Hierarchical Novelty Detection for Visual Object Recognition

Kibok Lee[*]   Kimin Lee[†]   Kyle Min[*]   Yuting Zhang[*]   Jinwoo Shin[†]   Honglak Lee[‡*]

[*]University of Michigan, Ann Arbor, MI, USA
[†]Korea Advanced Institute of Science and Technology, Daejeon, Korea
[‡]Google Brain, Mountain View, CA, USA

Abstract

Deep neural networks have achieved impressive success in large-scale visual object recognition tasks with a predefined set of classes. However, recognizing objects of novel classes unseen during training still remains challenging. The problem of detecting such novel classes has been addressed in the literature, but most prior works have focused on providing simple binary or regressive decisions, e.g., the output would be "known," "novel," or corresponding confidence intervals. In this paper, we study more informative novelty detection schemes based on a hierarchical classification framework. For an object of a novel class, we aim for finding its closest super class in the hierarchical taxonomy of known classes. To this end, we propose two different approaches termed top-down and flatten methods, and their combination as well. The essential ingredients of our methods are confidence-calibrated classifiers, data relabeling, and the leave-one-out strategy for modeling novel classes under the hierarchical taxonomy. Furthermore, our method can generate a hierarchical embedding that leads to improved generalized zero-shot learning performance in combination with other commonly-used semantic embeddings.

1. Introduction

Object recognition in large-scale image datasets has achieved impressive performance with deep convolutional neural networks (CNNs) [11, 12, 29, 31]. The standard CNN architectures are learned to recognize a predefined set of classes seen during training. However, in practice, a new type of objects could emerge (e.g., a new kind of consumer product). Hence, it is desirable to extend the CNN architectures for detecting the novelty of an object (i.e., deciding if the object does not match any previously trained object classes). There have been recent efforts toward developing efficient novelty detection methods [2, 13, 17, 20, 25], but most of the existing methods measure only the model uncertainty, i.e., confidence score, which is often too ambiguous for practical use. For example, suppose one trains a classifier on an animal image dataset as in Figure 1. A standard novelty detection method can be applied to a cat-like image to evaluate its novelty, but such a method would not tell
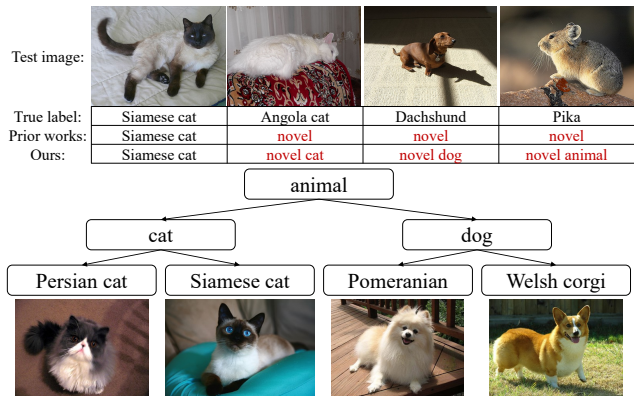

Figure 1: An illustration of our proposed hierarchical novelty detection task. In contrast to prior novelty detection works, we aim to find the most specific class label of a novel data on the taxonomy built with known classes.

whether the novel object is a new species of cat unseen in the training set or a new animal species.

To address this issue, we design a new classification framework for more informative novelty detection by utilizing a hierarchical taxonomy, where the taxonomy can be extracted from the natural language information, e.g., WordNet hierarchy [22]. Our approach is also motivated by a strong empirical correlation between hierarchical semantic relationships and the visual appearance of objects [5]. Under our scheme, a taxonomy is built with the hypernym-hyponym relationships between known classes such that objects from novel classes are expected to be classified into the most relevant label, i.e., the closest class in the taxonomy. For example, as illustrated in Figure 1, our goal is to distinguish "new cat," "new dog," and "new animal," which cannot be achieved in the standard novelty detection tasks. We call this problem *hierarchical novelty detection* task.

In contrast to standard object recognition tasks with a *closed set* of classes, our proposed framework can be useful for extending the domain of classes to an *open set* with taxonomy information (i.e., dealing with any objects unseen in training). In practical application scenarios, our framework can be potentially useful for automatically or interactively organizing a customized taxonomy (e.g., company's prod-

uct catalog, wildlife monitoring, personal photo library) by suggesting closest categories for an image from novel categories (e.g., new consumer products, unregistered animal species, untagged scenes or places).

We propose two different approaches for hierarchical novelty detection: top-down and flatten methods. In the top-down method, each super class has a confidence-calibrated classifier which detects a novel class if the posterior categorical distribution is close to a uniform distribution. Such a classifier was recently studied for a standard novelty detection task [19], and we extend it for detecting novel classes under our hierarchical novelty detection framework. On the other hand, the flatten method computes a softmax probability distribution of all disjoint classes. Then, it predicts the most likely fine-grained label, either a known class or a novel class. Although the flatten method simplifies the full hierarchical structure, it outperforms the top-down method for datasets of a large hierarchical depth.

Furthermore, we combine two methods for utilizing their complementary benefits: top-down methods naturally leverage the hierarchical structure information, but the classification performance might be degraded due to the error aggregation. On the contrary, flatten methods have a single classification rule that avoids the error aggregation, but the classifier's flat structure does not utilize the full information of hierarchical taxonomy. We empirically show that combining the top-down and flatten models further improves hierarchical novelty detection performance.

Our method can also be useful for generalized zero-shot learning (GZSL) [4, 33] tasks. GZSL is a classification task with classes both seen and unseen during training, given that semantic side-information for all test classes is provided. We show that our method can generate a hierarchical embedding that leads to improved GZSL performance in combination with other commonly-used semantic embeddings.

