Contextual Label Smoothing with a Phylogenetic Tree on the iNaturalist 2018 Challenge Dataset

Michael J. Trammell[1], Priyanka Oberoi[1], James Egenrieder[2], John Kaufhold[1]
[1]General Dynamics Mission Systems' Deep Learning Analytics Center [2]Virginia Tech

Abstract

Recognition of fine-grained visual categories (FGVC) in the natural world is a long-tailed problem, meaning recognizers must accurately recognize a large diversity of categories and most of those categories will naturally have limited training data, increasing the likelihood of overfitting in these many limited training data categories. The iNaturalist 2018 Challenge aimed to benchmark the state of the art performance on species identification from a photo, where the long-tailed aspect of training is compounded by the visual similarity of many species. We demonstrate a new state of the art on the iNaturalist 2018 Challenge with Contextual Label Smoothing (CLS). CLS extends label smoothing to narrow the list of categories smoothed to only those within the same branch of a phylogenetic tree. CLS regularization improves performance significantly—the best publicly reported Top3 error reported on the iNaturalist 2018 Challenge was approximately 13%, which we improve to 12% with an ensemble of CLS networks trained with dynamic minibatching and additional inference windows. We present evidence that a 1% improvement on the FGVC iNaturalist 2018 Challenge test score (public score) represents over a 5 sigma improvement (test score stdev = 0.17 %) over the former state of the art.

1. Introduction

The problem of fine-grained visual categorization (FGVC) has been studied across many domains with many image datasets, including FGVC-Aircraft [1], Stanford Cars [2], motorcycles [3] and shoes [4], among others. Many FGVC datasets of the natural world collect plant and animal species [5], birds [6], vegetables and fruits [7], plants [8], and dog breeds [9] to identify, among others. One of the largest and most imbalanced public datasets of natural imagery with these long-tailed FGVC challenges is the iNaturalist 2017 Challenge dataset, which the iNaturalist 2018 Challenge dataset made even larger and more imbalanced [10]. The iNaturalist 2018 Challenge training and validation data was made available by iNaturalist [11] and the competition was hosted on kaggle [12], which scored submissions on an unseen test set. Organizersof the iNaturalist 2018 Challenge aimed to:

> push the state of the art in automatic image classification for real world data that features a large number of fine-grained categories with high class imbalance. … The dataset features many visually similar species, captured in a wide variety of situations, from all over the world. [12]

1.1. iNaturalist 2018's Long Tails

We call the most represented training categories in the iNaturalist 2018 Challenge data the "head" and the least represented categories the "tail" of the distribution (as in [13]). Recent work [13] has highlighted key properties of FGVC of long-tailed distributions: (1) there are many categories (2) most of the categories have limited training data (the tail categories) (3) error rates improve only when more labeled data is made available for the tail categories and (4) additional training data for the head categories does not appreciably improve overall performance (i.e. the network does not transfer learn from the head categories to the tail categories). On the iNaturalist 2018 Challenge data, approximately 10% of the categories (~800) comprise the head of the distribution, where each category has between 100 and 1000 training examples, and 75% of the categories (~6000) comprise the tail categories, where each category has between 2 and 30 training examples.

The prohibitive cost curve associated with generating sufficient training data for long-tailed FGVC applications to reach a threshold accuracy is sketched in [13]:

> Collecting the eBird dataset took a few thousand motivated birders about 1 year. Increasing its size to the point that its top 2000 species contained at least 10^4 images would take 100 years.

