AMENDING MISTAKES POST-HOC IN DEEP NETWORKS BY LEVERAGING CLASS HIERARCHIES

Anonymous authors
Paper under double-blind review

ABSTRACT

There has been increasing interest in building deep hierarchy-aware classifiers, aiming to quantify and reduce the severity of mistakes and not just count the number of errors. The idea is to exploit the label hierarchy (e.g., WordNet ontology) and consider graph distances as a proxy for mistake severity. Surprisingly, on examining mistake-severity distributions of the top-1 prediction, we find that current state-of-the-art hierarchy-aware deep classifiers do not show practical improvement in making better mistakes than the standard cross-entropy baseline.The reason for reduction in average mistake-severity can be attributed to a large increase in low-severity or easily avoidable mistakes, which may explain the noticeable drop in accuracy. To this end, we use the classical Conditional Risk Minimization (CRM) framework for hierarchy aware classification. Given a cost matrix and a reliable estimate of likelihoods (obtained from a trained network), CRM simply amends mistakes at inference time; it needs no extra parameters; it requires adding just one line of code to the standard cross-entropy baseline. It significantly outperforms the state-of-the-art and consistently obtains large reductions in the average hierarchical distance of top-k predictions across datasets, with very little loss in accuracy. Since CRM does not require retraining or fine-tuning of any hyperparameter, it can be used with any off-the-shelf cross-entropy trained model.

1 INTRODUCTION

The conventional performance measure of accuracy for image classification treats all classes other than ground truth as equally wrong. However, some mistakes may have a much higher impact than others in real-world applications. An intuitive example being an autonomous vehicle mistaking a car for a bus is a *better mistake* than mistaking a car for a person. Consequently, it is essential to integrate the notion of mistake severity into classifiers and one convenient way to do so it to use a taxonomic hierarchy tree of class labels, where severity is defined by a distance on the graph (e.g., height of the Lowest Common Ancestor) between the ground truth and the predicted label (Deng et al., 2010; Zhao et al., 2011). This is similar to the problem of ranking classes and samples in a classification and retrieval setting respectively. Consider the case of an autonomous vehicle ranking classes for a thin, white, narrow band (a pole, in reality). A top-3 prediction of "pole, lamppost, tree" would be a *better* prediction than "pole, person, building". Notice that the top-k class predictions would have at least $k-1$ incorrect predictions here, and the aim is to reduce the severity of these mistakes, measured by the average hierarchical distance of the top k predictions each with the ground truth. Silla & Freitas (2011) survey classical methods leveraging class hierarchy when designing classifiers across various application domains and illustrate clear advantages over the flat classification, especially when the labels have a well-defined hierarchy.

There has been growing interest in the problem of deep hierarchy-aware image classification (Barz & Denzler, 2019; Bertinetto et al., 2020). These approaches seek to leverage the class hierarchy inherent in the large scale datasets (e.g., the ImageNet dataset is derived from the WordNet semantic ontology). Hierarchy is incorporated either using label embedding methods, hierarchical loss functions, or hierarchical architectures. These models improve the ranking of predicted classes – ensuring that the top alternative classes are closer in the hierarchy.

However, we find concerns with the state-of-the-art deep hierarchy-aware classification methods. Surprisingly, no hierarchy-aware deep classifier seems to make better mistakes than the cross-

entropy baseline on the top prediction, which is typically used to make decisions. Rather, they make additional low-severity mistakes, bringing down the average severity without reducing the number of high-severity mistakes. Hence, existing state-of-the-art hierarchy-aware classification models are *worse* than the cross-entropy baseline, as they show a sharp drop in classification accuracy without improving the severity of top-1 mistakes. Upon investigation, we find that the mismatch between the model's confidence and correctness (i.e. poor *calibration*), and the use of a proxy that is not representative of graph distance, like `word2vec` distance, could be potential reasons for the worse performance.

We explore a different direction for hierarchy-aware classification – amending mistakes by making post-hoc corrections. We discover that classical Conditional Risk Minimization (CRM) outperforms state-of-the-art deep hierarchy-aware classifiers by large margins at ranking classes with little loss in classification accuracy. CRM is simple, requiring the addition of just one line of code to the standard cross-entropy model, and requires no further training or optimization of hyperparameters. This makes it fast and easy to adapt to existing classifiers. CRM also guarantees optimal post-hoc re-ranking classes among other decision rules given the trained model. CRM is able to make better mistakes on top prediction and modestly shift the distribution of mistakes from higher severity to lower severity.

We want to emphasize that we do not claim any algorithmic novelty as CRM has been well explored in literature (Duda & Hart, 1973, Ch. 2). Ten years ago, Deng et al. (2010) had proposed a very similar solution using Support Vector Machines (SVMs) as classifiers over handcrafted features. However, this did not result in practically useful performance because of the lack of modern machine learning tools. Our primary goal is to bring this old, simple, and extremely effective approach back into the attention before we delve deeper into sophisticated ones with modern classifiers. Overall, our investigation into hierarchy-aware classification makes the following contributions:

- We highlight significant issues that have been overlooked by the past works that aim to leverage hierarchies to make better mistakes by first demonstrating how they do not really make better mistakes.
- We revisit an old post-hoc correction technique which significantly outperforms prior art when the ranking of the predictions made by the model are considered.
- We also investigate the reliability of prior art in terms of calibration and show that these methods are severely miscalibrated, limiting their practical usefulness.

2 Related Works

2.1 Cost-Sensitive Classification

Cost-sensitive classification assigns varying costs to different types of misclassification errors. The work by Abe et al. (2004) groups cost-sensitive classifiers into three main categories. The first category specifically extends one particular classification model to be cost-sensitive, such as support vector machine (Tu & Lin, 2010) or decision tree (Lomax & Vadera, 2013). The second category makes the training procedure cost-sensitive, which is typically achieved by assigning the training examples of different classes with different weights (rescaling) (Zhou & Liu, 2010) or by changing the proportions of each class while training using sampling (rebalancing) (Elkan, 2001). The third category makes the prediction procedure cost-sensitive (Domingos, 1999; Zadrozny & Elkan, 2001a). Such direct cost-sensitive decision-making is the most generic: it considers the underlying classifier as a black box and extends to any number of classes and arbitrary cost matrices. Our work comes under the third category of post-hoc amendment. We study cost-sensitive classification in a large scale setting (e.g., ImageNet) and explore the use of a taxonomic hierarchy to obtain the misclassification costs.