2. Related work

Novelty detection. For robust prediction, it is desirable to detect a test sample if it looks unusual or significantly differs from the representative training data. Novelty detection is a task recognizing such abnormality of data (see [14, 25] for a survey). Recent novelty detection approaches leverage the output of deep neural network classification models. A confidence score about novelty can be measured by taking the maximum predicted probability [13], ensembling such outputs from multiple models [17], or synthesizing a score based on the predicted categorical distribution [2]. There have also been recent efforts toward confidence-calibrated novelty detection, i.e., calibrating how much the model is certain with its novelty detection, by postprocessing [21] or learning with joint objective [19].

Object recognition with taxonomy. Incorporating the hierarchical taxonomy for object classification has been in-

vestigated in the literature, either to improve classification performance [6, 34], or to extend the classification tasks to obtain more informative results [8, 36]. Specifically for the latter purpose, Deng et al. [8] gave some reward to super class labels in a taxonomy and maximized the expected reward. Zhao et al. [36] proposed an open set scene parsing framework, where the hierarchy of labels is used to estimate the similarity between the predicted label and the ground truth. In contemporary work, Simeone et al. [28] proposed a hierarchical classification and novelty detection task for the music genre classification, but their settings are different from ours: in their task, novel classes do not belong to any node in the taxonomy. Thus, their method cannot detect classes which are novel but similar to existing classes. To the best of our knowledge, our work is the first to propose a unified framework for hierarchical novelty detection and visual object recognition.

Generalized zero-shot learning (GZSL). We remark that GZSL [4, 33] can be thought as addressing a similar task as ours. While the standard ZSL tasks test classes unseen during training only, GZSL tasks test both seen and unseen classes such that the novelty is automatically detected if the predicted label is not a seen class. However, the primary focus of ZSL and GZSL tasks is on transfer learning for a new domain, and they assume that semantic information of all test classes is given, e.g., attributes [1, 18, 27] or text description [3, 9, 10, 23, 26, 30] of the objects. Therefore, GZSL cannot recognize a novel class if prior knowledge about the specific novel class is not provided, i.e., it is limited to classifying objects with prior knowledge, regardless of their novelty. Compared to GZSL, the advantages of the proposed hierarchical novelty detection are that 1) it does not require any prior knowledge on novel classes but only utilizes the taxonomy of known classes, 2) a reliable super class label can be more useful and human-friendly than an error-prone prediction over excessively subdivided classes, and 3) high-quality taxonomies are available off-the-shelf and they are better interpretable than latent semantic embeddings. In Section 5, we also show that our models for hierarchical novelty detection can also generate a hierarchical embedding such that combination with other semantic embeddings improves the GZSL performance.

3. Approach

In this section, we define terminologies to describe hierarchical taxonomy and then propose models for hierarchical classification combined with novelty detection.

3.1. Taxonomy

A taxonomy represents a hierarchical relationship among classes, where each node in the taxonomy corresponds to a class or a set of indistinguishable classes.[1]

---

[1] For example, if a class has only one known child class, these two classes are indistinguishable as they are trained with exactly the same data.
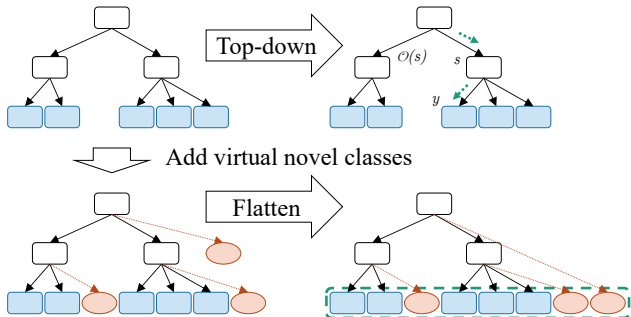
Figure 2: Illustration of two proposed approaches. In the top-down method, classification starts from the root class, and propagates to one of its children until the prediction arrives at a known leaf class (blue) or stops if the prediction is not confident, which means that the prediction is a novel class whose closest super class is the predicted class. In the flatten method, we add a virtual novel class (red) under each super class as a representative of all novel classes, and then flatten the structure for classification.

We define three types of classes as follows: 1) *known leaf classes* are nodes with no child, which are known and seen during training, 2) *super classes* are ancestors of the leaf classes, which are also known, and 3) *novel classes* are unseen during training, so they do not explicitly appear in the taxonomy.[2] We note that all known leaf and novel classes have no child and are disjoint, i.e., they are neither ancestor nor descendant of each other. In the example in Figure 1, four species of cats and dogs are leaf classes, "cat," "dog," and "animal" are super classes, and any other classes unseen during training, e.g., "Angola cat," "Dachshund," and "Pika" are novel classes.

In the proposed hierarchical novelty detection framework, we first build a taxonomy with known leaf classes and their super classes, and at test time, we aim at predicting in the most fine-grained way using the taxonomy. In other words, if an image is predicted as novel, then we try to assign one of the super classes, implying that the input is in a novel class whose closest known class in the taxonomy is that super class.

To represent the hierarchical relationship, let $\mathcal{T}$ be the taxonomy of known classes, and for a class y, $\mathcal{P}(y)$ be the set of parents, $\mathcal{C}(y)$ be the set of children, $\mathcal{A}(y)$ be the set of ancestors including itself, and $\mathcal{N}(y)$ be the set of novel classes whose closest known class is y. And let $\mathcal{L}(\mathcal{T})$ be the set of all descendant leaves under a taxonomy $\mathcal{T}$.

