7. Appendix

7.1. Schematics for CNN-based models

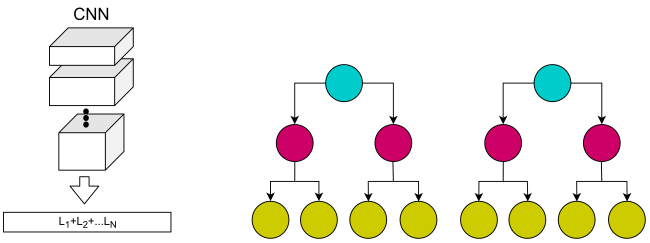


Figure 9: Model schematic for the hierarchy-agnostic classifier. The model is a multi-label classifier and does not utilize any information about the presence of an explicit hierarchy in the labels.

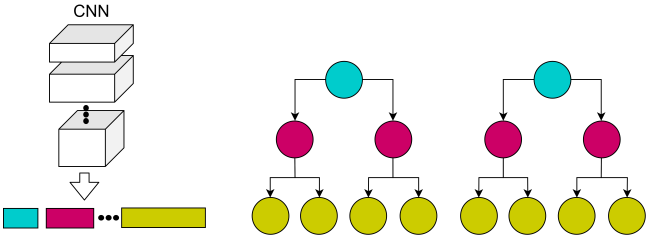


Figure 10: Model schematic for the per-level classifier (=L N_i-way classifiers). The model use information about the label-hierarchy by explicitly predicting a single label per level for a given image.

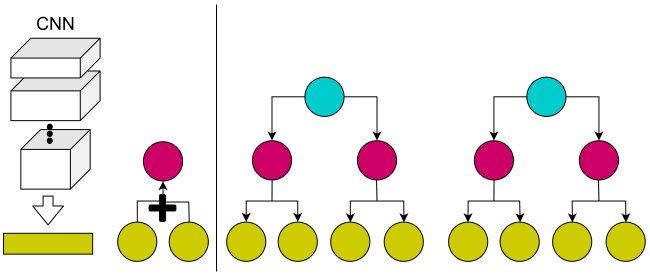


Figure 11: Model schematic for the Marginalization method. Instead of predicting a label per level, the model outputs a probability distribution over the leaves of the hierarchy. Probability for non-leaf nodes is determined by marginalizing over the direct descendants. The Marginalization method models how different nodes are connected among each other in addition to the fact that there are L levels in the label-hierarchy.

7.2. Performance metrics

True positive rate. True positive rate (TPR) is the fraction of actual positives predicted correctly by the method.

$$\text{TPR} = \frac{tp}{totalPositives} \tag{10}$$

True negative rate. True negative rate (TNR) is the fraction of actual negatives predicted correctly by the method.

$$\text{TNR} = \frac{tn}{totalNegatives} \tag{11}$$

Precision. Precision computes what fraction of the labels predicted true by the model are actually true.

$$\text{P} = \frac{tp}{tp + fp} \tag{12}$$

Recall. Recall computes what fraction of the true labels were predicted as true.

$$\text{R} = \frac{tp}{tp + fn} \tag{13}$$

F1-score.

$$\text{F1} = \frac{2 * P * R}{P + R} \tag{14}$$

Hit@k.

$$\text{Hit@K} = \frac{1}{N} \sum_{i=1}^{N} 1[\text{label}_i^{\text{gt}} \in \text{SortedPredictions}(i)] \tag{15}$$

where, $\text{SortedPredictions}(i) = \{\text{label}_0^{\text{pred}}, \text{label}_1^{\text{pred}}, ..., \text{label}_{k-1}^{\text{pred}}, \text{label}_k^{\text{pred}}\}$ is the set of the top-K predictions for the i-th data sample. **Macro-averaged score.** A macro-averaged score for a metric is calculated by averaging the metric across all labels.

$$\text{M-metric} = \frac{1}{N} \sum_{i=1}^{N} \text{metric}(\text{label}_i) \tag{16}$$

Micro-averaged score. A micro-averaged score for a metric is calculated by accumulating contributions (to the performance metric) across all labels and these accumulated contributions are used to calculate the micro score.