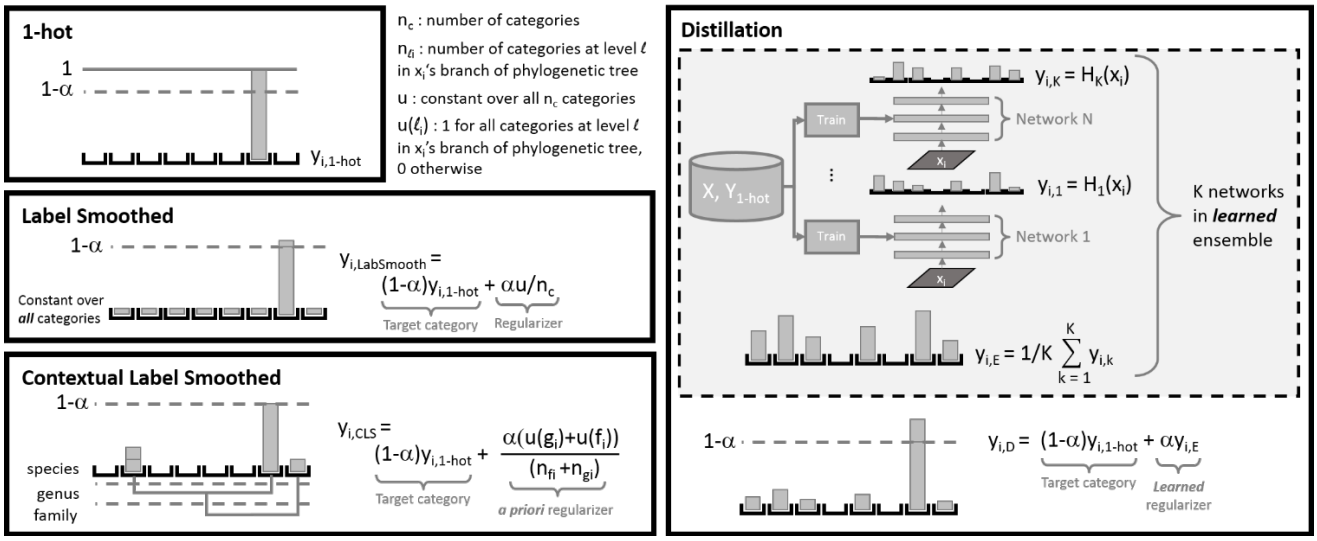

Figure 1: Contextual Label Smoothing (CLS) label form compared to related label smoothing forms: 1-hot encodings are *sparse labels* (top left). For example, for x_i only one nonzero value in $y_{i,1\text{-hot}}$ is the target category and all others are 0s. 1-hot labels incorporate no regularization (either via a prior or learned post hoc from ensembling). Label smoothing (middle left), contextual label smoothing (bottom left), and distillation (right) all incorporate into their *full label vectors* some degree of regularization. In label smoothing, the regularizer is very weak but effective—$y_{i,LabSmooth}$ spreads out a small constant residual contribution of α/n_c to every category (where n_c is the number of categories and u is a constant over all categories). In distillation, K classifiers are first trained with the 1-hot labels—the temperature-relaxed logits from the output layers of these K classifiers are then combined into a learned regularization term that is scaled and added to the 1-hot target category to form $y_{i,D}$. The distilled version's regularized $y_{i,D}$ has dense structure reflecting similarities among categories learned from the ensemble. Our method, contextual label smoothing (CLS), requires no learning as in distillation, and encodes label similarity from a phylogenetic tree into $y_{i,CLS}$. The number of categories shared at the genus and family level are n_g and n_f, respectively. The notation $u(\ell_i)$ takes the value 1 for all categories shared at the ℓ level with the target category for x_i.

1.2. Label-efficient Approaches to Long Tails

For this reason, we seek more label-efficient approaches that incorporate context to address long-tailed FGVC challenges. Our aim is to efficiently encode in the labels, themselves, information that mitigates the performance degradation to tail categories stemming from limited training data. In the spirit of [14], in our proposed Contextual Label Smoothing (CLS), we allow tail categories to learn from training data pooled from similar categories as defined on a hierarchy (a phylogenetic tree) with label vector encodings (i.e. soft targets), but unlike [14], we do not *learn* these relationships (which incurs a computational cost), but encode them directly with the phylogenetic tree [15] as the prior. The labels in the CLS approach are diagrammed in contradistinction to 1-hot encoding, label smoothing and distillation in Figure 1.

While 1-hot label encodings (where one category is assigned a 1 and all others 0s) of categories have become common in mainstream object recognition [16]–[18], we argue these 1-hot independent category labels are label-inefficient—they do not effectively share informative training examples across similar labels; they are also overconfident—they make deep networks more susceptible to overfitting, especially on categories with limited training data.

Two simple relaxations of the 1-hot label encoding to better calibrate confidences in FGVC have been shown to improve (A) the robustness of the learned networks [19] and (B) the ability to learn more accurate tail categories post hoc from ensembles with limited training data [20]. In both label smoothing and distillation, the training labels are not 1-hot, but full, and retain some nonzero dot product from label vector to label vector. Inspired by both label smoothing and distillation, we demonstrate that contextual label smoothing (CLS), like hierarchical semantic encoding (HSE), can improve recognition rates on long-tailed FGVC problems.

1.3. CLS is Hierarchical Label Smoothing

Uniform label smoothing is an a priori decision to spread contributions from a target label over all other labels (Figure 1) *uniformly*, which has the effect of penalizing overconfident predictions [19]. Intuitively, label smoothing allows *all* other categories to contribute training data to a target category, and spreading over *all* categories may spread the label information too thinly to efficiently transfer learn (as observed in [13]). In this work, we extend label smoothing to spread contributions from a label only within a branch of a phylogenetic tree provided a priori, not smooth over all other categories. Briefly, CLS exploits the phylogenetic tree to be more judicious about the label smoothing prior. Practically, we do not label smooth a

training example of a humpback whale to have a nonzero contribution to learning a monarch butterfly category, but we do label smooth a training example of a gluphisia moth to have a nonzero contribution to learning the monarch butterfly category. While branches of phylogenetic trees are not always indicative of visual similarity, we empirically demonstrate enough are to justify use of this prior.