As no prior knowledge of $\mathcal{N}(y)$ is provided during training and testing, all classes in $\mathcal{N}(y)$ are indistinguishable in our hierarchical novelty detection framework. Thus, we treat $\mathcal{N}(y)$ as a single class in our analysis.

3.2. Top-down method

A natural way to perform classification using a hierarchical taxonomy is following *top-down* classification decisions starting from the root class, as shown in the top of Figure 2. Let $(x, y) \sim Pr(x, y|s)$ be a pair of an image and its label sampled from data distribution at a super class s, where $y \in \mathcal{C}(s) \cup \mathcal{N}(s)$. Then, the classification rule is defined as

$$\hat{y} = \begin{cases} \arg\max_{y'} Pr(y'|x, s; \theta_s) & \text{if confident,} \\ \mathcal{N}(s) & \text{otherwise,} \end{cases}$$

where θ_s and $Pr(\cdot|x, s; \theta_s)$ are the model parameters of $\mathcal{C}(s) \cup \mathcal{N}(s)$ and the posterior categorical distribution for an image x, respectively. The top-down classification stops at s if the prediction is a known leaf class or the classifier is not confident with the prediction (i.e., the predicted class is in $\mathcal{N}(s)$). We measure the prediction confidence using the KL divergence with respect to the uniform distribution: intuitively, a confidence-calibrated classifier generates near-uniform posterior probability vector if the classifier is not confident about its prediction. Hence, we interpret that the prediction is confident at a super class s if

$$D_{KL}(U(\cdot|s) \parallel Pr(\cdot|x, s; \theta_s)) \geq \lambda_s,$$

where λ_s is a threshold, D_{KL} denotes the KL divergence, and $U(\cdot|s)$ is the uniform distribution when the classification is made under a super class s. To train such confidence-calibrated classifiers, we leverage classes disjoint from the class s. Let $\mathcal{O}(s)$ be such a set of all known classes except for s and its descendents. Then, the objective function of our top-down classification model at a super class s is

$$\min_{\theta} \; \mathbb{E}_{Pr(x,y|s)} \left[-\log Pr(y|x, s; \theta_s) \right]$$
$$+ \mathbb{E}_{Pr(x,y|\mathcal{O}(s))} \left[D_{KL} \left(U(\cdot|s) \parallel Pr(\cdot|x, s; \theta_s) \right) \right], \quad (1)$$

where $Pr(x, y|\mathcal{O}(s))$ denotes the data distribution of $\mathcal{O}(s)$.

However, under the above top-down scheme, the classification error might aggregate as the hierarchy goes deeper. For example, if one of the classifier has poor performance, then the overall classification performance of all descendent classes should be low. In addition, the taxonomy is not necessarily a tree but a directed-acyclic graph (DAG), i.e., a class could belong to multiple parents, which could lead to incorrect classification.[3] In the next section, we propose flatten approaches, which overcome the error aggregation issue. Nevertheless, the top-down method can be used for extracting good visual features for boosting the performance of the flatten method, as we show in Section 4.

---

[2] We note that "novel" in our task is similar but different from "unseen" commonly referred in ZSL works; while class-specific semantic information for unseen classes must be provided in ZSL, such information for novel classes is not required in our task.

[3] For example, if there are multiple paths to a class in a taxonomy, then the class may belong to (i.e., be a descendant of) multiple children at some super class s, which may lead to low KL divergence from the uniform distribution and the image could be incorrectly classified as $\mathcal{N}(s)$.

3.3. Flatten method

We now propose to enumerate all probability of known leaf and novel classes in a single probability vector, i.e., we *flatten* the hierarchy, as described on the bottom of Figure 2. The key idea is that a probability of super class s can be represented as $Pr(s|x) = \sum_{y' \in \mathcal{C}(s)} Pr(y'|x) + Pr(\mathcal{N}(s)|x)$, and $\sum_{l' \in \mathcal{L}(\mathcal{T})} Pr(l'|x) + \sum_{s' \in \mathcal{T} \setminus \mathcal{L}(\mathcal{T})} Pr(\mathcal{N}(s')|x) = 1$, where l' and s' are summed over all known leaf classes and super classes, respectively. Note that $\mathcal{N}(s)$ is considered as a single novel class under the super class s, as discussed in Section 3.1. Thus, as described in Figure 2, one can virtually add an extra child for each super class to denote all novel classes under it. Let $(x, y) \sim Pr(x, y)$ be a pair of an image and its most fine-grained label sampled from data distribution. Then, the classification rule is

$$\hat{y} = \arg\max_{y'} Pr(y'|x; \theta),$$

where y' is either a known leaf or novel class. Here, a problem is that we have no training data from novel classes. To address this, we propose two approaches to model the score (i.e., posterior probability) of novel classes.

Data relabeling. A naive strategy is to relabel some training samples to its ancestors in hierarchy. Then, the images relabeled to a super class are considered as novel class images under the super class. This can be viewed as a supervised learning with both fine-grained and coarse-grained classes where they are considered to be disjoint, and one can optimize an objective function of a simple cross entropy function over all known leaf classes and novel classes:

$$\min_{\theta} \ \mathbb{E}_{Pr(x,y)} \left[-\log Pr(y|x; \theta_{\mathcal{T}}) \right]. \quad (2)$$

In our experiments, an image is randomly relabeled recursively in a bottom-up manner with a probability of r, where $0 < r < 1$ is termed a relabeling rate. An example of relabeling is illustrated in Figure 3 (b).