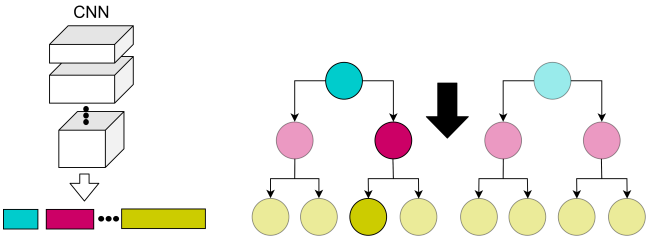


Figure 12: Model schematic for the Masked Per-level classifier. The model is trained exactly like the L N_i-way classifier. While predicting, one assumes the model performs better for upper levels than lower levels. Keeping this in mind, when predicting a label for a lower level, the model's prediction for the level above is used to mask all infeasible descendant nodes, assuming the model predicts correctly for the level above. This results in competition only among the descendants of the predicted label in the level above.

7.3. ETHEC dataset

The ETHEC dataset [1] contains 47,978 images of the "order" *Lepidoptera* with corresponding labels across 4 different levels. According to the way the taxonomy is defined, the *specific epithet* (species) name associated with a specimen may not be unique. For instance, two samples with the following set of labels, (*Pieridae*, *Coliadinae*, *Colias*, *staudingeri*) and (*Lycaenidae*, *Polyommatinae*, *Cupido*, *staudingeri*) have the same *specific epithet* but differ in all the other label levels - *family*, *subfamily* and *genus*. However, the combination of the *genus* and *specific epithet* is unique. To ensure that the hierarchy is a tree structure and each node has a unique parent, we define a version of the database where there is a 4-level hierarchy - *family* (6), *subfamily* (21), *genus* (135) and *genus* + *specific epithet* (561) with a total of 723 labels. We keep the *genus* level as according to experts in the field, information about genera helps distinguish among samples and result in a better performing model.

7.4. HAB details

Here we discuss the details of having a single threshold for every label or a common threshold for all labels in a multi-label classification setting. Here we observe the maximum and minimum labels predicted by the multi-label model across the whole dataset. We also look at the mean and standard deviation of the number of labels predicted.

7.4.1 Per-class decision boundary (PCDB) models

The ill-effects of such free rein are reflected in Table 3. Models with a high average number of predictions, especially the per-class decision boundary (PCDB) models, have high recall as they predict a lot more than just 4 labels for a given image. Predicting the image's membership in a lot of classes improves the chances of predicting the correct label but at the cost of a large number of false positives. The (min, max), $\mu \pm \sigma$ column clearly shows the reckless behavior of the model predicting a maximum of 718 labels for one such sample and 451.14 $\pm$ 136.69 on average for the worst performing multi-label model in our experiments.

7.4.2 One-fits-all decision boundary (OFADB) models

The one-fits-all decision boundary (OFADB) performs better than the same model with per-class decision boundaries (PCDB). We believe that the OFADB prevents over-fitting, especially in the case when many labels have very few data samples to learn from, which is the case for the ETHEC database. Here too, the nature of the multi-label setting allows the model to predict as many labels as it wants however, there is a marked difference between the (min, max), $\mu \pm \sigma$ statistics when comparing between the OFADB and PCDB. The best performing OFADB model predicts 3.10 $\pm$ 1.16 labels on average. This is close to the correct number of labels per specimen which is equal to the 4 levels in the label hierarchy.

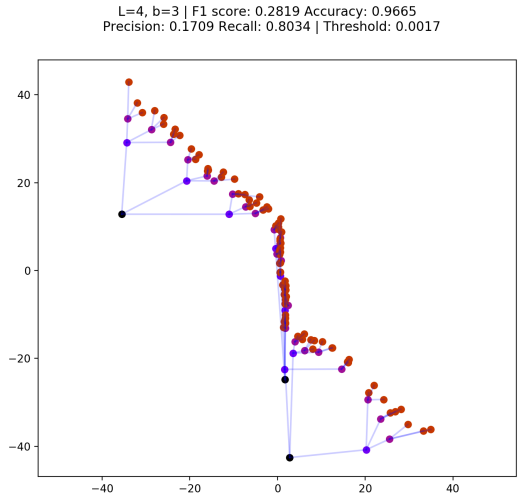
7.4.3 Loss reweighing and Data re-sampling

Both data re-sampling and loss re-weighing remedy imbalance across different labels but via different paradigms. Instead of modifying what the model sees during training, reweighing the loss instead penalizes different data points differently. We choose to use the inverse-frequency of the label as weights that scale loss corresponding to the data point belonging to a particular label.