1.4. CLS is Distillation with a Prior

Where distillation is an *empirical* post hoc approach to encode similarity into label vectors [20], our CLS work can be viewed as a form of *a priori* distillation (Figure 2). Specifically, in distillation, an ensemble of classifiers are trained (from 1-hot labels). After learning, the (temperature-relaxed) logits of this ensemble empirically develop higher values for both the true category and visually similar categories. These post-hoc logits from this ensemble are added to the true 1-hot (hard targets) label for every training example in a downstream distillation of the ensemble. Intuitively, if only a handful of other classes are visually similar to the true class, when downstream training occurs with these distilled label vectors (soft targets), every one of those visually similar categories will contribute non negligibly to the training set for the original 1-hot target label. In this way, distillation *reuses* training examples from *other* categories to train to recognize the target categories most visually similar to it—this makes distillation a more label-efficient strategy than 1-hot encoding (Figure 2). CLS is an *a priori* version of distillation, encoding similarity as shared parentage on a phylogenetic tree provided without any downstream ensemble training (as are *learned* in either distillation or HSE).

1.5. Fine-Tuning with more Balanced Categories

On similar FGVC tasks [21], better performance was obtained by further fine-tuning on a more balanced subset of FGVC validation data with a small learning rate. Improvements on head categories with ≥100 training images were relatively small compared to tail categories with <100 training images. This provides an empirical rationale for fine-tuning on validation data more uniformly distributed over categories to improve performance on underrepresented tail categories. We incorporate this type of fine-tuning into CLS.

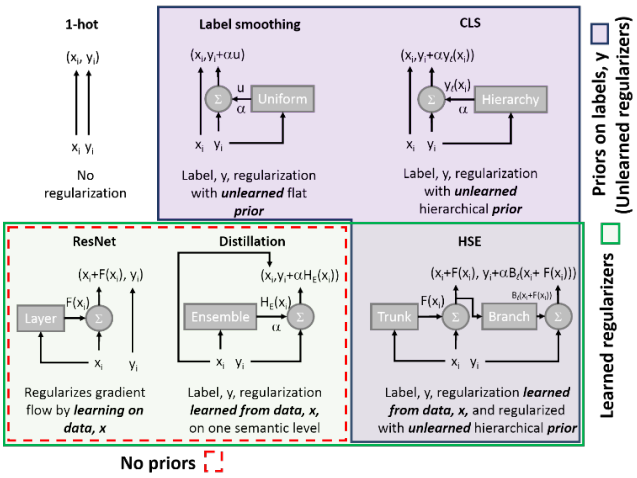

Figure 2: Residual connection blocks regularize data and labels: Five deep learning-based conceptual "blocks" to remedy well known overfitting and vanishing/noisy gradient issues of 1-hot label encodings (top left) are diagrammed. The well-known ResNet architecture ([22] bottom left) adds copies of the data to regularize gradients—this architectural change is common to many of the other methods (both the trunk and branch networks of HSE [14] implement ResNet models, e.g.). Label smoothing ([19], top middle) can be viewed as a residual connection between a 1-hot y_i, and an *unlearned uniform* prior. This same strategy inspires this work on CLS (top right), but we use an *unlearned hierarchical* prior. Distillation (bottom middle) can be viewed as a residual connection between a 1-hot y_i and a *learned* soft target (the posterior distribution from learning an ensemble was used in [20]). The most general form of these combinations we have found is the very recent work on HSE (bottom right), which incorporates residual connections *learned* within trunk and branch networks, *learns* to update soft target priors based on an *unlearned* hierarchical prior, and combines these with residual connections at each level of the hierarchy.

1.6. Contributions

We make a number of original contributions in this work:

- **Contribution 1: New State of the Art on the iNaturalist 2018 Challenge.** We demonstrate a new state of the art result on the long-tailed FGVC iNaturalist 2018 Challenge Data [11]. We estimate through a prediction set that this new state of the art outperforms the prior state of the art by greater than 5 σ on the unseen test data.
- **Contribution 2: CLS works best with uniform sampling over categories.** In contradistinction to natural sampling advocated in [13], CLS benefits from uniform sampling of categories in training.
- **Contribution 3: CLS improves ensemble performance more per marginal network than other methods.** Given a choice between adding a network trained with some other technique to increase model diversity in an ensemble, adding another CLS-trained network is a better choice. This

clarity can reduce the significant hyperparameter search and tuning costs over an ensemble.

- **Contribution 4: Larger Input Images Improve Performance.** We confirm empirically that larger input size images, which have recently been shown to improve performance on the same task without CLS [21], also improves performance of CLS.

2. Related Work

2.1. Deep Learning from 1-hot Labels

Since 2012 [17], deep networks have dominated the state of the art in object recognition on images, maturing year over year to include new network architectures [18], [22] until the performance of deep networks was on par with or better than human performance on a standard benchmark [23]. While significant attention has been paid to data augmentation [17], transfer learning [24], and new architectures [18], [22], less work has been devoted to improving the 1-hot labels [19], [20], themselves, for training data. This work addresses improvements to the design of labels, themselves.

2.2. Label Vector Benefits

Work on improved label vector engineering includes label smoothing [19] and distillation [20], among others (Figure 1 & Figure 2). Label smoothing is a simple method that incorporates a prior to drive deep networks to solutions with higher posterior entropy. Distillation, while originally proposed as a method to make networks smaller (in memory and computational cost of inference), has also demonstrated regularization and adversarial example defense properties.

