# Hierarchical Classification with Confidence using Generalized Logits

Jim Davis, Tong Liang, James Enouen
Ohio State University
Columbus, Ohio 43210
Email: {davis.1719, liang.693, enouen.8}@osu.edu

Roman Ilin
AFRL/RYAP
Wright-Patterson Air Force Base, Ohio 45433
Email: rilin325@gmail.com

***Abstract*—We present a bottom-up approach to hierarchical classification based on posteriors conditioned with logits. Beginning with the output logits for a set of terminal labels from a base classifier, an initial hypothesis is repeatedly generalized (softened) to a weaker label until a particular confidence measure is achieved. As conditioning the probabilistic model with the full set of terminal logits quickly becomes intractable for large label sets, we propose an alternative approach employing "generalized logits" spanning relevant hypotheses within the label hierarchy. Experimental results are compared with related methods on multiple datasets and base classifiers. The proposed approach provides an efficient and effective hierarchical classification framework with monotonic, non-decreasing inference behavior.**

## I. Introduction

Hierarchical reasoning is the process of employing a coarse-to-fine representation of objects/labels/etc. to enable inference at multiple levels of specificity. For example, a phylogenetic tree in biology hierarchically organizes relationships among species based upon physical or genetic features. A particular species can then be analyzed or categorized at different levels of generalization within the tree. In standard classification paradigms, hierarchical semantic relationships of the target (output) labels can be used to guide the classifier into either refining (top-down) or generalizing (bottom-up) labels. A node *within* the tree can be labeled with a semantic generalization of lower more-specific labels (e.g., 'Animal' generalizes 'dog', 'cat', 'bird', 'fish') or instead could be labeled with a subset of terminal labels when semantic generalizations are not available.

Methods employing a top-down strategy typically use a series of classifiers that repeatedly refine the label choice until reaching a label in the terminal set. However, if a classifier error occurs at any level, the process would continue down the wrong branch ending at an incorrect terminal label. This approach could potentially be augmented with a confidence measure/threshold at each classifier branch and terminate early if a target confidence could not be met (and thus return a more general label).

Alternatively, a bottom-up approach having monotonic, non-decreasing probabilistic behavior can act to reduce classifier uncertainty by starting with a terminal label hypothesis and repeatedly generalizing (*softening*) it to a less-specific label that has more confidence, until meeting some required confidence bound. For example, an initial hypothesis (e.g., the label having the largest logit/softmax or terminal posterior) may be for the terminal label 'dog', but with low confidence. However, a sufficiently confident generalized label of 'Canine' or 'Animal' may be achievable. Only when generalizing all the way to the root node would a fully 'Unknown' conclusion be drawn. The ability to ascertain an input is unclassifiable is still important, e.g., to determine if the example was out of context. We employ such a bottom-up generalization strategy in this work.

We approach hierarchical classification using a post-processing method on the output of a given/existing base classifier. Any task-based trained classifier that provides a set of logits for the output target classes can be employed. Given the output of the base classifier, we design our hierarchical module to generalize the initial hypothesis (terminal label with the largest posterior) until reaching a target confidence. Monotonic, non-decreasing probability/confidence of decisions is required to *reduce uncertainty* when moving from a lower, more specific label to a higher, more generalized, label. Note that a top-down approach would necessarily *increase uncertainty* as labels are continually refined into more precise labels. Our monotonic, non-decreasing inference framework is based on an efficient posterior conditioning vector containing logits derived for non-terminal labels.

There are multiple advantages and contributions with our approach:

1) Can be used with any base classifier that produces logit values for output labels.
2) Conditions label posteriors with fewer elements (using generalized logits), making the estimation process more compact and efficient, particularly with datasets having few examples per class.
3) Provides a posterior confidence of each classified, and potentially generalized, test example.
4) Corrects many errors from the original base classifier.
5) Has the desired property of monotonic, non-decreasing probabilistic inference as labels are generalized.

We will demonstrate the proposed generalized logit approach on several datasets with different base classifiers and compare to relevant methods.

## II. Related Work

Multiple techniques have been proposed for reasoning and classifying with hierarchical representations to increase performance, efficiency, or accuracy. In [1], a special decision forest was created that enforces hierarchical structure. A hierarchical sparse embedding based on example images was employed in [2]. The approach of [3] iteratively applied spectral clustering on examples to create a hierarchy. In [4], a series of SVMs between pairs of labels corresponding to siblings in the hierarchy was used. A graphical model of the relationships among terminal nodes was used in [5] to develop an extension of the softmax function which respects hierarchical structure (extended in [6] with probabilistic relationships). An RNN was employed in [7] to refine CNN features from coarse to fine labels. In [8], a dynamic dense network whose connections are dictated by hierarchical structure was attached to a CNN. In [9], a CNN was split into two levels where the network initially decides a general class label and then uses a more-specific classifier to determine the final prediction. The approach of [10] sought to learn a model for predicting "entry-level" categories (natural classes used by people) to classify images. In [11], a neural architecture search method was used to automatically learn a tree structure. Hierarchical reasoning has also been applied to transfer learning [12], [1], [13] and zero-shot learning [14], [15].

The most related approaches to our proposed method are from [16] and [17]. In [16], the output logits of the terminal nodes/labels are computed from separate SVM classifiers (one for each terminal node) and their posteriors are estimated with Platt scaling (logistic regression with a single logit value). Non-terminal posteriors are computed from the sum of descendant terminal posteriors, and therefore conditional independence of the logits is assumed. Their approach is formulated on the maximization of reward (depth of solutions) given a specified overall accuracy on the validation set. In [17], a posterior probability is separately computed for each node/label in the tree conditioned on its (uncalibrated) softmax value, where an equivalent softmax value for a non-terminal node is computed as the sum of softmax values from its terminal descendants. The posterior at each node was modeled non-parametrically using a normalized histogram. Inference was conducted in a bottom-up manner from the argmax-selected label of the base classifier until meeting a confidence threshold. However, monotonic, non-decreasing inference is not guaranteed.