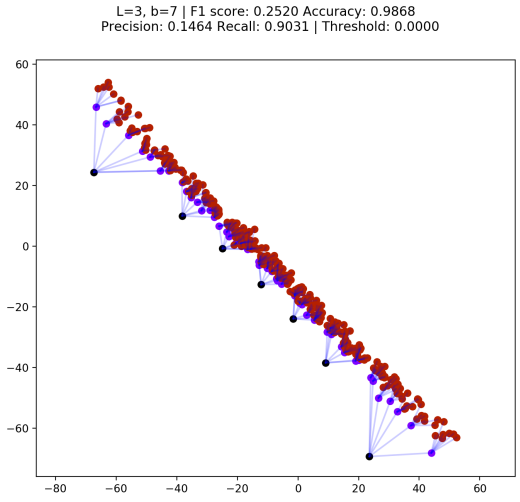
re-sampling involves choosing some samples multiple times while omitting others by over-sampling and under-sampling. We wish to prevent the model from being biased by the population of data belonging to a particular label. We perform re-sampling based on the inverse-frequency of a label in the *train* set. In our experiments re-sampling significantly outperforms loss reweighing confirming the observations made in [28].

cw	rs	m-P	m-R	m-F1	(min, max), $\mu \pm \sigma$
ResNet-50 - Per-class decision boundary					
✗	✗	0.0355	0.7232	0.0677	(3, 351), 81.4 $\pm$ 69.5
✗	✓	0.7159	0.7543	**0.3718**	(0, 13), 4.2 $\pm$ 2.1
✓	✗	0.0077	**0.8702**	0.0153	(84, 718), 451.1 $\pm$ 136.7
✓	✓	0.0081	0.7519	0.0161	(33, 714), 370.0 $\pm$ 120.6
ResNet-50 - One-fits-all decision boundary					
✗	✗	0.9324	0.7235	**0.8147**	(0, 7), 3.1 $\pm$ 1.2
✗	✓	**0.9500**	0.6564	0.7763	(0, 5), 2.8 $\pm$ 0.6
✓	✗	0.2488	0.2960	0.2704	(4, 9), 4.8 $\pm$ 0.8
✓	✓	0.1966	0.3800	0.2591	(4, 10), 7.7 $\pm$ 0.6

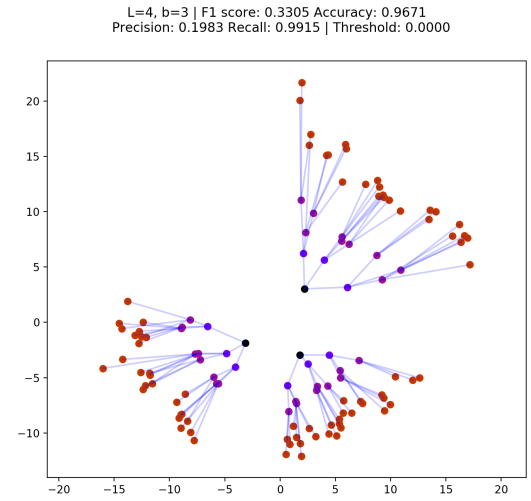
Table 3: Performance metrics for the HAB on the ETHEC dataset. The models used in this experiment are pre-trained on the 1000-class ImageNet data set. All weights are updated with a learning rate of 0.01, a batch-size of 64 and input spatial dimensions are 224x224 for 100 epochs. *P*, *R* and *F1* represent Precision, Recall and F1-score; *cw* and *rs* represent class weight and re-sampling. *m* are micro-averaged metrics. The top performing models are in boldface. Since, the model can predict any number of labels (between 0 and N_{total}), the table includes the minimum and the maximum number of labels predicted *(min, max)* as well as the number of labels predicted on average $\mu \pm \sigma$. These statistics, like the rest, are calculated for samples in the *test* set.