Work on Hierarchical Semantic Embedding is most similar in spirit to this work, but achieves its goals of incorporating category similarity through a trunk and branches architecture over a collection of 1-hot label vectors at various semantic levels (from coarse to fine) [14]. Similar to distillation, it adds a predicted category score vector (i.e. a soft target) from a coarser level to the 1-hot label vector at the next finer level. FGVC results on three natural datasets, CUB [6], butterflies [14], and VegFru [7], demonstrate the value of HSE. HSE outperforms 17 other state of the art methods on CUB. The strategies employed in HSE appear to be more general than the simpler unlearned CLS prior proposed here (Figure 2), but HSE benefits have not yet been demonstrated on as large a dataset as that of the iNaturalist 2018 Challenge, which has >25x more fine-grained categories and >100x larger category imbalance, which are critically relevant aspects of long-tailed FGVC challenges [13].

Importantly, none of the datasets used to demonstrate HSE has more than 292 fine grained categories (compared to 8,142 for the iNaturalist Challenge 2018 data), with CUB's 200 categories separated into 122 genera, 37 families, and 13 orders, where 75% of CUB categories fall into the head category with 60 training images/category, and where all categories have at least 41 training images, for a max class imbalance of 1.5 (compared to 500 for the iNaturalist 2018 Challenge). The authors' new butterfly dataset also only contains 200 categories. This smaller scale of the FGVC challenges addressed by nascent exploration of HSE is encouraging, but qualitatively smaller scope than evaluation on iNaturalist Challenge 2018 data, which is an open dataset and more comprehensive than those datasets HSE authors chose to evaluate on.

Interestingly, HSE training develops learned attention mechanisms, making a convincing case that without specifically labeled parts, HSE can learn features that exploit part-based attention to discriminate in FGVC, as was demonstrated to be critical for natural FGVC in other work [25]. The critical difference between the label vectors in HSE and our CLS work is that all of our label hierarchy information is encoded in label vectors without branches CLS is a de facto *flat* prior that is *not learned* and is modularly separable from the architecture—i.e. there is only one label vector for each example in CLS, whereas HSE requires different label vectors at different levels in the architecture, increasing hyperparameter search costs.

2.3. Long-tailed FGVC Implications

The properties and implications of long-tailed distributions in FGVC have been summarized with convincing evidence [13] that (1) statistics of natural image categories are long-tailed, (2) more training data for head categories does not improve performance on tail categories, and (3) natural sampling of categories in training minibatches outperforms uniform sampling over categories. In [13], authors used standard 1-hot label encodings and sampled "naturally" (as opposed to uniformly) during training. The argument for natural over uniform sampling was empirical—results demonstrated both head and tail category performances both improved more with natural sampling. In contrast, we argue that the thoughtful vector encoding of labels with CLS overturns that guidance on sampling method (Contribution 2). Choosing training minibatches from CLS with uniform sampling over categories outperforms natural sampling. Authors conclude: "As a community we need to face up to the long-tailed challenge and start developing algorithms for image collections that mirror real-world statistics" which outlines the core motivation for this work [13].

2.4. Prior State of the Art iNaturalist Performance

The iNaturalist 2017 Challenge was won by Google (*GMV*, for Google Mountain View, on the leaderboard) with a Top5 error rate of less than 5% with an ensemble of InceptionV3 and InceptionV4 models trained at both

299x299 and 560x560 input image sizes, and subsequently fine-tuned on a balanced subset of the data left out of the test set [21]. The fine-tuning on balanced data boosts performance on tail categories of the dataset [1] and during inference 12 crops outperformed inference on a single prediction for the entire image.

Compared to the iNaturalist 2017 Challenge, the iNaturalist 2018 Challenge reduced the number of training images provided from 675,170 to 461,939, increased the number of classes from 5,089 to 8,142, and perhaps most significantly, provided a complete taxonomy for each class. A team from Dalian University won the 2018 challenge with a Top3 error rate of 13% [12]. Their winning ensemble consisted of 12 ResNet-152 models trained at both 320x320 and 392x392 input image sizes, six of which used matrix power normalized covariance pooling of the last layer of convolutional features [2].

3. Training Methodology

3.1. Training and Validation Data Set Splits

The iNaturalist 2018 Challenge data includes three mutually exclusive data sets: training, validation, and test data, each containing photos drawn from one of 8,142 species categories distributed over 4412 genera. The training data distribution is imbalanced, with the most represented species, *Branta canadensis* the "Canada goose", having 1000 training examples, whereas the least represented species in the training data is the *Spatula clypeata,* the "Northern shoveler duck," with only two training examples. The validation set is uniformly distributed over species, with three validation images per species. The test set labels are not provided to entrants, but entrants can submit Top3 label lists for each of the 149k test images to be scored on a Top3 error rate that is blind to which examples were marked correctly or incorrectly. In the development that follows, 2/3 of the validation data (two photos per species) is used for validation fine-tuning and 1/3 of the validation data (one photo per species) is used as the test score prediction set. In "vanilla" label smoothing [19], we assign the target label 0.8 and distribute the remaining 0.2 of that example to all other 8,141 categories in the label vector.