2.2 Hierarchy Aware Classification

There is rich literature around exploiting hierarchies to improve the task of image classification. Embedding-based methods define each class as a soft embedding vector, instead of the typical one-hot vector. DeViSE (Frome et al., 2013) learn a transformation over image features to maximize the

cosine similarity with their respective `word2vec` label embeddings. The transformation is learned using ranking loss and places the image embeddings in a semantically meaningful space. Akata et al. (2015); Xian et al. (2016) explore variations of text embeddings, and ranking loss frameworks. Barz & Denzler (2019) project classes on a hypersphere, such that the correlation of class embeddings equals the semantic similarity of the classes. The semantic similarity is derived from the height of the lowest common ancestor (LCA) in a given hierarchy tree.

Another line of effort directly alters the loss functions or the algorithms/architectures. Zhao et al. (2011) propose a weighted (hierarchy aware) multi-class logistic regression formulation. Verma et al. (2012) optimize a context-sensitive loss to learn a separate distance metric for each node in the class taxonomy tree. Wu et al. (2016) combine losses at different hierarchies of the tree by learning separate, fully connected layers for each level post a shared feature space. Bilal et al. (2017) add branches at different depths of AlexNet architecture to fuse losses at different levels of the hierarchy. Brust & Denzler (2019) use conditional probability chains to derive a novel label encoding and a corresponding loss function.

Most deep learning-based methods overlook the severity of mistakes, and the evaluation revolves around counting the top-k errors. Bertinetto et al. (2020) has revived the interest in this direction by jointly analyzing the top-k accuracies with the severity of errors. They propose two modifications to cross-entropy to better capture the hierarchy: one based on soft-labels and the other, which factors the cross-entropy loss into the individual terms for each of the edges in the hierarchy tree and assigns different weights to them.

We propose a generalized alternative. Our method uses models trained with vanilla cross-entropy loss and alters the decision rule to pick the class minimizing conditional risk. On similar lines, Deng et al. (2010) study the effect of minimizing conditional risk on mean hierarchical cost. They leverage the ImageNet hierarchy for cost and compute posteriors by fitting a sigmoid function to the SVM's output or taking the percent of neighbours from a class for Nearest Neighbour classification. Our work investigates the relevance of CRM in the deep learning era and highlights the importance of looking beyond mean hierarchical costs and jointly analyzing the role of accuracy and calibration.

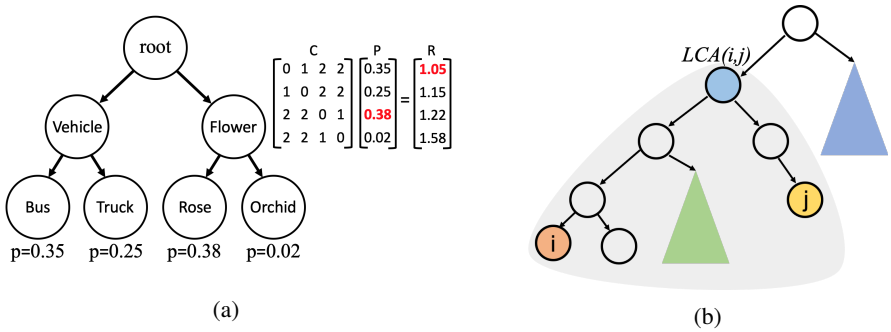
2.3 Calibration of Deep Neural Networks

Networks are said to be well-calibrated if their predicted probability estimates are representative of the true correctness likelihood. Calibrated confidence estimates are important for model interpretability and its use in downstream applications. Platt scaling (Platt et al., 1999), Histogram binning (Zadrozny & Elkan, 2001b) and Isotonic regression (Zadrozny & Elkan, 2002) are three common calibration methods. Although originally proposed for the SVM classifier, their variations are used in improving the calibration of neural networks (Guo et al., 2017). Calibrated probability estimates are particularly important when cost-sensitive decisions are to be made (Zadrozny & Elkan, 2001b) and are often measured using Expected Calibration Error (ECE) and Maximum Calibration Error (MCE) (Naeini et al., 2015; Niculescu-Mizil & Caruana, 2005).

We desire models with high accuracy that have low calibration error and make less severe mistakes. However, there is often a compromise. Studies in cost-sensitive classification (Jan et al., 2012) reveal a trade-off between costs and error rates. For instance, Bertinetto et al. (2020) trade the top-k error for a lower hierarchical cost. Similarly, reliability literature aims to obtain better calibrated deep networks while retaining top-k accuracy (Seo et al., 2019). We further observe that methods like soft labels or hierarchical cross-entropy successfully minimize the average top-k hierarchical cost, but result in poorly calibrated networks. In contrast, the proposed framework of posthoc amendment retains top-k accuracies and good calibration, while significantly reducing the hierarchical cost.

3 Approach

The K-class classification problem comes with a training set $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, where label $\mathbf{y}_i \in \mathcal{Y} = \{1, 2, ..., K\}$. The classifier is a deep neural network parametrized by θ given by $f_\theta : \mathcal{X} \rightarrow p(\mathcal{Y})$ which maps the input samples to a probability distribution over the label space $\mathcal{Y}$. The $p(\mathbf{y}|\mathbf{x})$ is typically derived using a softmax function on the logits obtained for an input $\mathbf{x}$. Given $p(\mathbf{y}|\mathbf{x})$, the network minimizes cross-entropy with the ground truth class over samples from the training set,



(a) (b)

Figure 1: (a) Consider a four class hierarchy tree and corresponding leaf predictions obtained using a cross entropy baseline. The risk computation is shown beside the tree. $\operatorname{argmax} p(\mathbf{y}|\mathbf{x})$ predicts the class "rose", while the $\operatorname{argmin} R(\mathbf{y}=k|\mathbf{x})$ predicts the class "bus". (b) Two nodes i and j and the subtree (shaded gray) originating at their lowest common ancestor $LCA(i,j)$.

and uses SGD to optimize θ, forming the standard hierarchy-agnostic cross entropy baseline. The decision rule is naturally given by $\operatorname{argmax}_k p(\mathbf{y}=k|\mathbf{x})$.