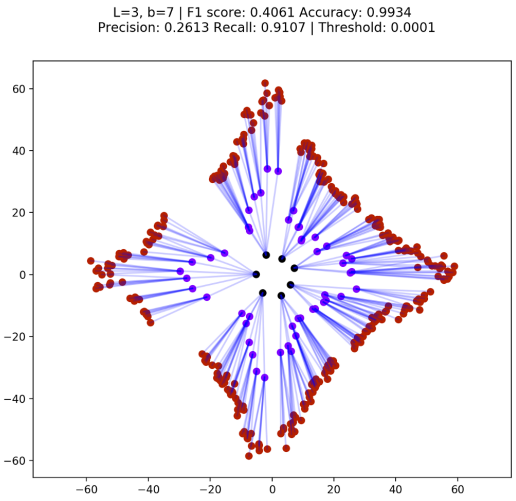
(a) Order-embeddings L=4, b=3



(b) Order-embeddings L=3, b=7



(c) Euclidean cones L=4, b=3



(d) Euclidean cones L=3, b=7

Figure 13: We embed 2 different toy graphs. One with 4 levels and a branching factor of 4 and another one with 3 levels and a branching factor of 7. The model is trained for 1000 epochs with Adam (learning rate of 0.01). The toy graphs are embedded using both order-embeddings and euclidean cones in $\mathcal{R}^2$. We draw an edge between each node that is connected in the original in order to better visualize the embedding quality. Nodes from different levels are colored differently. The illustrations show the levels and branching factor, the edges are split into *train*, *val* and *test* and report F1-score, precision, recall and accuracy; and the threshold to decide if a pair of nodes have a directed edge or equivalently if they are hypernyms.

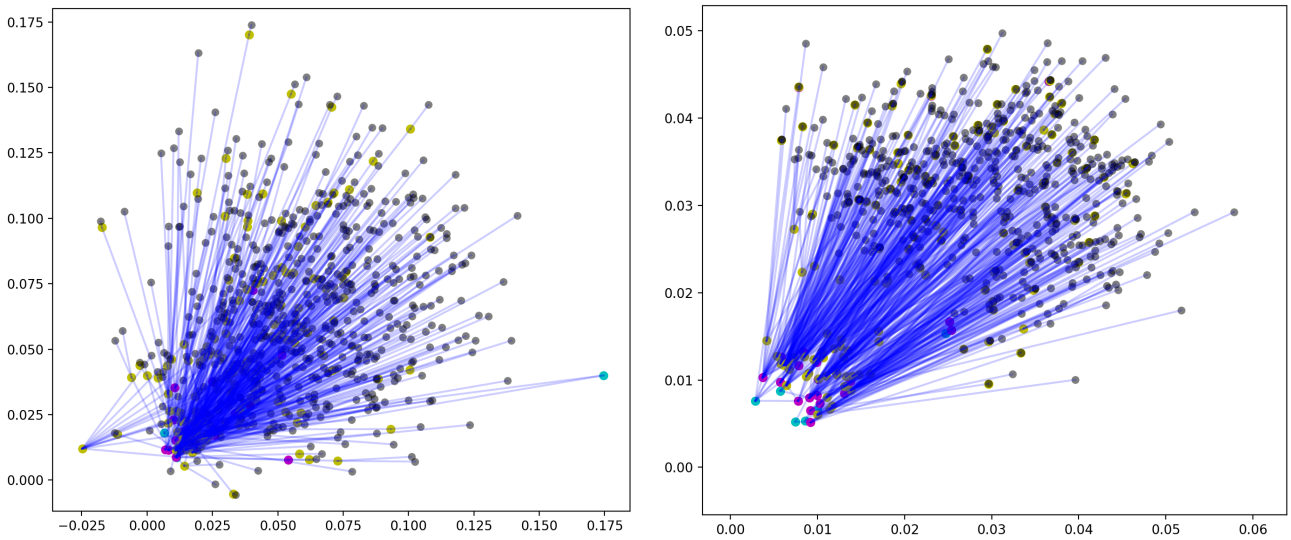
(a) *Aporia crataegi* [ENT01_2017_03_27_007897]

(b) *Parnassius stubbendorfii* [ENT01_2018_03_09_132877]

(c) *Parnassius delphius* [ENT01_2018_03_09_133076]

(d) *Parnassius delphius* [ENT01_2018_03_09_133091]

Figure 14: Both semantic similarity and visual similarity are required to perform tasks relating to image understanding. Here, we see an example from the ETHEC dataset [1]. At first glance, (a) and (b) look like they belong to the same class and so do (c) and (d) considering the visual similarities. However, this is not so straight-forward as (a) and (b) belong to two separate genera and species but have a really low inter-class variance. On the other hand, (b), (c) and (d) all share the same genus *Parnassius* but have a larger intra-class variance than (a) and (b). This demonstrates how visual similarity might not imply semantic similarity and vice-versa.



(a) Hyperbolic Cones 100-D

(b) Hyperbolic Cones 1000-D

Figure 15: Projected visualization of labels embedded using hyperbolic cones in 100 and 1000 dimensions. The cyan nodes represent *family*, the magenta nodes represent *sub-family*, the yellow nodes *genus* and black nodes *genus+species*. This resembles a flower-like shape where the more generic concepts are closer to the origin and at the base of this flower-like shape and most specific concepts at the tip of the petals which forms the periphery are a visible the most (=black nodes).