3.2. Initialization with Pretrained Networks

Closely following the winning *GMV* entrant from the iNaturalist 2017 Challenge, we start from an IRV2 and IV4 pretrained on ImageNet [18], [22]. These two network architectures are the starting points for training across all input sizes (299x299 and 598x598) and label smoothing methods (1-hot, vanilla label smoothing, and CLS). As in *GMV*, for each network in an ensemble, we strip the final layer of ImageNet-1K classes from the pretrained network and replace it with the iNaturalist 2017 output layer of 5,089 categories and sample minibatches of 32 images per minibatch without replacement from all training examples (we trained on 4 GPUs in parallel for an effective minibatch size of 128 for the IRV2 model and 6 GPUs in parallel for an effective minibatch size of 192 for the IV4 model). We fine-tuned on the iNaturalist 2017 training data for {80, 84} epochs for {IRV2, IV4}. We then fine-tuned on 90% of the iNaturalist 2017 validation data for {30, 14} epochs for {IRV2, IV4} using {8, 4} GPUs for effective minibatch sizes of {256, 128}. We used SGD with an initial learning rate of 0.018 and momentum=0.9 in the first round of training for the IRV2 model, reducing the learning rate by 10% every {8,6} epochs for {IRV2, IV4}. We used RMSProp for all other training. The second round of training began with learning rates of 0.002 for the IRV2 model and 0.001 for the IV4 model, and the training rate was multiplied by 0.9 every 10 epochs. Note that all minibatches in this pretraining were sampled naturally (as opposed to uniformly with replacement).

3.3. Base Fine-Tuning on iNaturalist 2018 Challenge Data

We strip the final layer of iNaturalist Challenge 2017 categories from each pretrained network and replace it with the iNaturalist 2018 Challenge output layer with 8,142 categories. When training, we sample minibatches uniformly over categories with replacement (i.e. we sample *uniformly*); this produces minibatches with approximately equal contributions from all 8,142 categories. We train for 1M-1.4M iterations using RMSprop with a base learning rate of 0.0045 in base fine-tuning. We use a batch size of 32. We retain only the model with the highest performance on the validation set, as assessed every 50k iterations.

3.4. Validation Fine-Tuning on iNaturalist 2018 Challenge Data

We fine-tune on the validation fine-tuning set only. The validation fine-tuning regime is identical to the base fine-tuning regime with the exception that training begins with a base learning rate of 0.0002, and continues for only 25k iterations.

3.5. Ensembling

We compute unweighted model average ensemble results from multiple label smoothing methods to conduct a post hoc ablation study via ensemble composition. We rank the performance boosts from different components of the ensembles to assess the benefits of individual components of each ensemble. Ensemble components vary in input image size, network type, and label smoothing type.

3.6. Test Performance Error Analysis

Additional inference windows: When scoring, we include the standard middle, whole image, and four corner inference windows (with LR reflections). As an

approximation to attention, we also include additional inference windows favoring the sides and top of the image calculated based on the aspect ratio of each image, under the assumption that this is where photographers are more

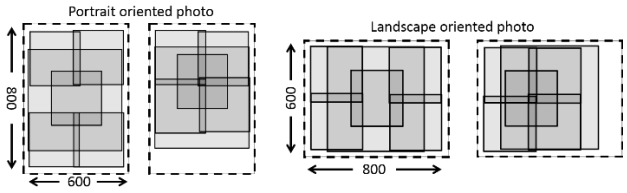

Figure 3: Additional inference windows on a photo. The "standard" twelve inference windows (six with the original image, the same six with the image flipped horizontally) are shown on the left of each orientation. For portrait oriented photos, a second set of inferences is made on twelve more windows biased toward the top of the photo; for landscape oriented photos, the second set of inferences is made on twelve more windows biased toward the left of the photo.

likely to include the subject of the photo.

Test score prediction error rates: Nominally small (<0.5%) differences in Top3 error rates on leaderboards can be difficult to assess the relative merits of. By estimating a test score from the test score prediction data on many model outputs, we estimate a practical error bar on our test performances.

4. Results

The results collected here represent approximately 20,000 total GPU hours across a mix of NVIDIA GTX® 1080s, V100s and Titan® Xs.

For practical perspective, training a single one of our models through to final scoring on 2 GPUs requires approximately 10 days of compute on 299x299 input image sizes and 20 days on 598x598 input image sizes. Note that due to the size of our images and batches, only V100s can be used to train some of our models at our largest image sizes.

4.1. Label Smoothing Method Comparison

We show final iNaturalist 2018 Challenge test score results from kaggle on 299x299 pixel resolution images for the three label smoothing methods: 1-hot (i.e. no label smoothing), vanilla label smoothing (with 0.2 redistributed across all non-target classes), and CLS (with 0.2 redistributed across non-target classes in the same branch of the phylogenetic tree). Results of 3 runs each of {IRV2,IV4} and their ensembles demonstrate CLS outperforms both label smoothing and no label smoothing (i.e. 1-hot) encodings (Figure 4).