The classical CRM framework (Duda & Hart, 1973) can be adapted to image classification by taking the trained model with a given θ and incorporating the hierarchy information at deployment time. A symmetric class-relationship matrix $\mathbf{C}$ is created using the given hierarchy tree (which can either be drawn from the WordNet ontology or an application specific taxonomy), where $\mathbf{C}_{i,j}$ is the height of the lowest common ancestor ($LCA(\mathbf{y}_i, \mathbf{y}_j)$) between the classes i and j. The height is the number of edges between the given node and the furthest leaf. $\mathbf{C}_{i,j}$ is zero when $i=j$ and is bounded by the maximum height of the hierarchy tree.

Given an input $\mathbf{x}$, we obtain $p(\mathbf{y}|\mathbf{x})$ by passing it through the deep network $f_\theta(\mathbf{x})$. The only modification is in the decision rule, which now selects the class which minimizes the conditional risk $R(\mathbf{y}=k|\mathbf{x})$, given by:

$$\operatorname*{argmin}_k R(\mathbf{y}=k|\mathbf{x}) = \operatorname*{argmin}_k \sum_{j=1}^{K} \mathbf{C}_{k,j} \cdot p(\mathbf{y}=j|\mathbf{x})$$

For the ease of the reader, we illustrate a four-class example in Figure 1a, showing predictions from the cross entropy baseline (leaf nodes); cost matrix for the given tree, and the risk computation. Given the probability of each class $p(\mathbf{y}|\mathbf{x})$, $\operatorname{argmin} R(\mathbf{y}|\mathbf{x})$ is the Bayes optimal prediction. It is guaranteed to achieve the lowest possible overall cost, i.e. lowest expected cost over all possible examples weighted by their probabilities (Duda & Hart, 1973, Ch. 2).

Depending on the cost-matrix and $p(\mathbf{y}|\mathbf{x})$, the top-1 prediction of the CRM applied on crossentropy might differ from the top-1 prediction of the crossentropy baseline. However, with deep neural networks, we observe that the top-1 probability of $p(\mathbf{y}|\mathbf{x})$ is greater than 0.5 in roughly 70% of the samples. Below we prove that in such situations where $\max p(\mathbf{y}|\mathbf{x})$ is higher than the sum of other probabilities, irrespective of the structure of the tree, the post-hoc correction (CRM) does not change the top-1 prediction. However, since the second highest probability is guaranteed to be less than 0.5 by definition, our correction can effectively re-rank the classes and drastically reduce hierarchical distance@k for $k > 1$ - as we later demonstrate through exhaustive experiments.

Theorem 1. *If* $\max(p(\mathbf{y}|\mathbf{x})) > 0.5$, *then* $\operatorname{argmin}_i \sum_{j=1}^{K} \mathbf{C}_{i,j} \cdot p(\mathbf{y}=j|\mathbf{x})$ *and* $\operatorname{argmax} p(\mathbf{y}|\mathbf{x})$ *are identical irrespective of the tree structure and lead to the same top-1 prediction.*

Proof. Consider the tree illustrated in Figure 1b; two leaf nodes (class labels) i,j and the subtree (T_{ij}) rooted at their Lowest Common Ancestor. Assuming the height of the $LCA(i,j)$ = h and $\operatorname{argmax} p(\mathbf{y}|\mathbf{x}) = i$, the risk $R(\mathbf{y}=j|\mathbf{x}) = R(j)$ is given as:

$$R(j) = h \cdot p(i) + \sum_{k \in T_{ij} \setminus \{i\}} C_{j,k} \cdot p(k) + \sum_{\forall k \notin T_{ij}} C_{j,k} \cdot p(k)$$

Ignoring the cost of other nodes inside T_{ij}, we get $R(j) \geq h \cdot p(i) + \sum_{\forall k \notin T_{ij}} C_{j,k} \cdot p(k)$. Similarly, for the risk of class i:

$$R(i) \leq h \cdot (1 - p(i)) + \sum_{\forall k \notin T_{ij}} C_{i,k} \cdot p(k)$$

Outside the subtree rooted at $T_{i,j}$, $C_{i,k} = C_{j,k} \forall k$ and therefore without loss of generality we can say that $R(i) < R(j)$, if $p(i) > 0.5$. □

4 EXPERIMENTS

We evaluate our method on two large-scale hierarchy-aware benchmarks: (i) tieredImageNet-H for a broad range of classes and (ii) iNaturalist-H for fine-grained classification, both of which are complex enough to cover a large number of visual concepts. We closely follow the experimental pipeline from Bertinetto et al. (2020) including the train/validation/test splits, hyperparameters for training models, and evaluation metrics.

