. (4).

Now we prove that the optimal solution of Eq. (4) minimizes Eq. (5). This can be easily verified. On one hand, in Eq. (4) we have \( \text{CE}(\mathbf{T}^k p^{k-1}, y^k) \geq 0 \) for \( k \in \{2, \cdots, K\} \) and the zero loss can only be obtained by minimizing the cross-entropy loss of each hierarchy to zero, which prove the results. On the other hand, given the cross-entropy loss of the \( k \)-th hierarchy as \( \text{CE}(\mathbf{T}^k p^{k-1}, y^k) \), if the loss value is minimized to zero by \( \hat{w} \), we have \( \mathbf{T}^k p^{k-1} = y^k \). If \( \hat{w} \) does not result in zero loss of Eq. (5), one can always find a direction to move \( w \) towards \( w^* \) such that the equation \( \mathbf{T}^k p^{k-1} = y^k \) holds and the problem Eq. (5) is minimized as well. \( \square \)

C. Ablation analysis on \( \lambda \)

As discussed in the main text, the proposed confusion loss plays an important role in regularizing our LHT model, and the hyper-parameter \( \lambda \) in Eq. (8) (in the main text, Sec. 3.2) could be used as trade-off between the cross entropy-loss and the confusion loss during training. We herein conduct ablation study on \( \lambda \) to clarify its effect. As shown in Fig. 5, it can observed that: (i) the performance is improved by a significant margin when \( \lambda > 0 \) compared to \( \lambda = 0 \), which indicates that the regularization term is indeed effective; (ii) the results are stable over a wide range of \( \lambda \), e.g., \( \lambda \in [0.5, 10] \), empirically \( \lambda = 2 \) achieves better results on all cases and we thus set \( \lambda = 2 \) in the main text. (iii) for a relatively large \( \lambda \) value, such as \( \lambda = 100 \), where the cross-entropy loss function is over regularized by the confusion loss, the performance degrades by different amounts for different hierarchies. While we can prove that with even larger \( \lambda \) as \( \lambda \rightarrow \infty \), the performance of the fine-grained hierarchy is guaranteed the same as the results provided by a network trained with plain cross-entropy loss at the finest-grained hierarchy, as the following lemma states.

Lemma 1. Given the loss function as
\[
\mathcal{L} = \sum_{i=1}^{N} \sum_{k=1}^{K} \text{CE}(\hat{p}^k_i, y^k_i) + \lambda \mathcal{L}_{\text{Conf}}.
\]
when \( \lambda \rightarrow \infty \), the Eq. (11) is equivalent to
\[
\mathcal{L} = \sum_{i=1}^{N} \text{CE}(\hat{p}^1_i, y^1_i)
\]
\[
\quad (12)
\]

Proof. Minimizing Eq. (11) is equivalent to minimizing
\[
\mathcal{L} = \sum_{i=1}^{N} \sum_{k=1}^{K} \text{CE}(\hat{p}^k_i, y^k_i) + \lambda \mathcal{L}_{kl},
\]
with \( \mathcal{L}_{kl} = \lambda \sum_{i=1}^{N} \sum_{k=2}^{K} \sum_{j=1}^{C_{k-1}} \frac{1}{C_{k-1}} \text{KL}(\mathbf{T}^k(x_i) || u(\frac{1}{C_{k-1}})) \), where \( u(\frac{1}{C_{k-1}}) \) denotes uniform distribution. If the parameter \( \lambda \) is set to be \( \infty \), the we have \( \mathcal{L}_{kl} = 0 \), which means that the label transition matrix \( \mathbf{T}^k, k = \{2, \cdots, K\} \) become constant matrices as
\[
\mathbf{T}^k = \frac{1}{C_k} I
\]
\[
\quad (14)
\]
\[^2\text{Note that for CE}(p(x), y), \text{as } p(x) \text{ is obtained by applying softmax function on the logits } z \text{ as } p(x) = \frac{e^{z_k}}{\sum_{k=1}^{C} e^{z_k}} \text{, by zero loss we mean that } z_y \rightarrow \infty.\]

11
where $I \in \mathbb{R}^{C_k \times C_{k-1}}$ is all-ones matrix. Therefore, we have $T^k \hat{p}^{k-1} = \frac{1}{C_k}[1, 1, \cdots, 1]^T$ for any $\hat{p}^{k-1}$ with $\sum_{i=1}^{C_k-1} \hat{p}^{k-1}_i = 1$. Correspondingly, we have $\text{CE}(\hat{p}^k_i, y^k_i) = \log(C_k)$ which are constants for all $k \in 2, \cdots, K$. Then the training loss function becomes

$$\mathcal{L} = \sum_{i=1}^{N} \text{CE}(\hat{p}^1_i, y^1_i) + \sum_{k=2}^{K} \log(C_k)$$

$$= \sum_{i=1}^{N} \text{CE}(\hat{p}^1_i, y^1_i) + \text{const}$$

which is equivalent to Eq. (12). Thus we finish the proof.

\[\square\]