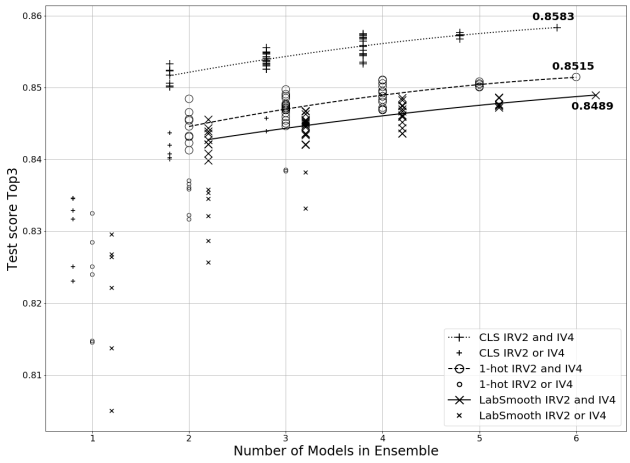

Figure 4: CLS vs. label smoothing vs. 1-hot encodings. CLS networks and ensembles of CLS networks outperform label smoothing and no label smoothing for both IRV2 and IV4 architectures assessed. The iNaturalist 2018 Challenge test scores returned from kaggle for the unseen test set is plotted vs. the number of models ensembled for each label smoothing method. A second-degree spline fit is plotted through the mean score of each set of IRV2 and IV4 ensembles for visual clarity.

4.2. Image Size Ensemble Ablation

We trained ensembles of CLS on both smaller (299x299) and larger (598x598) image input sizes into both IRV2 and IV4. The CLS performance on larger images consistently outperforms CLS trained on smaller images, whether on specific network types or ensembles of the same or different network types (Figure 5).

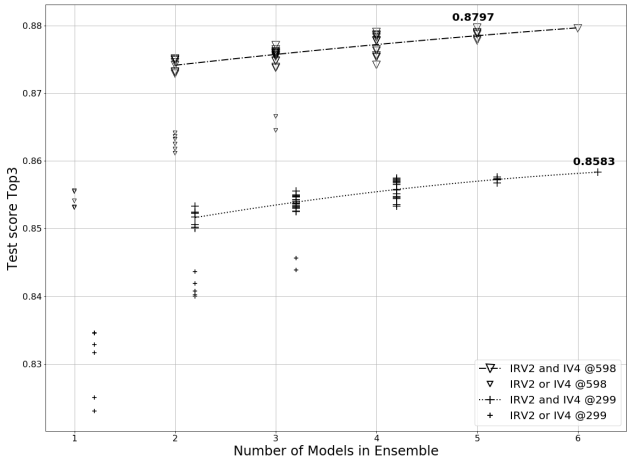

Figure 5: CLS input size comparison. We find that CLS on larger input image sizes (598x598) consistently outperforms CLS on smaller input image sizes (299x299). A second-degree spline fit is plotted through the mean score of each set of IRV2 and IV4 ensembles for visual clarity.

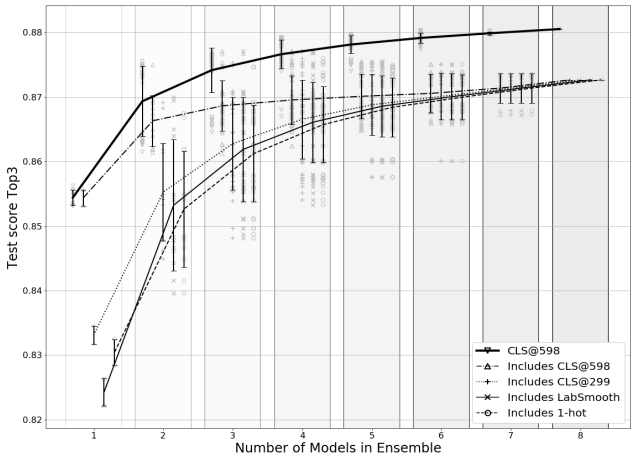

Figure 6: Ensemble Ablation: Only including 598x598 CLS networks in an ensemble with many networks provides state of the art performance with significantly reduced training and hyperparameter search and tuning costs compared to training a larger ensemble with a diversity of networks. Combining CLS networks trained with smaller input image sizes or networks not trained with CLS does not improve performance per network as much as adding another 598x598 CLS network (top curve).

4.3. CLS Ensemble Ablation

Throughout testing, we find that additional CLS networks trained on larger input image sizes (598x598) improve ensembled results the most per additional network in the ensemble (Figure 6). We find that unweighted network type diversity (including networks trained with and without label-smoothing, i.e. 1-hot, IRV2 and IV4 architectures, and smaller input image sizes) do not improve ensemble performance per additional network as much as adding a CLS-trained network at a 598x598 input image size, indicating that CLS with large imagery dominates the potential expected benefit of model diversity in these ensembles. When ensembles contain four or more networks, we observe that adding networks trained with either 1-hot or vanilla label smoothing label vectors can hurt performance.

4.4. Test Performance Error Analysis

We estimate from the iNaturalist 2018 Challenge test score prediction set that our new CLS state of the art result on iNaturalist 2018 Challenge test score has a +/- 0.17% σ error (Figure 7). Our 1.0% improvement over the former state of the art represents a greater than 5 σ improvement over the best prior reported public test score of 0.8693 (compared to our 0.8805) with this estimate of score variability.