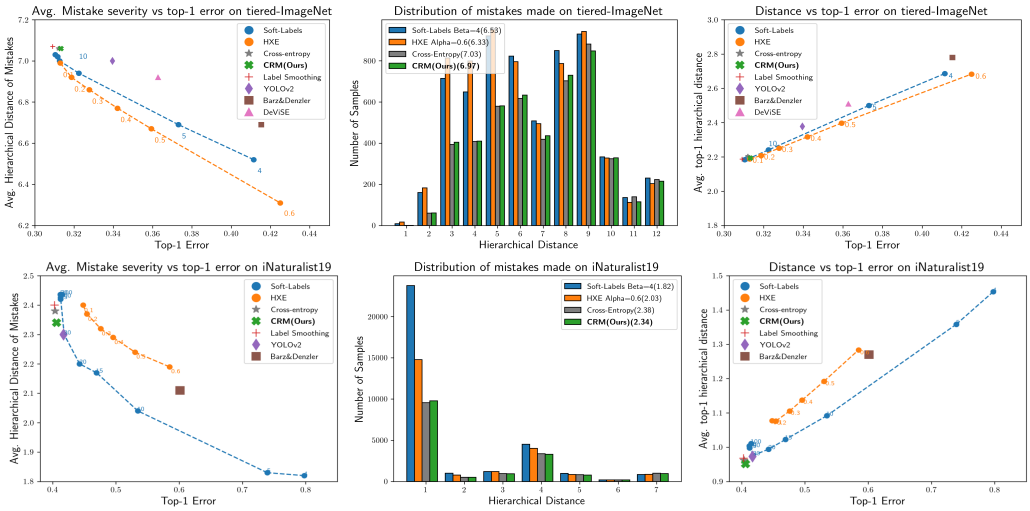
Experimental Details: All models are trained using a ResNet-18 architecture (pre-trained on ImageNet) using an Adam optimizer for 200K updates using a mini-batch of 256 samples, a learning rate of 10^{-5}, and standard data augmentation of flips and randomly resized crops. We train all the hierarchy-aware models – Hierarchical Cross-Entropy (HXE) (Bertinetto et al., 2020), Soft-labels (Bertinetto et al., 2020), YOLO-v2 (Redmon & Farhadi, 2017), DeViSE (Frome et al., 2013), and Barz & Denzler (2019) – along with a cross-entropy baseline. We pick the epoch corresponding to the lowest loss on the validation set along with two epochs preceding and succeeding it and report the average of the results obtained from these five checkpoints on the test set. Unlike Bertinetto et al. (2020) we do not preprocess the dataset to downsample the images to 224x224 as it noticeably reduces the accuracy. Instead we use `RandomResizedCrop()` augmentation to crop the images to a 224x224 resolution. This accounts for a small, but significant improvement in performance across models, thus leading to stronger baselines.

Metrics: We follow Bertinetto et al. (2020) and use two metrics: (i) top-1 error, (ii) average hierarchical distance@k, which is the mean LCA height between the ground truth and each of the k most likely classes. These metrics capture different views of the problem: top1 error treats all classifier mistakes the same, whereas average hierarchical distance@1 captures a notion of mistake severity, i.e., *better* or *worse* mistakes. Average hierarchical distance@k captures the notion of ranking/ordering the predicted classes closer to the ground truth class. Note that LCA is a log-scaled distance: an increment of 1.0 signifies an error of an entire level of the tree. In the simple case of a full binary tree, an increase by one level implies that the number of possible leaf nodes doubles. Another metric used in the prior art is the average mistake severity, which measures the average hierarchical distance@1 on only those samples where the model's prediction was incorrect. We measure calibration in terms of Expected Calibration Error (ECE) and Maximum Calibration Error (MCE).

4.1 MAKING BETTER MISTAKES

Hierarchy-aware classification methods typically seek to make better mistakes (less costly in terms of hierarchical distance). It is essential that the evaluation metric correctly measures this goal, i.e. a higher value of the evaluation metric should reflect that the model indeed makes better mistakes. We discover a serious flaw with the metric of computing average mistake severity over incorrectly classified samples employed in the prior art. We illustrate this in Figure 2.

In Figure 2(a), following prior art, we evaluate different approaches only on the set of incorrectly classified samples (hence *different* test sets for different models as the mistakes will be different). It seems to indicate that recently proposed methods are able to achieve a good tradeoff between accuracy and mistake severity. We select the best performing models in terms of the mistake severity metric – Softlabels with $\beta = 4$ and HXE with $\alpha = 0.6$ – and analyze the frequency of mistakes at different severity (illustrated in Figure 2(b)). Surprisingly, we observe that HXE and Soft-labels largely do not make better mistakes; they mostly make additional low-severity mistakes. This behaviour is not captured in Figure 2(a) as the metric here involves division by the number of mistakes



(a) (b) (c)

Figure 2: (a) Trade-off between hierarchical distance@1 and the top-1 error. The mean here is taken only over the misclassified samples. (b) Histogram of the severity of top-1 mistakes (height of the LCA). The number in the bracket is the mean mistake severity. (c) Trade-off between hierarchical distance@1 and the top-1 error. The mean is taken over all the test samples (the hierarchical distance@1 is zero for correctly classified samples). The top and bottom row correspond to tiered-ImageNet and iNaturalist19 datasets respectively.

made by a model. For example, say the high severity mistakes made by two models is exactly the same ($d_h > 0$) over the same number of mistakes ($m > 0$). Now, if the second model makes additional $n > 0$ mistakes with overall distance severity of $d_l > 0$, then it is straightforward to observe that $\frac{d_h}{m} \geq \frac{d_h+d_l}{m+n}$ if $\frac{d_h}{m} \geq \frac{d_l}{n}$. Implying, the metric would prefer a model which is making additional low severity mistakes as long as the impact of the severity due to these additional mistakes is less than the impact by the high-severity ones.

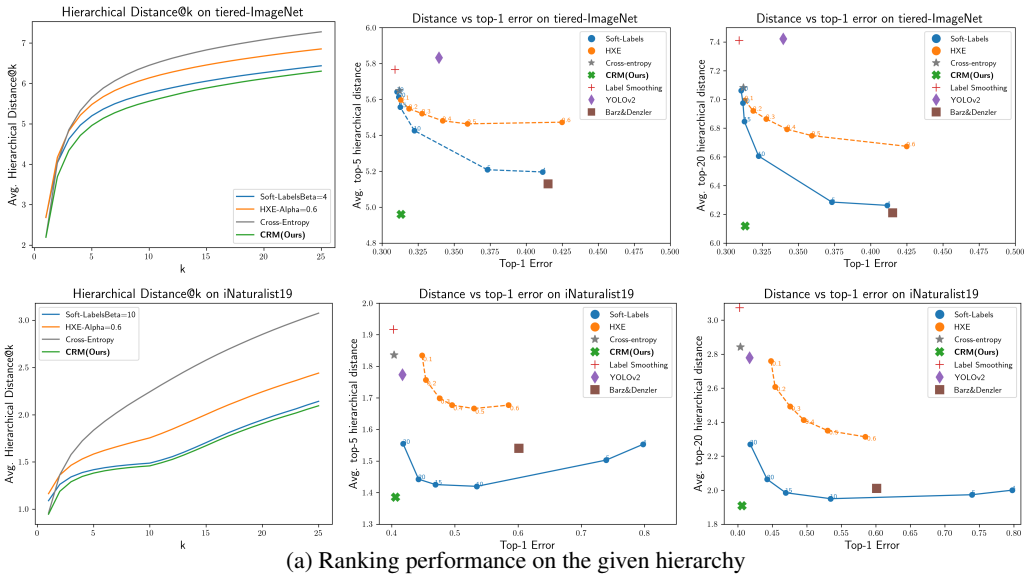
We fix the above problem by simply setting the evaluation set to be the *same* across all the models. This simple fix now provides a more reliable evaluation. As shown in Figure 2(c), the best-performing ones in Figure 2(a) now show the highest hierarchical distance@1 as we account for the additional number of low-severity mistakes made. We refer the reader to Appendix (A.1) for histogram plots across different values of α and β, clearly showing that the number of low-severity mistakes increases as the model moves farther from the cross-entropy baseline. In this modified and more reliable evaluation set-up, we observe CRM (ours) reduces mistake severity compared to cross-entropy, shifting the graver mistakes in cross-entropy towards the left of the histogram—resulting in a marginal shift of mistakes from buckets (9,10,12) to (3,6,7,8) in Imagenet and similarly from buckets (7,5,4) to (1) in iNaturalist19.