In comparison to [16] and [17], our method uses an idea similar to the summed softmax in [17], but derives a formulation for generalized logits. Unlike [17], our approach and [16] have monotonic, non-decreasing inference behavior. We provide a statistical inference guarantee on test data, similar to [17], but [16] only provides a guarantee on the validation set. We also employ a higher-order logistic regression model (using more logits) than [16]. Overall, our method leverages favorable properties of these algorithms, while introducing new techniques to help alleviate some of their limitations.

## III. Approach

The most straightforward post-processing approach to bottom-up, generalized hierarchical inference with a base classifier that outputs logits (e.g., a neural network) is to first model the label posterior probability at each terminal node conditioned on the *entire* output logit vector $\mathcal{L} = [\ell_1, \ldots, \ell_n]$. The posterior of the initial-guess label (e.g., the argmax-selected class from logits or terminal posteriors), conditioned on $\mathcal{L}$, would then be examined to verify sufficient confidence. If the confidence is lower than desired, the posterior of its parent node would next be examined, continuing upward until the required confidence is achieved.

Any non-terminal posterior can be computed from the summation of all its terminal descendant node posteriors, as each node $\mathcal{N}_i \in \mathcal{N}$ in the tree is conditioned on the same logit vector $\mathcal{L}$ and the terminal classes are mutually exclusive, i.e.,

$$P(\mathcal{N}_i|\mathcal{L}) = \sum_{k \in \downarrow(\mathcal{N}_i)} P(\mathcal{N}_k|\mathcal{L}) \tag{1}$$

where $\downarrow(\mathcal{N}_i)$ returns the set of all terminal descendants of node $\mathcal{N}_i$. Therefore any node $\mathcal{N}_i$ cannot be more probable/confident than its parent

$$P(\mathcal{N}_i|\mathcal{L}) \leq P(\text{Parent}(\mathcal{N}_i)|\mathcal{L}) \tag{2}$$

This monotonic, non-decreasing property of posteriors during generalization also holds for *any* ancestral node of $\mathcal{N}_i$. We will refer to this inference method as the "Reference" approach.

With small dimensionality of $\mathcal{L}$ and a large number of examples, the Reference approach may be computationally feasible, but many datasets have a very large number of target labels (perhaps hundreds or more) which makes the estimation of the probabilities difficult unless an extremely large number of examples can be employed (curse of dimensionality).

Alternatively, approaches exist that make various assumptions to reduce the dimensionality issue. For example, one could condition a terminal label only on the logit value for that class, as in [16]. Independence would therefore be required to properly compute a non-terminal posterior using the summation of descendant terminal posteriors. With logits produced from a single deep neural network (instead of multiple SVM classifiers [16]) there could indeed exist significant dependancies. In [17], each non-terminal posterior was individually modeled and conditioned on the softmax sum of its terminal descendants, however there is no guarantee of monotonic, non-decreasing generalization.

In our framework, we derive a compact conditional logit vector for a generalization process that has monotonic, non-decreasing behavior. Our approach models posteriors conditioned on a smaller set of logits that includes "generalized logits" corresponding to relevant generalized labels within the hierarchy. The use of logits allows the direct use of a logistic regression model for the posterior (as used in [16]). We argue that the use of generalized logits is more comprehensive than a *single* logit and is a sufficient (and smaller) alternative to employing *all* logits.

### A. Generalized Logits

The base classifier directly produces logits for the terminal classes, but an equivalent generalized logit corresponding to a non-terminal superclass node can be recovered based on the relationship between logits and softmax. Consider a softmax vector $\mathcal{S} = [s_1, \ldots, s_n]$ derived from the base classifier logits. The $i$th softmax element $s_i$ is computed as

$$s_i = \frac{e^{\ell_i}}{\sum_{j=1}^{n} e^{\ell_j}} \tag{3}$$

Joining the first $m < n$ elements into a new superclass node/label can be represented by softmax value $\hat{s}_{1:m}$ which contains the softmax mass from elements 1 through $m$

$$\hat{s}_{1:m} = \sum_{i=1}^{m} s_i = \frac{\sum_{i=1}^{m} e^{\ell_i}}{\sum_{i=1}^{m} e^{\ell_i} + \sum_{j=m+1}^{n} e^{\ell_j}} \tag{4}$$

$$= \frac{e^{\hat{\ell}_{1:m}}}{e^{\hat{\ell}_{1:m}} + \sum_{j=m+1}^{n} e^{\ell_j}} \tag{5}$$

This allocation of proportional softmax to $\hat{s}_{1:m}$ thus mimics a new classifier trained to $[\hat{s}_{1:m}, s_{m+1}, \ldots, s_n]$.

The corresponding superclass logit for $\hat{s}_{1:m}$ is therefore

$$\hat{\ell}_{1:m} = \ln(e^{\hat{\ell}_{1:m}}) = \ln(\sum_{i=1}^{m} e^{\ell_i}) \tag{6}$$

To recover the generalized logit for any non-terminal node within a label hierarchy, the base classifier logit values for all its terminal descendant classes are used within Eqn. 6. For example, the generalized logit associated with node $N$ in Fig. 1 is computed from its 4 terminal descendants

$$\hat{\ell}_N = \ln(\sum_{i \in \{E,F,G,H\}} e^{\ell_i}) \tag{7}$$

A generalized logit will always be larger than any of the individual logits that make up its composition. To show this, consider first sorting the logits in descending order/magnitude and separating the largest logit

$$\hat{\ell}_{1:m} = \ln(e^{\ell_1} + \sum_{i=2}^{m} e^{\ell_i}) = \ln(e^{\ell_1} + \Delta) \tag{8}$$

As $e^{\ell_i} > 0$ for any $\ell_i$ and the value of $\Delta > 0$, thus $\hat{\ell}_{1:m} > \ell_1$.