Leave-one-out strategy. A more sophisticated way to model novel classes is to temporarily remove a portion of taxonomy during training: specifically, for a training label y, we recursively remove one of its ancestor $a \in \mathcal{A}(y)$ from the taxonomy $\mathcal{T}$ in a hierarchical manner. To represent a deficient taxonomy, we define $\mathcal{T} \setminus a$ as a taxonomy where a and its descendants are removed from the original taxonomy $\mathcal{T}$. At each stage of removal, a training label y becomes a novel class of the parent of a in $\mathcal{T} \setminus a$, i.e., $\mathcal{N}(\mathcal{P}(a))$. Figure 3 (a, c–d) illustrates this idea with an example: in Figure 3 (a), when y is "Persian cat," the set of its ancestor is $\mathcal{A}(y) = \{$ "Persian cat," "cat," "animal" $\}$. In Figure 3 (c), images under $a =$"Persian cat" belong to $\mathcal{N}(\mathcal{P}(a)) =$"novel cat" in $\mathcal{T} \setminus a$. Similarly, in Figure 3 (d), images under $a =$"cat" belong to $\mathcal{N}(\mathcal{P}(a)) =$"novel animal" in $\mathcal{T} \setminus a$. As we leave a class out to learn a novel class,
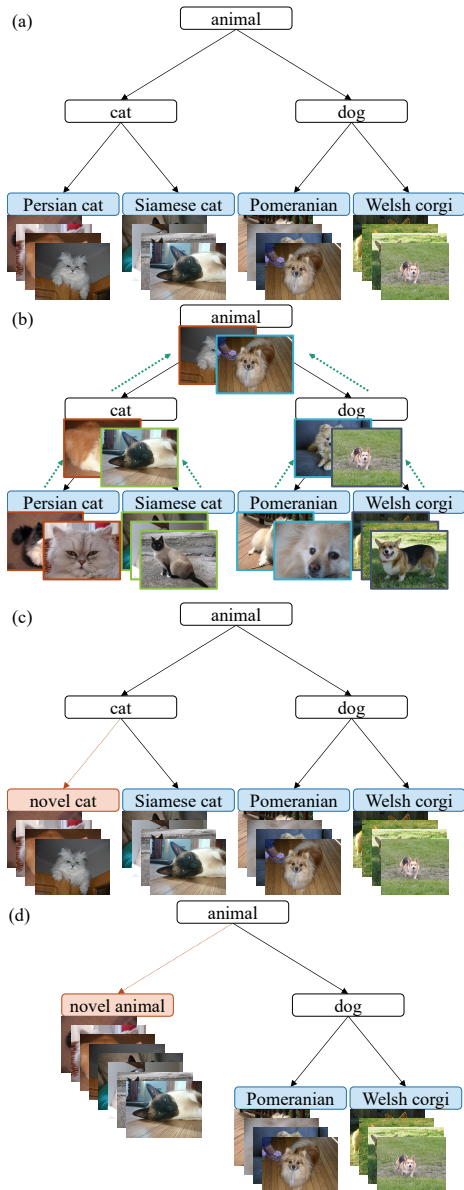

Figure 3: Illustration of strategies to train novel class scores in flatten methods. (a) shows the training images in the taxonomy. (b) shows relabeling strategy. Some training images are relabeled to super classes in a bottom-up manner. (c–d) shows leave-one-out (LOO) strategy. To learn a novel class score under a super class, one of its child is temporarily removed such that its descendant known leaf classes are treated as novel during training.

we call this *leave-one-out* (LOO) method. With some notation abuse for simplicity, the objective function of the LOO model is then

$$\min_{\theta} \ \mathbb{E}_{Pr(x,y)} \left[-\log Pr(y|x; \theta_{\mathcal{L}(\mathcal{T})}) \right. $$
$$\left. + \sum_{a \in \mathcal{A}(y)} -\log Pr(\mathcal{N}(\mathcal{P}(a))|x; \theta_{\mathcal{T} \setminus a}) \right], \quad (3)$$

where the first term is the standard cross entropy loss with the known leaf classes, and the second term is the summation of losses with $\mathcal{N}(\mathcal{P}(a))$ and the leaves under $\mathcal{T} \backslash a$. We provide further implementation details in Supplementary material.

As we mentioned earlier, the flatten methods can be combined with the top-down one in sequence: the top-down method first extracts multiple softmax probability vectors from visual features, and then the concatenation of all probabilities can be used as an input of the LOO model. We name the combined method TD+LOO for conciseness.

4. Evaluation: Hierarchical novelty detection

We present the hierarchical novelty detection performance of our proposed methods combined with CNNs on ImageNet [7], Animals with Attributes 2 (AwA2) [18, 33], and Caltech-UCSD Birds (CUB) [32], where they represent visual object datasets with deep, coarse-grained, and fine-grained taxonomy, respectively. Experimental results on CIFAR-100 [16] can be found in Supplementary material, where the overall trends of results are similar to others.

4.1. Evaluation setups

Compared algorithms. As a baseline, we modify the dual accuracy reward trade-off search (DARTS) algorithm [8] for our purpose. Note that DARTS gives some rewards to labels in hierarchy, where fine-grained prediction gets higher reward. Under this algorithm, for a novel class, its closest super class in the taxonomy would give the maximum reward. At test time, the modified DARTS generates expected rewards for all known leaf and novel classes, so prediction can be done in the same way as the flatten methods.