Overall, we observe that existing methods reduce the average mistake-severity metric by largely making additional low-severity or easily avoidable mistakes. This also explains why such models provide lower test accuracy. Resolving this issue, we see that no prior art significantly outperforms the cross-entropy baseline, and make better mistakes.

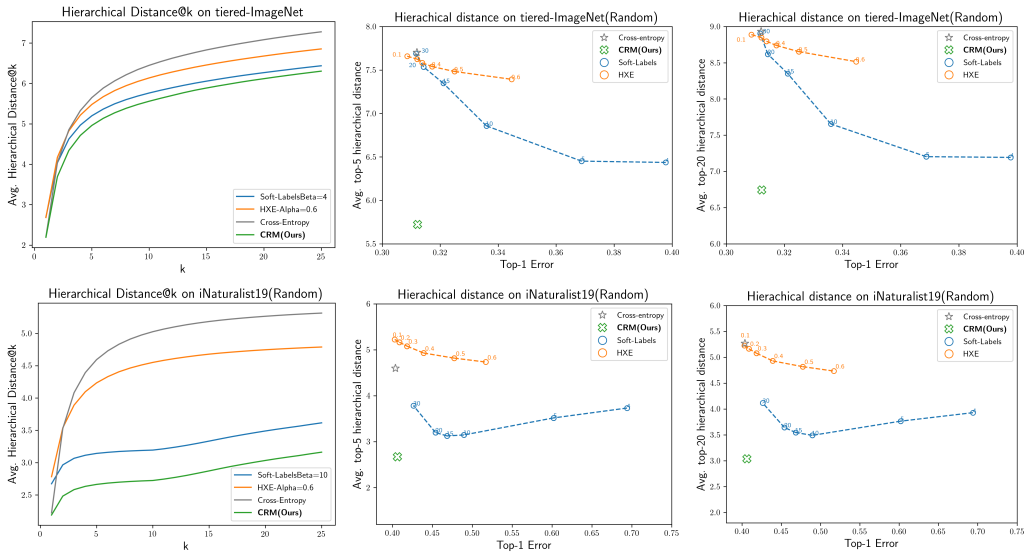
4.2 Improving Label Ranking

We now compare the ordering of classes provided by each of these classifiers. Ranking predictions give us significant insight into how reliably the predictions align with the hierarchy, as discussed in Section 1. We measure the quality of ranking using the average hierarchical distance@k, for various values of k and present them in Figure 3 (a) (left). We find that CRM significantly outperforms all the competing methods, giving the best hierarchically aligned models on the hierarchical distance@k for all k.

A better ranking of classes often comes with a significant trade-off with top-1 accuracy. We plot the hierarchical distance@k with top-1 accuracy for k=5 and k=20 in Figure 3 (a) (middle and right) to



(a) Ranking performance on the given hierarchy



(b) Ranking performance on a randomly class-shuffled hierarchy.

Figure 3: Left: Average hierarchical distance@k with varying k. Middle: Average hierarchical distance@5 vs top-1 error. Right: Average hierarchical distance@20 vs top-1 error.

better understand this trade-off. Interestingly, we observe that CRM improves ranking with almost no loss in top-1 performance and outperforms the other methods by a substantial margin.

An interesting extension is to analyze how dependent these approaches are to a given hierarchy, and how modifying the hierarchy might impact their behaviour. To test this, we randomly shuffle the classes at the leaf nodes of a given tree structure and compare ranking performance in Figure 3 (b). We observe that even though CRM does not explicitly use the hierarchy while training, it provides drastic reduction in the hierarchical distance@k compared to all the previous methods. High accuracy of CRM in this case is because of the fact that it is post-hoc and for highly confident models such as deep networks, its top-1 accuracy remains largely unchanged (refer Theorem1). However, on the other hand, models depending on the tree-structure during training (directly or indirectly) will try to fit to the structure, which can be harmful in situations where the tree structure is not very reliable. For example, if the tree structure implies that 'cat' is closer to 'person' than a 'dog', then the models incorporating such information while learning the feature space might not be able to learn a robust classifier and might potentially end-up making more mistakes, as also validated in our experiments.

Loss Function	tieredImageNet-H				iNaturalist-H			
	ECE		MCE		ECE		MCE	
	pre T	post T	pre T	post T	pre T	post T	pre T	post T
Soft-labels (β=15)	7.05%	4.79%	18.55%	13.03%	34.64%	16.63%	54.86%	26.87&
Soft-labels (β=10)	29.55%	6.36%	39.95%	22.07%	29.55%	19.87%	39.95%	33.29%
Soft-labels(β=5)	58.99%	10.92%	83.53%	26.86%	24.53%	17.20	88.32%	55.68%
Soft-labels (β=4)	57.16%	11.12%	86.92%	27.44%	19.29%	11.46%	19.29%	56.06%
HXE (α=0.2)	1.53%	1.53%	5.84%	5.84%	4.37%	1.50%	7.73%	3.61%
HXE (α=0.4)	2.44%	2.44%	5.48%	5.48%	1.13%	1.13%	2.62%	2.62%
HXE(α=0.5)	3.95%	2.61%	7.84%	5.40%	2.46%	2.46%	6.77%	6.77%
HXE (α=0.6)	6.25%	3.28%	10.75%	6.58%	5.24%	5.24%	11.20%	11.20%
Label-Smoothing	9.61%	2.33%	15.43%	6.13%	4.93%	1.11%	7.35%	3.35%
Cross-Entropy	1.61%	1.61%	4.27%	4.27%	4.32%	1.42%	8.18%	3.26%

Table 1: ECE and MCE for the various models on the tiered-ImageNet and iNaturalist19 datasets before and after temperature scaling. The optimal temperature was found on the validation set and results reported on the test set.