We note the following relationship can be used to eliminate any numerical overflow with large $\ell_i$ values

$$\ln(\sum_i e^{\ell_i}) = a + \ln(\sum_i e^{(\ell_i - a)}) \tag{9}$$

where $a$ is the maximum logit used within the sum (shifting the largest logit to zero). Even if any of the remaining shifted logits underflows, a reasonable answer is still attained as the smallest values do not have any considerable contribution.

We next describe the inference process and show how generalized logits can be employed to reduce the dimensionality of the posterior conditional.

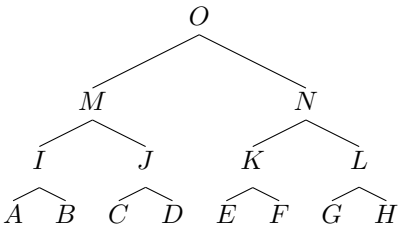


Fig. 1. Example tree with terminals $A$-$H$ and non-terminals $I$-$O$.

### B. Inference

The inference procedure begins with selection of the initial terminal label hypothesis. This could be given as 1) the base classifier argmax-selected class label or 2) the terminal class label having the largest posterior. In this work, we chose the largest posterior approach. If the posterior of this selected label is below the specified confidence threshold, the immediate parent of the initial label is examined, followed by examination of the remaining ancestors on the upward path to the root until the target confidence is achieved.

Consider the tree in Fig. 1 when the base classifier selects terminal $A$ as the initial hypothesis. The possible upward generalizations of $A$ are $I$, $M$, and $O$. Given a set of conditioning logits $\mathcal{L}_A$ (to be defined later) associated with starting at class $A$, we begin by evaluating the posterior $P(A|\mathcal{L}_A)$. If this posterior does not meet a given confidence threshold, then the posterior of its parent $P(I|\mathcal{L}_A)$ is examined. Since node $I$ is composed of the mutually exclusive terminals $A$ and $B$, $P(I|\mathcal{L}_A) = P(A|\mathcal{L}_A) + P(B|\mathcal{L}_A)$. This requires 2 terminal posteriors conditioned on $\mathcal{L}_A$.

If the posterior for $I$ is still below the confidence threshold, then its parent $P(M|\mathcal{L}_A)$ is evaluated. This posterior can be decomposed into $P(M|\mathcal{L}_A) = P(I|\mathcal{L}_A) + P(J|\mathcal{L}_A)$, where the posterior $P(I|\mathcal{L}_A)$ has already been computed and therefore only $P(J|\mathcal{L}_A)$ is needed. Instead of decomposing $P(J|\mathcal{L}_A)$ into the sum of its terminal posteriors $P(C|\mathcal{L}_A) + P(D|\mathcal{L}_A)$, which requires 2 terminal posteriors to be modeled, we can directly model $P(J|\mathcal{L}_A)$ with $P(C \cup D|\mathcal{L}_A)$. Similarly, for root node $O$, we can model $P(O|\mathcal{L}_A)$ as $P(M|\mathcal{L}_A) + P(N|\mathcal{L}_A)$, where $P(N|\mathcal{L}_A) = P(E \cup F \cup G \cup H|\mathcal{L}_A)$. Therefore, when starting from node $A$, posterior information need only be stored for nodes $A$, $B$, $J$, and $N$ as opposed to retaining a fully decomposed set of 8 terminal posteriors ($A$-$H$) conditioned on $\mathcal{L}_A$.

We can now specify the conditioning logits $\mathcal{L}_A$ based on the limited posteriors that are needed (for nodes $A$, $B$, $J$, and $N$). When starting from $A$, the conditioning vector $\mathcal{L}_A$ can be composed of the set

$$\mathcal{L}_A = [\ell_A, \ell_B, \hat{\ell}_J, \hat{\ell}_N] \tag{10}$$

where $\ell_A$ and $\ell_B$ are the sibling terminal logits given directly by the base classifier, and $\hat{\ell}_J$ and $\hat{\ell}_N$ are the generalized logits corresponding to the union posteriors for $J$ and $N$, and are

computed using Eqn. 6 as

$$\hat{\ell}_J = \ln(\sum_{i\in\{C,D\}} e^{\ell_i}) \quad (11)$$

$$\hat{\ell}_N = \ln(\sum_{i\in\{E,F,G,H\}} e^{\ell_i}) \quad (12)$$

The use of generalized logits considerably reduces the number of logits needed from the original full logit set while providing the information relevant to each required posterior. With the binary tree of Fig. 1, having depth $d = 3$, only $d+1 = 4$ logits are needed in the generalized vector for any starting hypothesis instead of the entire $2^d = 8$ terminal logits. The summation of the posteriors conditioned on $\mathcal{L}_A$ retains monotonic, non-decreasing probabilistic inference.

For any starting terminal label hypothesis for a full binary tree of depth $d$, our approach requires $d + 1$ posteriors (and conditioning logits). To accommodate any possible initial hypothesis (there are $2^d$ terminal labels), a total of $(d+1)\cdot 2^d$ posteriors are needed overall. In comparison, the straightforward REFERENCE approach with terminal-only posteriors conditioned on the full logit vector has a total of $2^d$ posteriors. Thus the proposed approach is only a modest linear increase in the number of posteriors for trees of moderate depth.

*1) Extension to non-binary trees:* A binary tree was used to motivate the approach and convey computation bounds, but rarely are label hierarchies truly binary (as will be shown in the experiments). However, due to our bottom-up method of directed inference from an initial hypothesis, any tree can be *evaluated* in a binary manner regardless of its width.