As our proposed methods, Relabel, LOO, and TD+LOO are compared. For a fair comparison in terms of the model capacity, deep Relabel and LOO models are also experimented, where a deep model is a stack of fully connected layers followed by rectified linear units (ReLU). We do not report the performance of the pure top-down method since 1) one can combine it with LOO methods for better performance as mentioned in Section 3.2, and 2) fair comparisons between the pure top-down method and others are not easy: intuitively, the confidence threshold λ_s in Section 3.2 can be tuned. For example, the novel class score bias in the flatten method would improve the novel class detection accuracy, but large λ_s does not guarantee the best novel class performance in the top-down method because hierarchical classification results would tend to stop at the root class.

Datasets. ImageNet [7] consists of of 22k object classes where the taxonomy of the classes is built with the hypernym-hyponym relationships in WordNet [22]. We take 1k mutually exclusive classes in ILSVRC 2012 as known leaf classes, which are a subset of the ImageNet.[4]

---

[4] Except "teddy bear," all classes in ILSVRC 2012 are in ImageNet.

Based on the hypernym-hyponym relationships in WordNet, we initially obtained 860 super classes of 1k known leaf classes, and then merged indistinguishable super classes. Specifically, if a super class has only one child or shares exactly the same descendant leaf classes, it is merged with classes connected to the class. After merging, the resultant taxonomy is a DAG and has 396 super classes where all super classes have at least two children and have different set of descendant leaf classes. On the other hand, the rest of 21k classes can be used as novel classes for testing. Among them, we discarded super classes, classes under 1k known leaf classes, and classes with less than 50 images for reliable performance measure. After filtering classes, we obtain about 16k novel classes. ILSVRC 2012 has about 1.3M training images and another 50k images in 1k known leaf classes. We put the 50k images aside from training and used for test, and we sampled another 50k images from 1.3M training images for validation. For novel classes, we sampled 50 images from each class. In summary, we have about 1.2M training images, 50k validation images, and 50k test images from known leaf classes, and 800k test images from novel classes.

AwA2 [18, 33] consists of 40 known leaf classes and 10 novel classes with 37k images, and CUB [32] consists of 150 known leaf classes and 50 novel classes with 12k images. Similar to ImageNet, the taxonomy of each dataset is built based on the hypernym-hyponym relationships in WordNet. The resultant taxonomy is a tree and has 21 and 43 super classes for AwA2 and CUB, respectively.

Training. We take ResNet-101 [12] as a visual feature extractor (i.e., the penultimate layer of the CNN before the classification layer) for all compared methods. The CNNs are pretrained with ILSVRC 2012 1k classes, where they do not contain any novel classes of datasets experimented. Then, the final classification layer of the CNNs is replaced with our proposed models. Note that CNNs and our proposed models can be trained in an end-to-end manner, but we take and freeze the pretrained parameters in all layers except for the final layer for the sake of faster training.

For ImageNet, we use mini-batch SGD with 5k center-cropped data per batch. As a regularization, L2 norm weight decay with parameter 10^{-2} is applied. The initial learning rate is 10^{-2} and it decays at most two times when loss improvement is less than 2 % compared to the last epoch.

For AwA2 and CUB, the experiments are done in the same environment with the above except that the models are trained with the full-batch GD and Adam optimizer [15].

Metrics. We first consider the top-1 accuracy by counting the number of predicted labels exactly matching the ground truth. Note that we have two types of classes in test datasets, i.e., known and novel classes. Performances on two types of classes are in trade-off relation, i.e., if one tunes model parameters for favoring novel classes, the accuracy of known
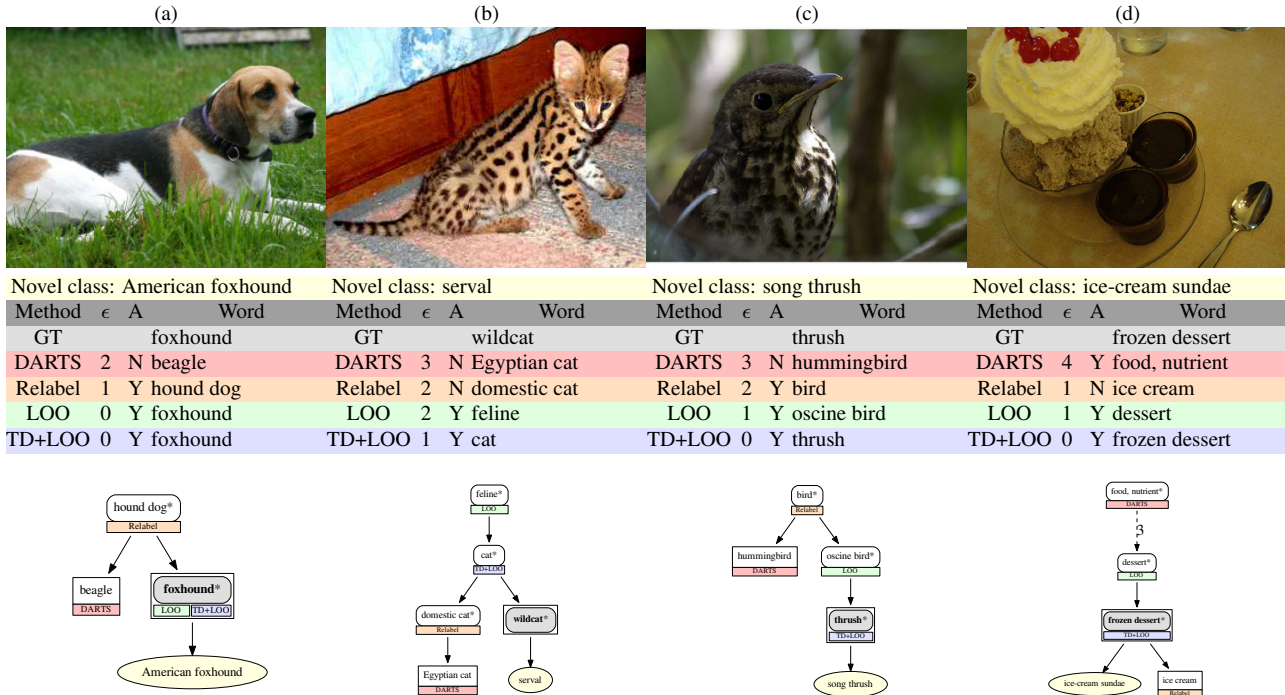
| | (a) | | (b) | | (c) | | (d) |