4.3 Impact of Label Hierarchy on the Reliability

Via various examples and insights, we have already established that models that make better mistakes are extremely important for real-world applications. However, in order for such models to be useful in safety-critical scenarios, they must be reliable. In other words, these models should be calibrated so that they are not wrong with high confidence—an extremely undesirable property of recent deep models. To this end, we analyse the reliability of the output probabilities of Softlabels, HXE, label smoothing, and CRM (which is same as cross-entropy) using well known and widely accepted metrics such as ECE (Expected Calibration Error) and MCE (Maximum Calibration Error) in Table 1. Softlabels and HXE, for example, show clear trends of increasing degradation in calibration on better class ranking (as measured by distance@k), i.e., *the more they attempt to adhere to the hierarchy, the less reliable their probability estimates become*. We additionally experiment with improving calibration in all the above models using temperature scaling. We observe that it reduces miscalibration as measured by the ECE and MCE scores, but most models remain far worse than the cross-entropy baseline. The cross-entropy baseline is not affected significantly by temperature scaling, indicating that it is calibrated to a large extent. Changes in ECE/MCE were unnoticeable when using the probability estimates corresponding to CRM predictions (taking $p(\mathbf{y}|\mathbf{x})$ corresponding to $\operatorname{argmin} R(\mathbf{y}|\mathbf{x})$) instead of maximum cross-entropy prediction.

These experiments clearly suggest that while the focus should turn into developing models that make better mistakes, we should also make sure that such models are reliable by understanding how incorporating the label hierarchy during training might impact the likelihood estimates.

5 Conclusion

Our paper identifies two major problems in existing approaches for tackling hierarchy-aware classification: (i) making better mistakes and (ii) better ranking predictions. We study post-hoc amendments for hierarchy aware image classification, different from the three dominant paradigms: hierarchy-aware losses, hierarchy-aware architectures, and label embedding methods. We make a case for exhausting classical alternatives before moving to more sophisticated options.

Regarding making better mistakes, we demonstrate that no recent hierarchy-aware classifier helps make better mistakes, contrary to the motivation. We illustrate and correct the issue with the mistake severity metric and show that CRM achieves modest improvements. We provide insights on why it is hard to improve further in this direction under similar constraints.

In terms of better ranking predictions, our proposed post-hoc correction consistently outperforms state-of-the-art methods in deep hierarchy-aware image classification by large margins in terms of decrease in hierarchical distance@k, furthermore with little to no loss in top-1 accuracy. We find the direction of post-hoc corrections promising because it can simultaneously deliver calibration, accuracy, and better class ranking efficiently with surprisingly little tradeoffs in either. We hope that CRM will act as a strong baseline in future efforts.

Overall, the literature on hierarchy-aware image classification has shown the effectiveness of the WordNet hierarchy in improving performance. However, previous works assume that all edges in the tree are equally important. A future avenue for exploration is learning weights of the edges in the tree to compute a more effective measure of misclassification cost.

REFERENCES

Naoki Abe, Bianca Zadrozny, and John Langford. An iterative method for multi-class cost-sensitive learning. In *KDD*, 2004.

Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015.

Björn Barz and Joachim Denzler. Hierarchy-based image embeddings for semantic image retrieval. In *WACV*, 2019.

Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *CVPR*, 2020.

Alsallakh Bilal, Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. Do convolutional neural networks learn class hierarchy? *IEEE transactions on visualization and computer graphics*, 2017.

Clemens-Alexander Brust and Joachim Denzler. Integrating domain knowledge: using hierarchies to improve deep classifiers. In *ACPR*, 2019.

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

Pedro Domingos. Metacost: A general method for making classifiers cost-sensitive. In *KDD*, 1999.

Richard O Duda and Peter E Hart. Pattern recognition and scene analysis, 1973.

Charles Elkan. The foundations of cost-sensitive learning. In *IJCAI*, 2001.

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NeurIPS*, 2013.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

Te-Kang Jan, Da-Wei Wang, Chi-Hung Lin, and Hsuan-Tien Lin. A simple methodology for soft cost-sensitive classification. In *KDD*, 2012.

Susan Lomax and Sunil Vadera. A survey of cost-sensitive decision tree induction algorithms. *ACM Computing Surveys (CSUR)*, 2013.

Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, 2015.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *ICML*, 2005.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 1999.

Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017.

Seonguk Seo, Paul Hongsuck Seo, and Bohyung Han. Learning for single-shot confidence calibration in deep neural networks through stochastic inferences. In *CVPR*, 2019.

Carlos N Silla and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 2011.

Han-Hsing Tu and Hsuan-Tien Lin. One-sided support vector regression for multiclass cost-sensitive classification. In *ICML*, 2010.

Nakul Verma, Dhruv Mahajan, Sundararajan Sellamanickam, and Vinod Nair. Learning hierarchical similarity metrics. In *CVPR*, 2012.

Hui Wu, Michele Merler, Rosario Uceda-Sosa, and John R Smith. Learning to make better mistakes: Semantics-aware visual food recognition. In *ACM MM*, 2016.

Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016.

Bianca Zadrozny and Charles Elkan. Learning and making decisions when costs and probabilities are both unknown. In *KDD*, 2001a.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, 2001b.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *KDD*, 2002.

Bin Zhao, Fei Li, and Eric P Xing. Large-scale category structure aware image categorization. In *NeurIPS*, 2011.

Zhi-Hua Zhou and Xu-Ying Liu. On multi-class cost-sensitive learning. *Computational Intelligence*, 2010.