Consider a non-binary version of Fig. 1 where non-terminal $I$ now has 3 children: $A$, $B$, and $B^*$ (new). With the initial hypothesis of $A$, our specified logit vector would now be $\mathcal{L}_A = [\ell_A, \ell_B, \ell_{B^*}, \hat{\ell}_J, \hat{\ell}_N]$ and also require an additional posterior $P(B^*|\mathcal{L}_A)$. As we are only considering generalizations of $A$ to its ancestors $\{I, M, O\}$, a binary set of posteriors *per level* in the tree is all that is required. For the posterior evaluation of $I$ (when $P(A|\mathcal{L}_A)$ is below confidence), we need only compose it from the sum of 2 posteriors, $P(A|\mathcal{L}_A)+P(B\cup B^*|\mathcal{L}_A)$, rather than using all three terminal posteriors. Thus only a target posterior $P(A|\mathcal{L}_A)$ and a non-target siblings posterior $P(B \cup B^*|\mathcal{L}_A)$ are necessary. This two-case posterior approach extends to each level above when evaluating a potential generalized label that has more than 2 children. It is worth noting that any posterior estimation error should be reduced when estimating a single union model (e.g., $P(B \cup B^*|\mathcal{L}_A)$) as compared with the accumulated error from the sum of individually estimated posteriors (e.g., $P(B|\mathcal{L}_A) + P(B^*|\mathcal{L}_A)$).

The vector of generalized logits for starting at $A$ can therefore be re-written as

$$\mathcal{L}_A = [\ell_A, \hat{\ell}_{I\setminus A}, \hat{\ell}_{M\setminus I}, \hat{\ell}_{O\setminus M}] \quad (13)$$

where $\hat{\ell}_{X\setminus Y}$ represents the generalized logit for the set of children for node $X$ with child $Y$ removed. To compute $\hat{\ell}_{I\setminus A}$, for example, when $I$ has children $A$, $B$, and $B^*$, the following can be employed

$$\hat{\ell}_{I\setminus A} = \ln(e^{\hat{\ell}_I} - e^{\ell_A}) = \ln(e^{\ell_B} + e^{\ell_{B^*}}) = \hat{\ell}_{B\cup B^*} \quad (14)$$

When using the actual binary tree, the logit vector does not change from before using this formulation ($\mathcal{L}_A = [\ell_A, \ell_B, \hat{\ell}_J, \hat{\ell}_N]$).

This binary approach allows the method to retain a compact conditioning logit vector and small number of posteriors independent of the tree *width*. The length of the logit vector and number of posteriors are dictated only by the tree *depth* to the initial hypothesis.

### C. Posterior Model

We model the posterior of a label/node using logistic regression, where a sigmoid is estimated using our conditioning logits $\mathcal{L}_i = [\ell_1, \cdots, \hat{\ell}_k]$ with all ground truth positive and negative examples of class $\mathcal{N}_i$ from a *validation* set (as not to overfit the same training data used by the base classifier [17]). The logistic regression model is given by

$$P(\mathcal{N}_i|\mathcal{L}_i) = \frac{1}{1+\exp(-f(\mathcal{L}_i))} \quad (15)$$

with the linear function $f(\mathcal{L}_i) = (a_0 + a_1\ell_1 + \cdots + a_k\hat{\ell}_k)$. In [16], a similar Platt Scaling model using only a single logit was employed $(a_0 + a_1\ell_1)$. To estimate the parameters in Eqn. 15, the Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm [18] was employed, though other methods could be used. We will also use this logistic regression model for the REFERENCE posteriors employed as a baseline technique in the experiments.

To account for any possible numerical precision deviations during inference with a test example, the set of posterior values on the generalization path (corresponding to the initial starting hypothesis and conditioned with the logit vector for the test example) should be L1-normalized (as used in [16]) prior to inference. This guarantees a summation to 1 of the posteriors for the test example at the root node.

## IV. EXPERIMENTS

To demonstrate the advantages of the proposed method, we compared our approach with [16] and [17] (described in Sect. II) across multiple datasets and base classifiers, and additionally measured the similarity of the prediction posteriors against the desired REFERENCE posteriors, where applicable.

### A. Datasets, Label Hierarchies, and Base Classifiers

We examined four established classification datasets for the comparison of techniques. ImageNet-Animal was employed in [17] and contains 398 classes of animals (from the full ImageNet collection [19]), each with different numbers of training examples and 50 testing examples (given as validation). The standard CIFAR-100 dataset [20] contains 100 various objects and scenes, each with 500/100 training/testing examples. In addition, the smaller, yet distinct, CIFAR-10 [20] dataset has 10 object classes, each with 5K/1K training/testing examples.

Lastly, the Fashion-MNIST (F-MNIST) dataset [21] similarly contains 10 classes (shirts, pants, shoes, etc.), each with 5K/1K training/testing examples. As a validation set is required for the methods compared, each class in every testing dataset was randomly, and equally, partitioned into separate validation and test examples.

The semantic trees for the datasets used in the experiments were derived from WordNet using the approach provided in [17] and are shown in Figs. 2-5. The bold numbers in parentheses for Figs. 2 and 3 denote the number of terminals associated with the higher generalized labels, and '$\cdots$' denotes additional generalized nodes present but not displayed due to space. We note that other trees could be used and that they will certainly be application dependent. The min/median/max length of conditioning generalized logit vectors associated with the terminals in each tree are ImageNet-Animal: 3/6/10, CIFAR-100: 2/9/12, CIFAR-10: 3/4/5, and F-MNIST: 2/5/5 (all are much smaller than the original number of terminal labels).

We selected different CNN base classifiers (one for each dataset) spanning different accuracies to provide the terminal class logits. A CNN adapted from the VGGNet structure [22] was trained on F-MNIST to an accuracy of 92.4% (high accuracy). A ResNet-20 (v1) model [23] was trained on CIFAR-10 to 85.6% (medium accuracy). For ImageNet-Animal, we employed a ResNet-152 (v1) model [24] (pre-trained on [19]) giving an accuracy of 84.6% (medium accuracy). Lastly, another ResNet-20 model was purposely trained on CIFAR-100 to only 65.2% (low accuracy). We used this range of base classifier accuracies (strong to weak) to compare the approaches across various possible classifier situations.