Novel class: American foxhound			Novel class: serval			Novel class: song thrush			Novel class: ice-cream sundae		
Method	ϵ	A	Word	Method	ϵ	A	Word	Method	ϵ	A	Word
GT			foxhound	GT			wildcat	GT			thrush	GT			frozen dessert
DARTS	2	N	beagle	DARTS	3	N	Egyptian cat	DARTS	3	N	hummingbird	DARTS	4	Y	food, nutrient
Relabel	1	Y	hound dog	Relabel	2	N	domestic cat	Relabel	2	Y	bird	Relabel	1	N	ice cream
LOO	0	Y	foxhound	LOO	2	Y	feline	LOO	1	Y	oscine bird	LOO	1	Y	dessert
TD+LOO	0	Y	foxhound	TD+LOO	1	Y	cat	TD+LOO	0	Y	thrush	TD+LOO	0	Y	frozen dessert

Figure 4: Qualitative results of hierarchical novelty detection on ImageNet. "GT" is the closest known ancestor (super class) of the novel class, which is the expected prediction, "DARTS" is the baseline method proposed in [8] where we adapt their method to our task, and the others are our proposed methods. "ϵ" stands for the distance between the prediction and GT, and "A" indicates whether the prediction is an ancestor of GT. Dashed edges represent multi-hop connection, where the number indicates the number of edges between classes. If the prediction is on a super class (marked with * and rounded), then the test image is classified as a novel class whose closest class in the taxonomy is the super class.

classes would be decreased. Specifically, by adding some positive bias to the novel class scores (e.g., logits in the softmax), one can decrease known class accuracy while increasing novel class accuracy, or vice versa. Hence, for a fair comparison, we measure the novel class accuracy with respect to some fixed known class accuracy, e.g., 50 %. As a more informative evaluation metric, we also measure the area under known-novel class accuracy curve (AUC). Varying the novel class score bias, a curve of known class accuracy versus novel class accuracy can be drawn, which depicts the relationship between the known class accuracy and the novel class accuracy. The AUC is the area under this curve, which is independent of the novel class score bias.

4.2. Experimental results

We first compare the hierarchical novelty detection results of the baseline method and our proposed methods qualitatively with test images on ImageNet in Figure 4. We remark that our proposed methods can provide informative prediction results by utilizing the taxonomy of the dataset. In Figure 4 (a), LOO and TD+LOO find the ground truth label (the most fine-grained label in taxonomy), while DARTS classifies it as "beagle," which is in fact visually similar to "American foxhound." In Figure 4 (b), none of the method finds the ground truth, but the prediction of

Table 1: Hierarchical novelty detection results on ImageNet, AwA2, and CUB. For a fair comparison, 50 % of known class accuracy is guaranteed by adding a bias to all novel class scores (logits). The AUC is obtained by varying the bias. Known-novel class accuracy curve is shown in Figure 5. Values in bold indicate the best performance.

Method	ImageNet		AwA2		CUB	
	Novel	AUC	Novel	AUC	Novel	AUC
DARTS [8]	10.89	8.83	36.75	35.14	40.42	30.07
Relabel	15.29	11.51	45.71	40.28	38.23	28.75
LOO	15.72	12.00	50.00	43.63	40.78	31.92
TD+LOO	**18.78**	**13.98**	**53.57**	**46.77**	**43.29**	**33.16**

TD+LOO is the most informative, as it is the closest label in the hierarchy. In Figure 4 (c–d), only the prediction of TD+LOO is correct, but the rest of the methods also give a reasonable amount of information. More qualitative results can be found in Supplementary material.

Table 1 shows the hierarchical novelty detection performance on ImageNet, AwA2, and CUB. One can note that the proposed methods significantly outperform the baseline method in most cases, except the case of Relabel on CUB, because validation could not find the best relabeling rate for test. Also, we remark that LOO outperforms Relabel.
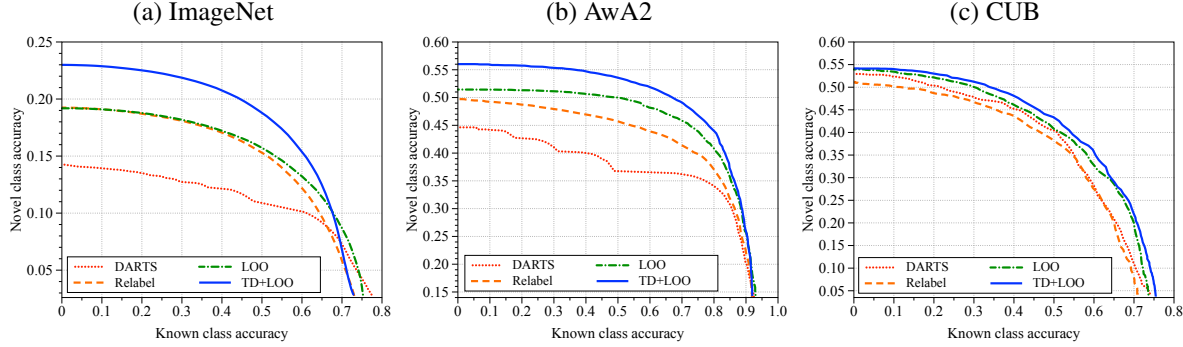
Figure 5: Known-novel class accuracy curves obtained by varying the novel class score bias on ImageNet, AwA2, and CUB. In most regions, our proposed methods outperform the baseline method.