A APPENDIX

A.1 VARIATION OF MISTAKE SEVERITY WITH HIERARCHY ALIGNMENT

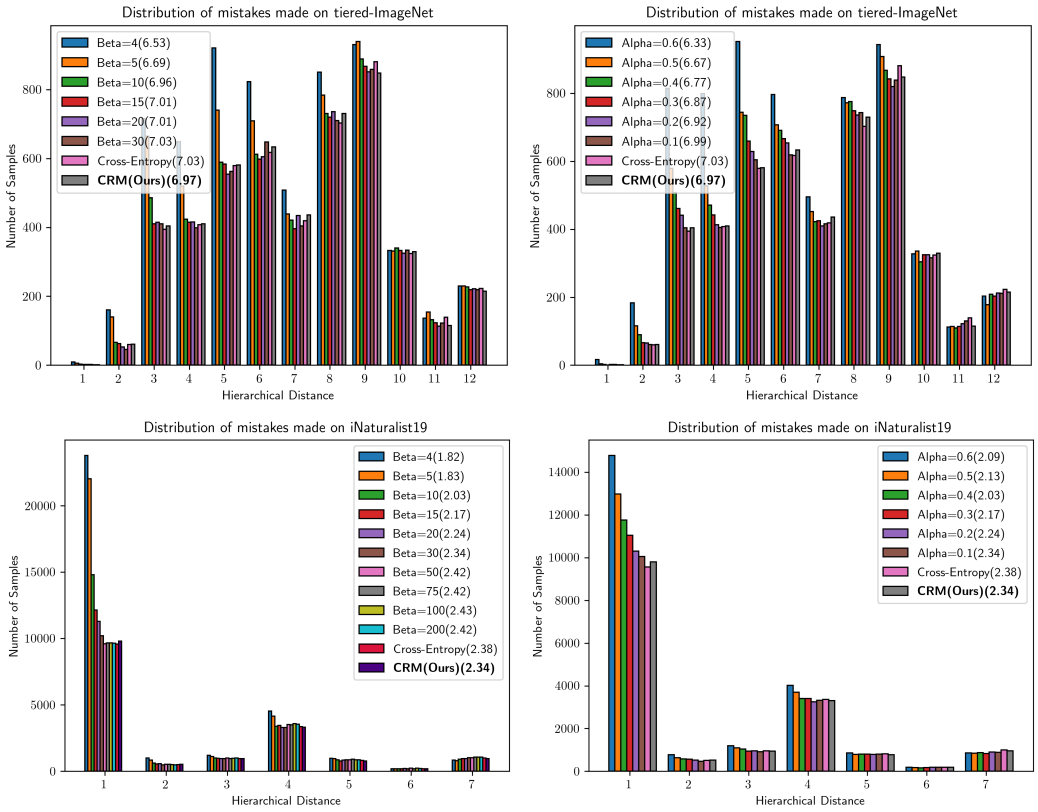
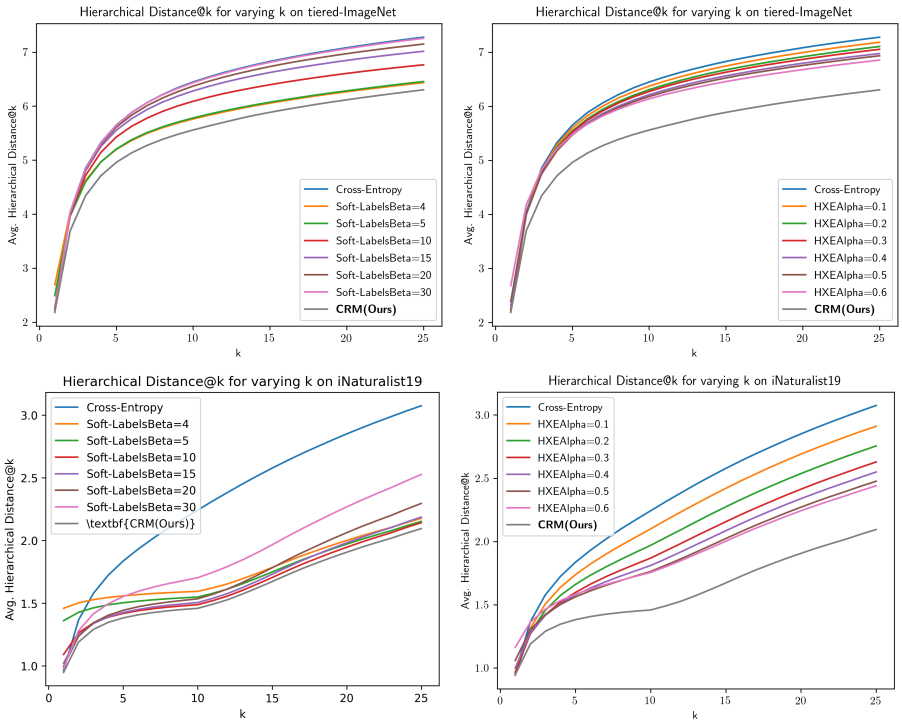


Figure 4: Mistake distribution for all values of Softlabels across β (left) and HXE across α (right) as well baselines including cross-entropy and CRM for reference across tieredImagenet-H (top) and iNaturalist-19 (bottom).