### B. Metrics

To evaluate the inference methods, we included metrics used in [16] and [17], along with a few new metrics. The various metrics give credit for predictions that exist on the correct IS-A ancestral path of the ground truth, including the root [16]. No partial credit is given for a prediction off the upward path of the ground truth. Some of the metrics are based on the sets of originally *Correct* (C) and originally *Incorrect* (IC) base classifier predictions (given the ground truth and argmax of the logits). We also compared the posterior values of predictions for each method (for applicable datasets) with the corresponding REFERENCE posteriors (described at the beginning of Sect. III), where logistic regression was employed on the full set of base classifier logits at the terminal level (though a model other than logistic regression could be used). We report the following values:

- **C-Corrupt** is the fraction of originally *Correct* (C) predictions from the base classifier that are relabeled to an incorrect label off the ancestral path of ground truth. Lower proportions are desired.
- **IC-Reform** [17] is the fraction of originally *Incorrect* (IC) base predictions that are generalized to a correct label on the ancestral path of the ground truth. Larger proportions are desired.
- **Accuracy** is the fraction of predictions of the generalized classification results that are correct, where any non-terminal node on the ancestral path of ground truth (including the root) is considered a correct label (as used in [16]).
- **avg-sIG** corresponds to the depth of the generalizations in terms of Information Gain (IG), as similarly used in [16]. The scaled IG (sIG) for a correct prediction at node $\mathcal{N}_i$ is

$$\text{sIG}(\mathcal{N}_i) = \left(\log_2 |\mathcal{T}| - \log_2(|\downarrow(\mathcal{N}_i)|)\right) / \log_2 |\mathcal{T}| \qquad (16)$$

  where $\mathcal{T}$ is the set of all terminals. When a correct prediction is at the terminal level (most precise), the scaled gain is $\text{sIG} = (\log_2 |\mathcal{T}| - \log_2 1) / \log_2 |\mathcal{T}| = 1$. When a prediction is withdrawn to the root ('Unknown'), the scaled gain is $\text{sIG} = 0$. The sIG is 0 by default for any incorrect prediction. We compute the average across all test examples to get **avg-sIG**.
- **C-Withdrawn** [17] is the fraction of originally *Correct* (C) base predictions assigned to the root node ('Unknown').
- **IC-Withdrawn** [17] is the fraction of originally *Incorrect* (IC) base predictions assigned to the root node ('Unknown'). As these predictions were originally incorrect and potentially unclassifiable (e.g., due to occlusion), this value could be large.
- **avg-Ref-diff** is the average of the difference acquired by subtracting the associated REFERENCE posterior value from the examined inference method's posterior value for each of the test predictions. A value close to zero is desired, as it signifies the posteriors of the inference method are similar to the desired REFERENCE posteriors. The standard deviation will also be provided. Note that this score is applicable only for datasets where the actual REFERENCE posteriors can be computed.
- **avg-Ref** is the average of the corresponding REFERENCE posterior values for all test predictions determined from a particular inference method. A value that meets/exceeds the given confidence threshold is desired as it shows that the inference approach actually adheres to the confidence threshold with respect to the REFERENCE posteriors. The standard deviation will also be provided. Again, this score is applicable only for datasets where the REFERENCE posteriors can be computed.

### C. Hierarchical Inference Comparison

We compared our approach with [16] and [17] on each dataset/classifier and report the results in Tables I-IV for the specified metrics at 90% and 95% confidence thresholds (100% confidence typically drives inference to top-level 'Unknown' decisions). We also provide the performance of the base classifiers (with no hierarchical inference).

We begin with evaluation of the approaches on CIFAR-10 and F-MNIST as their posteriors can be scrutinized against the desired REFERENCE posteriors (which can be computed for these two datasets). Given a semantic tree and a confidence threshold for a dataset, we seek to evaluate whether an inference approach picks the "right" labels for the test examples. Choosing the root ('Unknown') for an example could actually be the appropriate decision. Evaluation of the corresponding REFERENCE posterior values for each algorithm's predictions is used to show which inference method provides the most appropriate predictions (with respect to the REFERENCE method).

Comparative results for CIFAR-10 are shown in Table I. For [16], the learned parameters (on the validation set) were $\lambda_{90\%} = 0.934$ ($\epsilon = 90\%$ confidence) and $\lambda_{95\%} = 3.328$ ($\epsilon = 95\%$ confidence), and a single $\tilde{\epsilon} = 0.001$ was used for all datasets. The proposed approach has the overall largest IC-Reform and has the smallest C-Corrupt along with [17] (and

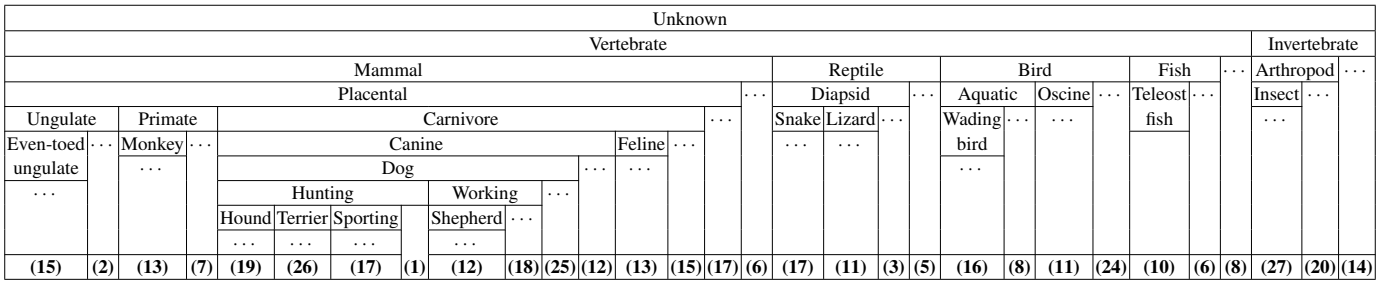


Fig. 2. ImageNet-Animal semantic hierarchy.