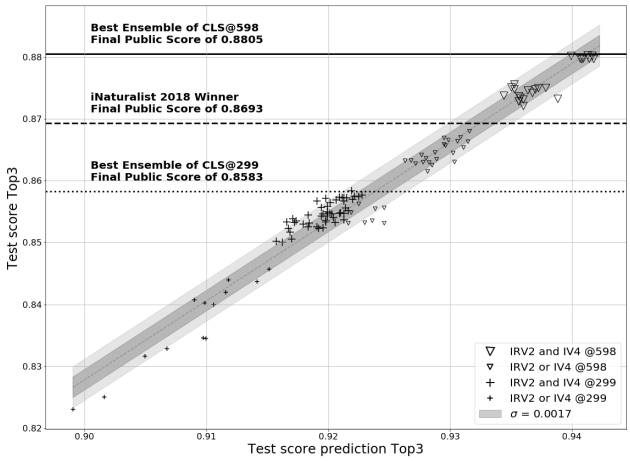

Figure 7: Test Score Error Analysis: By predicting the test error rate on the unseen test data based on a test score prediction subset (1/3) of the validation data we can observe, we develop confidence +/- 1-σ and 2-σ band estimates on the test scores returned by the kaggle server on the unseen test data. The iNaturalist 2018 Challenge final test score winner as reported on the iNaturalist 2018 Challenge leaderboard [12] at 13% Top3 error is shown as a dashed line.

5. Discussion

CLS shares training data among categories: By encoding non-zero values in the label vectors for categories that are not the true target category, CLS learns from a more diverse set of examples than only those formally labeled as the putative target type. In long-tailed FGVC tasks, we expect a number of benefits from this approach.

In theory, for each target tail category, the relatively few training examples of that category with their much larger label vector component (0.8) will anchor the learned latent space of activations for that category with data from that target category. Without full vector labels of any type (i.e. 1-hot labels), the deep network could overfit to these relatively few training examples of the target category (i.e. memorize them), suffering poor generalization with no other information available to prevent this overfitting. Relatively fewer categories (but each with more training examples) from the head of the distribution that share the same branch of the phylogenetic tree as the target category will also contribute to training the target category. These examples will bias the learned latent space of activations for the target tail category to move closer to those related head categories, encouraging transfer learning from the head to the tail. Relatively more non-target tail categories, each with fewer examples, will more diffusely contribute to training the target tail category, ensuring that the network does not overfit to either the relatively fewer training examples of the target tail category or the more represented contributing head categories.

In practice, any of these three effects may dominate, and rigorously calibrating them is left for future work devoted to that detailed analysis to compare to HSE.

Focused Ensemble Performance with One Label Smoothing Method: Since each CLS network at the 598x598 input size added to an ensemble improves performance more than adding another marginal network, this CLS benefit also reduces training time by focusing only on the CLS-trained models. For instance, in our ensemble ablation, we see that five CLS networks trained at the 598x598 input image size outperforms five CLS networks with the addition of any other network type that is not CLS 598x598. This clarity allows us to focus computational resources on only one type of network and not risk losing potentially beneficial diversity in our ensembles that might accrue from other models with complementary strengths had we trained them. This is a critical benefit to downstream work comparing different methods because it guides efficient allocation of limited compute resources on an already computationally intensive task.

Test Score Prediction Analysis: The scores from the test score prediction set (part of the validation set, which entrants see) are highly correlated with the test scores for the same model (network, or ensemble of networks, e.g) on the unseen test data provided per blinded submission by kaggle (Figure 7). In independent testing, we submitted a number of single category labels to kaggle to interrogate the iNaturalist 2018 Challenge test data and found in each case that the resulting test scores were very close to each other. This indicated that the mutually exclusive test set, while unseen and held out from training and validation data, was likely uniformly distributed over categories, as was the provided validation set. Based on this insight, we used only a portion of the validation set for validation fine-tuning (following [21]), leaving out a portion also uniformly distributed over categories to predict the test score. We found that the estimation error between scores on this test score prediction set and the actual test score were highly correlated.

We note the interrogation of the test set in this way does not confer significant benefit on the test score, as relatively tight bounds can be estimated [25], and that large numbers of submissions will typically not improve test scores. To wit, we did not tune, nor overfit to the test set here, except to establish that it was uniformly distributed over categories.

By predicting the test set score from a presumably identically distributed (over categories) test score prediction set, we estimate a conservative error bar on the test score—meaning that the actual error bar is likely smaller than our estimate. Specifically, the error bar fit estimate degrades with both the test score variability on the y-axis (the iNaturalist 2018 Challenge test score σ we seek to estimate) as well as the prediction test set score variability on the x-axis (which is a nuisance parameter). We cannot separate out these two sources of variability, but since the test set has many more examples in it, we anticipate its contribution to the estimation error, σ_{test}, is smaller than the contribution to the estimation error of the test score prediction set, $\sigma_{predict}$.