The main difference of two methods is that Relabel gives a penalty to the original label if it is relabeled during training, which turns out to be harmful for the performance. Finally, TD+LOO exhibits the best performance, which implies that the multiple softmax probability vectors extracted from the top-down method is more useful than the vanilla visual features extracted from the state-of-the-art CNNs in the hierarchical novelty detection tasks. Figure 5 shows the known-novel class accuracy curves by varying the bias added to the novel class scores. Our proposed methods have higher novel class accuracy than the baseline in most regions.

5. Evaluation: Generalized zero-shot learning

We present the GZSL performance of the combination of the hierarchical embedding obtained by the top-down method and other semantic embeddings on Animals with Attributes (AwA1 and AwA2)[5] [18, 33] and Caltech-UCSD Birds (CUB) [32].

5.1. Evaluation setups

Hierarchical embeddings for GZSL. GZSL requires an output semantic embedding built with side information, e.g., attributes labeled by human, or word embedding trained with a large text corpus. In addition to those two commonly used semantic embeddings, Akata et al. [1] proposed to use hierarchical relationships of all classes, including classes unseen during training. Specifically, they measured the shortest path distance between classes in the taxonomy built with both known and novel classes, and take the vector of distance values as output embedding. We refer to this embedding as Path.

Motivated by the effectiveness of the features extracted from the top-down method shown in Section 4.2, we set the enumeration of the ideal multiple softmax probability vectors as the semantic embedding: let $\mathcal{C}(s)[i]$ be the i-th

child of a super class s. Then, for a label y and a super class s, the i-th element of an ideal output probability vector $t^{(y,s)} \in [0,1]^{|\mathcal{C}(s)|}$ is

$$t^{(y,s)}[i] = \begin{cases} 1 & \text{if } y \text{ belongs to } \mathcal{C}(s)[i] \\ 0 & \text{if } y \text{ belongs to } \mathcal{C}(s)[j] \text{ where } i \neq j \\ \frac{1}{|\mathcal{C}(s)|} & \text{if } y \text{ is novel or does not belong to } s \end{cases}$$

where $|\mathcal{C}(s)|$ is the number of known child classes under s. The final visual embedding is the concatenation of them with respect to the super classes, i.e., the ground truth semantic vector of a class y is $t^y = [\ldots, t^{(y,s)}, \ldots]$, and we call this embedding TD. See Supplementary material for an example of the ideal output probability vector t^y.

Since classes who share the same closest super class have exactly the same desired output probability vector, we made random guess for fine-grained classification in the experiment only with the hierarchical embedding.

Datasets. AwA1 and AwA2 [18, 33] consists of 40 seen classes and 10 unseen classes with 37k images, and CUB [32] consists of 150 seen classes and 50 unseen classes with 12k images,[6] where the taxonomy can be built in the same way with Section 4.

Training. We note that the performance of combined models is reported in [1], but the numbers are outdated, due to the old CNNs and ZSL models. Thus, instead of making direct comparison with theirs, we construct the environment following the state-of-the-art setting and compared the performance gain obtained by ensembling different hierarchical embedding models to other semantic embeddings. We take ResNet-101 as a pretrained visual feature extractor, and we apply deep embedding model proposed in [35] for training attribute embedding and word embedding models, where it learns to map semantic embeddings to the visual feature embedding with two fully connected layers with ReLU between them. As a combination strategy, we calculate prediction scores of each model and then used their
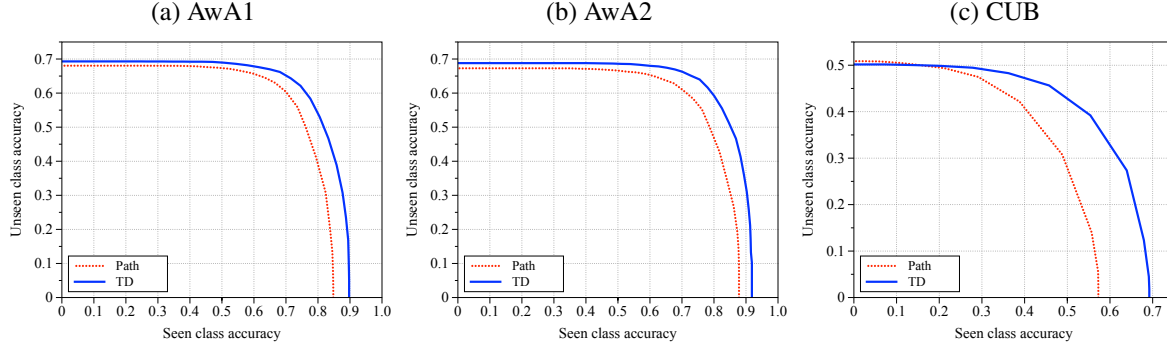

Figure 6: Seen-unseen class accuracy curves of the best combined models obtained by varying the unseen class score bias on AwA1, AwA2, and CUB. "Path" is the hierarchical embedding proposed in [1], and "TD" is the embedding of the multiple softmax probability vector obtained from the proposed top-down method. In most regions, TD outperforms Path.