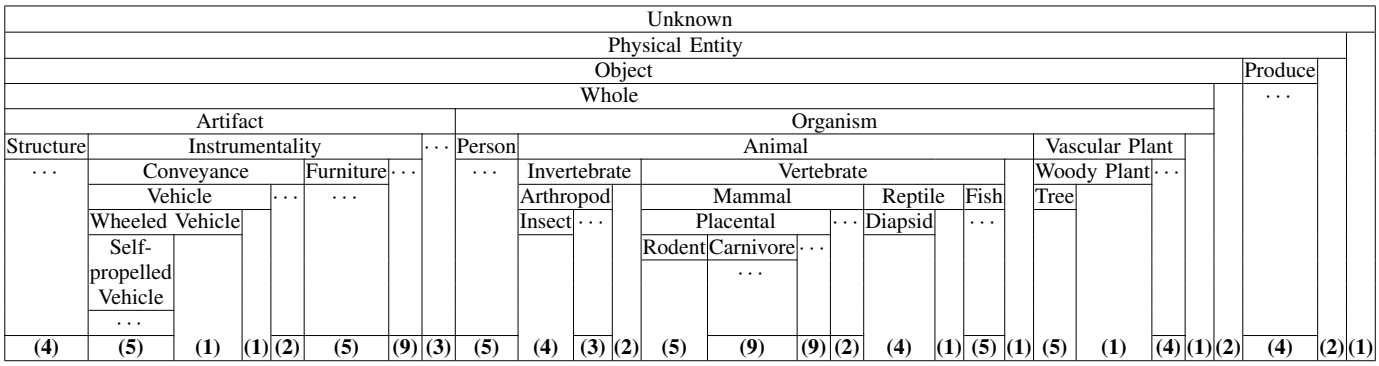


Fig. 3. CIFAR-100 semantic hierarchy.

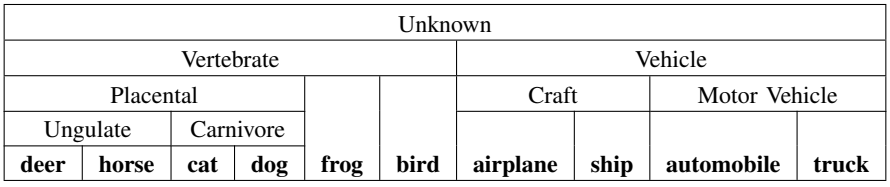


Fig. 4. CIFAR-10 semantic hierarchy.

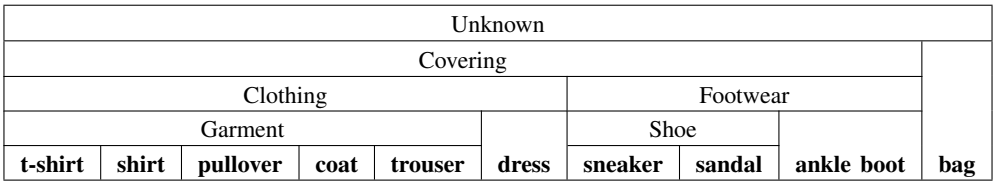


Fig. 5. F-MNIST semantic hierarchy.

[16] at 95%). Note that [17] will always have C-Corrupt = 0 as it starts with the original base classifier hypothesis. The proposed approach has the highest hierarchical accuracy score.

Examining the avg-sIG scores (reflecting the depth of predictions), the largest values (deepest predictions) are given by [16], and are similarly supported by their low C/IC-Withdrawn values. Many more examples were assigned to the root 'Unknown' node for the proposed approach and [17]. However, we must evaluate the posteriors against the REFERENCE posteriors to justify whether the deeper predictions and lower withdrawals are actually warranted. The avg-REF-diff (and standard deviation) for the proposed approach shows that its posterior values are actually much closer and tighter to the desired REFERENCE posterior values. The method of [17] is overly confident and thus falsely assigns labels deeper in the tree. Furthermore, the avg-REF (and standard deviation) shows that the proposed method better meets/exceeds both confidence thresholds ([17] only passes at 90%). Since [16] optimizes only on the validation set during training, it cannot guarantee that any test prediction meets/exceeds the confidence threshold. It actually shows under-confidence in the avg-REF values. From comparison to the REFERENCE method, the results of proposed approach should therefore be accepted over the other methods.

The results for F-MNIST are shown in Table II. The parameters used for [16] were $\lambda_{90\%} = 0.000$ and $\lambda_{95\%} = 1.972$. As before, the proposed approach has a much stronger IC-Reform than the other methods and a similar C-Corrupt. The avg-sIG and C/IC-Withdrawn comparative trends are similar to the previous dataset. The avg-REF-diff scores again show

TABLE I
CIFAR-10 AT 90% AND 95% CONFIDENCE.

| | Base | Proposed | | [16] | | [17] | |
|---|---|---|---|---|---|---|---|
| | | 90% | 95% | 90% | 95% | 90% | 95% |
| C-Corrupt | - | **.00** | **.00** | .02 | **.00** | **.00** | **.00** |
| IC-Reform | - | **.93** | **.98** | .47 | .74 | .79 | .82 |
| Accuracy | .86 | **.99** | **1.00** | .91 | .96 | .97 | .97 |
| avg-sIG | .86 | .59 | .48 | .84 | .77 | .76 | .72 |
| C-Withdrawn | - | .11 | .16 | .00 | .01 | .04 | .04 |
| IC-Withdrawn | - | .26 | .34 | .00 | .08 | .21 | .21 |
| avg-REF-diff | - | **.01** | **.01** | -.04 | -.03 | .06 | .06 |
| st. dev. | - | **.05** | **.03** | .10 | .08 | .10 | .09 |
| avg-REF | .79 | .96 | .98 | .85 | .91 | .91 | .92 |
| st. dev. | .22 | **.06** | **.04** | .18 | .12 | .10 | .09 |