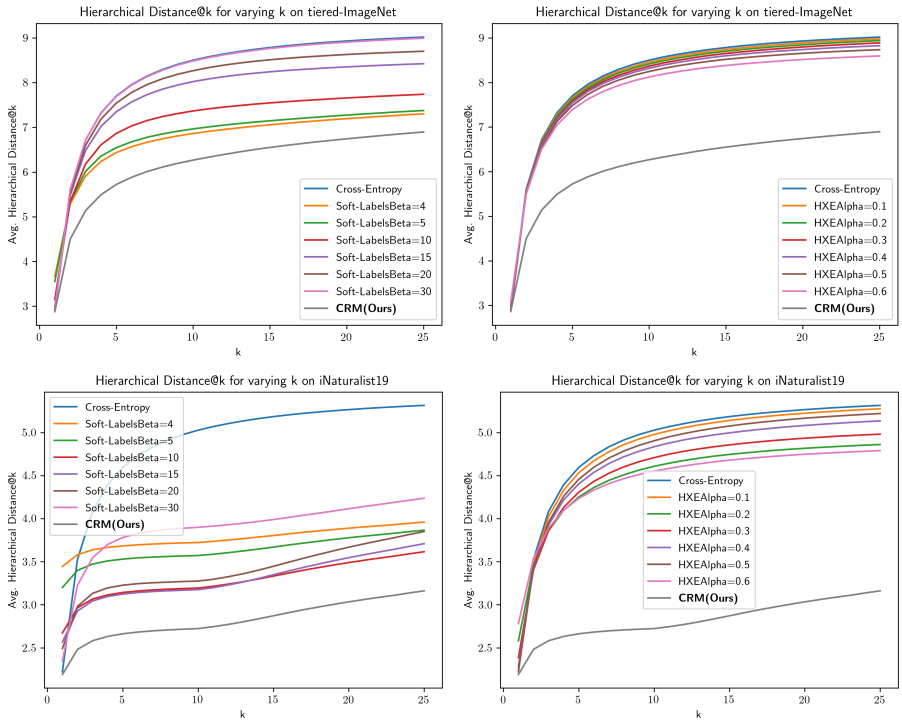
We first analyze the change of the distribution of mistakes across Softlabels (left column) and HXE (right column) models in Figure 4 with results in tieredImageNet-H (top row) and iNaturalist-H (bottom row). As we try to align the model better with the hierarchy by decreasing β and increasing α, we observe the same trends with better alignment to hierarchy – their mistakes are equally bad compared to cross-entropy near the right end (high severity), and they increasingly make more mistakes in the left end (lower severity), lowering the average over mistakes but not making better mistakes. This gives evidence that the models decrease their mistake severity (shown in legend) by largely making lots of additional mistakes and not making *better mistakes* as their primary claim.

A.2 VARIATION OF CLASS RANKING WITH HYPERPARAMETERS

We similarly analyze the variation of ranking classes measured by average hierarchical distance@k for Softlabels and HXE with different hyperparameters. We present our results in Figure 5, by varying Softlabel with different β values on the left and HXE with different α values on the right. We observe that the previously chosen values $\beta = 4$ and $\alpha = 0.6$ perform the best in ranking in all cases except $\beta = 4$ in iNaturalist where $\beta = 10$ performs the best, which we updated in Figure 3. We observe there that CRM still outperforms these methods by large margins.



(a) Ranking performance on the original hierarchy



(b) Ranking performance on a randomly class-shuffled hierarchy.

Figure 5: Average hierarchical distance@k with varying k across different hyperparameters of Soft-labels (left) and HXE (right). We observe our selected hyperparameters (alpha=0.6) and (beta=4) perform the best among others.

A.3 Variation of Calibration Across Hierarchy

We can additionally calculate ECE across levels in the hierarchy by sequentially shrinking leaf nodes from the maximum depth (corresponding to flat classification). The ECE at depth i (from root

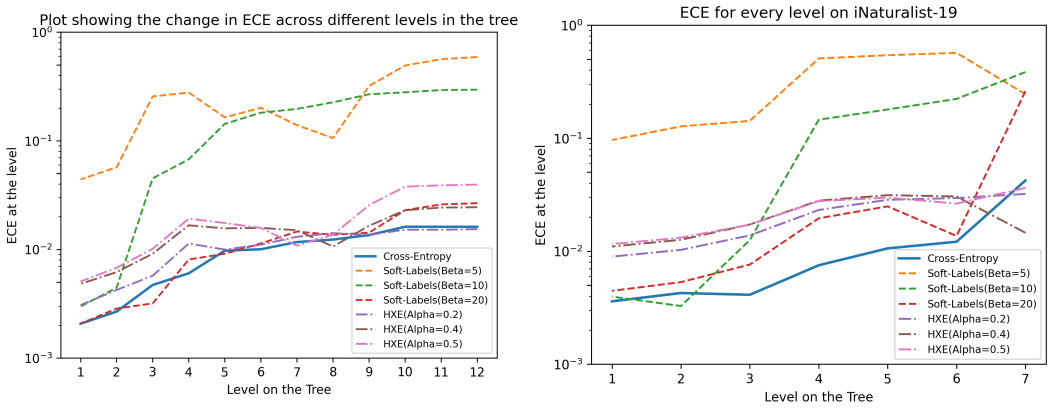


Figure 6: Calibration across various levels in the hierarchy for different models

assigned depth 0) is defined as obtaining probabilities for nodes with most depth which are at most at level i. Their probabilities are obtained by summing up probabilities of their children nodes (level $> i$). We present the results in Figure 6, where we observe that overall the calibration increases as you go up the hierarchy. The cross-entropy baseline shows a consistent decreasing trend, but the other models have aberrations where the calibration error increases going up the hierarchy, especially the models with high calibration errors.

A.4 Increasing Beta for Soft-Labels on iNaturalist19

We also experiment with increasing β for Soft-Labels on the iNaturalist19 dataset by trying the values of 50, 75, 100 and 200 respectively. These results are shown in Table 2. We can see that increasing Beta does not significantly improve top-1 accuracy while worsening the hierarchical distance metrics. We additionally observe this in Figure 2, where as we increase β the graph shoots up with little leftward shift.

Model	Accuracy	Hier. Distance@1	Hier. Distance@5	Hier. Distance@20
Soft-Labels(Beta=30)	0.5819± 0.001	0.976± 0.006	1.554±0.006	2.267±0.004
Soft-Labels(Beta=50)	0.5880 ± 0.003	1.003 ± 0.01	1.879± 0.003	2.823±0.011
Soft-Labels(Beta=75)	0.5876±0.002	0.997±0.005	1.916±0.005	2.909±0.017
Soft-Labels(Beta=100)	0.585±0.001	1.01±0.004	1.926±0.005	2.93±0.014
Soft-Labels(Beta=200)	0.5877±0.004	0.999±0.009	1.924±0.005	2.935±0.012
Cross-Entropy	0.5962±0.002	0.96±0.004	1.836±0.003	2.841±0.01

Table 2: Increasing Beta for Soft-Labels on iNaturalist19 beyond 30 does not increase accuracy while significantly worsening the hierarchical distance metrics.