Table 2: ZSL and GZSL performance of semantic embedding models and their combinations on AwA1, AwA2, and CUB. "Att" stands for continuous attributes labeled by human, "Word" stands for word embedding trained with the GloVe objective [24], and "Hier" stands for the hierarchical embedding, where "Path" is proposed in [1], and "TD" is output of the proposed top-down method. "Unseen" is the accuracy when only unseen classes are tested, and "AUC" is the area under the seen-unseen curve where the unseen class score bias is varied for computation. The curve used to obtain AUC is shown in Figure 6. Values in bold indicate the best performance among the combined models.

Embedding			AwA1		AwA2		CUB	
Att	Word	Hier	Unseen	AUC	Unseen	AUC	Unseen	AUC
✓			65.29	50.02	63.87	51.27	50.05	23.60
	✓		51.87	39.67	54.77	42.21	27.28	11.47
✓	✓		67.80	52.84	65.76	53.18	49.83	24.13
		Path	42.57	30.58	44.34	33.44	24.22	8.38
✓		Path	67.09	51.45	66.58	53.50	50.25	23.70
	✓	Path	52.89	40.66	55.28	42.86	27.72	11.65
✓	✓	Path	68.04	53.21	67.28	54.31	**50.87**	24.20
		TD	33.86	25.56	31.84	24.97	13.09	7.20
✓		TD	66.13	54.66	66.86	57.49	50.17	30.31
	✓	TD	56.14	46.28	59.67	49.39	29.05	16.73
✓	✓	TD	**69.23**	**57.67**	**68.80**	**59.24**	50.17	**30.31**

weighted sum for final classification, where the weights are cross-validated. See [1] for more details about the combination strategy as well as the semantic embeddings.

Metrics. The ZSL performance is measured by testing unseen classes only, and the GZSL performance is measured by the area under seen-unseen curve (AUC) following the idea in [4]. We measure the class-wise accuracy rather than the sample-wise accuracy to avoid the effect of imbalanced test dataset, as suggested in [33].

5.2. Experimental results

Table 2 shows the performance of the attribute, word, and path embedding model, the hierarchical embedding model derived from the proposed top-down method, and their combinations on AwA1, AwA2, and CUB. In Table 2, the standalone performance of top-down method is not better than the path embedding, as it does not distinguish unseen classes sharing the same closest super class. In the same reason, the improvement on ZSL performance with the combined models is fairly small. However, in the GZSL task, the top-down hierarchical embedding shows significantly better performance in the combined models, which means that the top-down embedding is better when distinguishing seen classes and unseen classes together. Compared to the best single semantic embedding model (with attributes), the combination with the top-down embedding leads to absolute improvement of AUC by 7.65%, 7.97%, and 6.71% on AwA1, AwA2 and CUB, respectively, which is significantly better than that of the path embedding.

6. Conclusion

We propose a new hierarchical novelty detection framework, which performs object classification and hierarchical novelty detection by predicting the closest super class in a taxonomy. We propose several methods for the hierarchical novelty detection task and show that our models achieve significantly better performance over prior work. In addition, the hierarchical embedding learned with our model can be combined with other semantic embeddings such as attributes and words to improve generalized zero-shot learning performance. As future work, augmenting textual information about labels for hierarchical novelty detection would be an interesting extension of this work.

References

[1] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, pages 2927–2936, 2015. 2, 7, 8

[2] A. Bendale and T. Boult. Towards open set deep networks. In *CVPR*, 2016. 1, 2

[3] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, 2016. 2

[4] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*, pages 52–68. Springer, 2016. 2, 8

[5] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *ECCV*, pages 71–84. Springer, 2010. 1

[6] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *ECCV*, pages 48–64. Springer, 2014. 2

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE, 2009. 5

[8] 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 *CVPR*, pages 3450–3457. IEEE, 2012. 2, 5, 6

[9] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, pages 2121–2129, 2013. 2

[10] Y. Fu and L. Sigal. Semi-supervised vocabulary-informed learning. In *CVPR*, 2016. 2

[11] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015. 1

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 5

[13] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2016. 1, 2

[14] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126, 2004. 2

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[16] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009. 5

[17] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*, 2017. 1, 2

[18] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *PAMI*, 36(3):453–465, 2014. 2, 5, 7

[19] K. Lee, H. Lee, K. Lee, and J. Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*, 2018. 2

[20] Y. Li and Y. Gal. Dropout inference in bayesian neural networks with alpha-divergences. In *ICML*, 2017. 1

[21] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018. 2

[22] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 1, 5

[23] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. In *ICLR*, 2014. 2

[24] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43, 2014. 8

[25] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014. 1, 2

[26] S. Reed, Z. Akata, B. Schiele, and H. Lee. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016. 2

[27] M. Rohrbach, S. Ebert, and B. Schiele. Transfer learning in a transductive setting. In *NIPS*, pages 46–54, 2013. 2

[28] P. Simeone, R. Santos-Rodríguez, M. McVicar, J. Lijffijt, and T. De Bie. Hierarchical novelty detection. In *International Symposium on Intelligent Data Analysis*, pages 310–321. Springer, 2017. 2

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[30] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, pages 935–943, 2013. 2

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 1

[32] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds 200. 2010. 5, 7

[33] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *arXiv preprint arXiv:1707.00600*, 2017. 2, 5, 7, 8

[34] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu. Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition. In *ICCV*, pages 2740–2748, 2015. 2

[35] L. Zhang, T. Xiang, and S. Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, 2016. 7

[36] H. Zhao, X. Puig, B. Zhou, S. Fidler, and A. Torralba. Open vocabulary scene parsing. In *ICCV*, 2017. 2