TABLE II
F-MNIST AT 90% AND 95% CONFIDENCE.

| | Base | Proposed | | [16] | | [17] | |
|---|---|---|---|---|---|---|---|
| | | 90% | 95% | 90% | 95% | 90% | 95% |
| C-Corrupt | - | **.00** | **.00** | .02 | .01 | **.00** | **.00** |
| IC-Reform | - | **.93** | **.98** | .22 | .46 | .81 | .85 |
| Accuracy | .92 | **.99** | **1.00** | .92 | .95 | **.99** | .99 |
| avg-sIG | .92 | .79 | .72 | .92 | .90 | .85 | .84 |
| C-Withdrawn | - | .01 | .01 | .00 | .00 | .00 | .00 |
| IC-Withdrawn | - | .07 | .08 | .00 | .00 | .02 | .03 |
| avg-REF-diff | - | **.02** | **.01** | -.03 | -.03 | .04 | .04 |
| st. dev. | - | **.03** | **.03** | .09 | .08 | .07 | .07 |
| avg-REF | .88 | .97 | .98 | .88 | .92 | .95 | .95 |
| st. dev. | .15 | **.05** | **.03** | .15 | .12 | .08 | .07 |

that the posteriors used in the proposed approach are much more similar to the target REFERENCE posteriors. Lastly, both the proposed approach and [17] successfully meet/exceed both confidence thresholds. The REFERENCE comparisons indicate again that the proposed method is more preferred.

We also examined the inference methods on CIFAR-100 and ImageNet-Animal. As the REFERENCE posteriors cannot be computed on these datasets due to their dimensionality, we therefore present the results for the approaches with the expectation of similar inference behavior and support for the proposed approach. We employed a weak base classifier (65.2% accuracy) for CIFAR-100 and a more reasonable base classifier (84.6% accuracy) on ImageNet-Animal to examine the methods with disparate initial base classifier performances.

Results for CIFAR-100 are provided in Table III. For this dataset, the parameters for [16] were $\lambda_{90\%} = 2.102$ and $\lambda_{95\%} = 3.575$. The proposed approach retains similar performance, though the avg-sIG values are much lower. Given that the C/IC-Withdrawn values are fairly similar across the methods, this states that the predictions are more generalized in the semantic tree with the proposed approach. Lastly, the results for ImageNet-Animal are given in Table IV. The parameters for [16] were $\lambda_{90\%} = 0.076$ and $\lambda_{95\%} = 2.404$. These results show a much stronger IC-Reform for the proposed approach, though with higher C/IC-Withdrawn values. As it was shown that the posteriors of the proposed approach were much more similar to the REFERENCE posteriors for CIFAR-10 and F-MNIST, we therefore expect the CIFAR-100 and ImageNet-Animal results to be more appropriate with the proposed approach.

We show a few test predictions across the datasets from the proposed approach at 90% confidence in Fig. 6. Note that CIFAR-10/100 and F-MNIST are comprised of very small, low-resolution images and thus appear blurry in the figure. In the generalized examples of **apple** and **hammerhead**, the base classifier actually predicted the correct labels, though they were deemed unreliable and reasonably generalized by the approach to 'Produce' and 'Fish'. The C-Withdrawn examples of **automobile** and **bag** were also correctly classified by the base classifier, but again they were not confident or standard instances and therefore set to 'Unknown'. It is difficult to see any obvious object in these images. The IC-Reform examples

TABLE III
CIFAR-100 AT 90% AND 95% CONFIDENCE.

| | Base | Proposed | | [16] | | [17] | |
|---|---|---|---|---|---|---|---|
| | | 90% | 95% | 90% | 95% | 90% | 95% |
| C-Corrupt | - | **.00** | **.00** | .02 | .01 | **.00** | **.00** |
| IC-Reform | - | **.98** | **.99** | .74 | .86 | .80 | .81 |
| Accuracy | 0.65 | **.99** | **1.00** | .90 | .95 | .93 | .93 |
| avg-sIG | 0.65 | .16 | .11 | .55 | .48 | .51 | .46 |
| C-Withdrawn | - | .02 | .04 | .02 | .05 | .01 | .01 |
| IC-Withdrawn | - | .04 | .06 | .02 | .08 | .02 | .03 |

TABLE IV
IMAGENET-ANIMAL AT 90% AND 95% CONFIDENCE.

| | Base | Proposed | | [16] | | [17] | |
|---|---|---|---|---|---|---|---|
| | | 90% | 95% | 90% | 95% | 90% | 95% |
| C-Corrupt | - | **.00** | **.00** | .03 | .01 | **.00** | **.00** |
| IC-Reform | - | **.96** | **.98** | .48 | .72 | .67 | .70 |
| Accuracy | .85 | **.99** | **1.00** | .89 | .95 | .95 | .95 |
| avg-sIG | .85 | .30 | .20 | .78 | .69 | .71 | .68 |
| C-Withdrawn | - | .07 | .13 | .00 | .01 | .01 | .01 |
| IC-Withdrawn | - | .09 | .14 | .00 | .06 | .03 | .05 |

of **ankle boot** and **bear** show reasonable generalizations to 'Footwear' and 'Placental' given the confusing appearances. It is understandable how the base classifier mislabeled **bear** as *elephant*. In each case, the lowest common ancestor in the semantic tree was selected for the incorrect base classifier prediction and ground truth. Lastly, the IC-Withdrawn images of **lobster** and **deer** have no obvious visual distinction in the imagery and thus were incorrectly labeled by the base classifier and set to 'Unknown' by our approach.

Overall, the proposed approach using generalized logits provided the best C-Corrupt, IC-Reform, and Accuracy scores, and was shown with CIFAR-10 and F-MNIST to better approximate the desired REFERENCE posteriors. The largest (deepest) avg-sIG across the datasets was achieved by [16], as they directly optimize for deeper labels, though they reformed less and adhered least to the REFERENCE posteriors. The histogram-based method of [17] was more similar to the proposed approach in some situations, however that method does not have monotonic, non-decreasing inference behavior as labels are generalized.

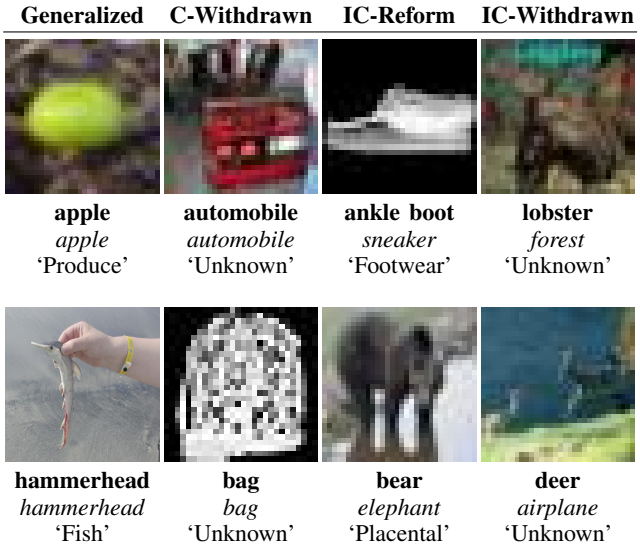


Fig. 6. Example classification results at 90% confidence from the four datasets (**Ground truth** / *Base Classifier* / 'Proposed method').

## V. Conclusion

In this work, we presented a bottom-up probabilistic inference framework that generalizes an initial label hypothesis within a hierarchical representation until a highly confident prediction can be found. We condition the estimation of label posteriors using a compact vector of generalized logits. The proposed post-processing architecture offers an efficient means to hierarchical inference while retaining monotonic, non-decreasing inference behavior. Experimental results compared related methods on multiple datasets and base classifiers trained to various accuracies to demonstrate the applicability of the approach. A further comparison to a reference posterior (when applicable) was used to determine the reliability of the predictions. The method is applicable to multiple classification scenarios in which confident, generalized output labels are preferred over flat, unconfident predictions.

## Acknowledgment

This work was supported by the U.S. Air Force Research Laboratory contracts #GRT00044839 and #GRT00054740.

## References

[1] M. Ristin, J. Gall, M. Guillaumin, and L. van Gool, "From categories to subcategories: Large-scale image classification with partial class label refinement," in *CVPR*, 2015.

[2] B. Kim, J. Park, A. C. Gilbert, and S. Savarese, "Hierarchical classification of images by sparse approximation," in *BMVC*, 2013.

[3] Y. Qu, L. Lin, F. Shen, C. Lu, Y. Wu, Y. Xie, and D. Tao, "Joint hierarchical category structure learning and large-scale image classification," in *IEEE Transactions on Image Processing*, 2017.

[4] S. Albaradei and Y. Wang, "Object classification using a semantic hierarchy," in *ISVC*, 2014.

[5] 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*, 2014.

[6] N. Ding, J. Deng, K. P. Murphy, and H. Neven, "Probabilistic label relation graphs with Ising models," in *ICCV*, 2015.

[7] Y. Guo, Y. Liu, E. M. Bakker, Y. Guo, and M. S. Lew, "CNN-RNN: A large-scale hierarchical image classification framework," in *Multimedia Tools and Applications*, 2018.

[8] X. Liang, H. Zhou, and E. Xing, "Dynamic-structured semantic propagation network," in *CVPR*, 2018.

[9] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu, "HD-CNN: Hierarchical deep convolutional neural network for large scale visual recognition," in *ICCV*, 2015.

[10] V. Ordonez, J. Deng, Y. Choi, A. C. Berg, and T. L. Berg, "From large scale image categorization to entry-level categories," in *ICCV*, 2013.

[11] C. Murdock, Z. Li, H. Zhou, and T. Duerig, "Blockout: Dynamic model selection for hierarchical deep networks," in *CVPR*, 2016.

[12] R. Salakhutdinov, A. Torralba, and J. Tenenbaum, "Learning to share visual appearance for multiclass object detection," in *CVPR*, 2011.

[13] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *CVPR*, 2017.

[14] H. Zhao, X. Puig, B. Zhou, S. Fidler, and A. Torralba, "Open vocabulary scene parsing," in *ICCV*, 2017.

[15] X. Sun, Y. Zi, T. Ren, J. Tang, and G. Wu, "Hierarchical visual relationship detection," in *International Conf. on Multimedia*, 2019.

[16] 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*, 2012.

[17] J. Davis, T. Liang, J. Enouen, and R. Ilin, "Hierarchical semantic labeling with adaptive confidence," in *ISVC*, 2019.

[18] C. Zhou, R. H. Byrd, P. Lu, and J. Nocedal, "L-BFGS-B - Fortran subroutines for large-scale bound constrained optimization," *ACM Transactions on Mathematical Software*, vol. 23, pp. 550–560, 1994.

[19] "ImageNet Large Scale Visual Recognition Challenge 2012." [Online]. Available: http://image-net.org/challenges/LSVRC/2012/index

[20] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html

[21] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," https://arxiv.org/pdf/1708.07747.pdf, 2017.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conf. on Learning Representations*, 2015.

[23] "Trains a ResNet on the CIFAR-10 Dataset - Keras Documentation." [Online]. Available: https://keras.io/examples/cifar10_resnet/

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2015.