This error analysis helps in two ways. First, it provides a rough measure of the real performance improvement from method to method based on an empirically estimated confidence interval. Roughly, for CLS that translates to slightly larger than an approximately 5 σ improvement over the former state of the art reported on the iNaturalist 2018 Challenge [12]. Second, and more important to guide future work, such an estimation error together with the measured performance improvement per marginal ensemble network provides a rough means to estimate the expected performance improvement per additional trained network in an ensemble. This provides an ensembling stopping criterion to focus compute resources, which, along with the insight of Contribution 3, that CLS improves ensemble performance more per marginal network than other methods, is critical to efficiently allocating compute resources for methodological comparisons at scale (such as between CLS and HSE, e.g.) in downstream work.

Improving Tail Category Performance with Fine-Tuning: Prior work [21] inspired our adoption of fine-tuning on a more uniformly distributed set of categories. In our case, we used a fraction of the validation data for this purpose. We see similar gains in this work—i.e. CLS also benefits from this fine-tuning approach.

6. Conclusion

The long tails of FGVC tasks for natural image corpora present daunting training data collection requirements to achieve required accuracy objectives on tail categories with mainstream deep learning methods. Namely, the tail categories are many, sparse, and similar, making their per-category accuracies difficult to improve on with 1-hot labels that treat them independently in training. In this work we demonstrate that CLS' hierarchical prior on vector labels in the form of a phylogenetic tree can pool training data contributions from many of the tail classes, exploit their similarities, and thereby improve the accuracy on tail classes compared to 1-hot labels or other less judicious vector label smoothings.

CLS is Encoded by Domain Experts: The benefit of CLS alone is significant and does not require expertise in deep learning to realize—the phylogenetic tree prior came directly from a phylogenetic tree curated by biologists [15]. This is the only change from other methods [21] benchmarked on this same dataset that we show underperform without CLS compared to the same methods incorporating CLS.

CLS is Compatible with more Data-Driven Methods: While we present results only on CLS without a CLS-specific hyperparameter search, the CLS method proposed

is compatible with more empirical distillation and HSE methods which adjust label vectors based on training. Specifically, CLS can be incorporated directly into the trunk network of HSE, for instance. The CLS ensembles can be distilled into a single network to realize the benefits of distillation, including distillation benefits of adversarial example defense and compute reduction, e.g..

CLS's Prior Models can be Extended by Human or Machine: While we demonstrate a simple CLS approach that exploits an a priori provided phylogenetic tree, this *unlearned* prior can very likely be improved because the phylogenetic tree is not, by design, a guide to visual similarity, even within a species. For instance, even within species, there can be further training example pooling with visual similarity as encoded through latent activation clustering. Among butterflies, for instance, the within-species separation of chrysalis, caterpillar and butterfly stages may create separable clusters in an embedding of latent activations (as with t-SNE, e.g.). Within a bird species, the visual ornamentation of males vs. females may similarly cluster in an embedding of latent activations. Similarly, dog breeds may cluster. All of these finer levels may be similarly encoded into the CLS prior by either machine or human curator. As with all FGVC tasks, this presents additional challenges as training data fragments among the categories because categories with very little training data are split further, dividing the sparse training data among the finer subcategories. We show that CLS can still effectively pool training data in that scenario at the genus to species level of granularity and leave for future work the demonstration of even more fine-grained applications of CLS.

Future Work: Demonstrating and evaluating the combined benefits of both the *a priori* hierarchical CLS prior and the post hoc *learned* latent encodings of similarities (as in HSE and distillation, e.g.) together is left for future work, as is the significant challenge of comparing other methods that make use of the phylogenetic tree prior (like HSE) to CLS on the scale of the iNaturalist 2018 dataset. For perspective, even with no CLS hyperparameter tuning, the present study required >20,000 of GPU compute time. The GPU compute costs of rigorously comparing HSE to CLS with the hyperparameter searches required to reach conclusive results are anticipated to be even larger, and may warrant additional AutoML investigations, further increasing the computational costs.

7. Acknowledgements

This research was developed with funding from the Defense Advanced Research Projects Agency. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[1] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-Grained Visual Classification of Aircraft," Jun. 2013.

[2] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 554–561.

[3] A. B. Hillel and D. Weinshall, "Subordinate class recognition using relational object models," in *Advances in Neural Information Processing Systems*, 2007, pp. 73–80.

[4] T. L. Berg, A. C. Berg, and J. Shih, "Automatic attribute discovery and characterization from noisy web data," in *European Conference on Computer Vision*, 2010, pp. 663–676.

[5] G. Van Horn *et al.*, "The inaturalist species classification and detection dataset," 2018.

[6] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[7] S. Hou, Y. Feng, and Z. Wang, "Vegfru: A domain-specific dataset for fine-grained visual categorization," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017, pp. 541–549.

[8] H. Goeau, P. Bonnet, and A. Joly, "Plant identification based on noisy web data: the amazing performance of deep learning (LifeCLEF 2017)," in *CLEF 2017-Conference and Labs of the Evaluation Forum*, 2017, pp. 1–13.

[9] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur, "Dog breed classification using part localization," in *European Conference on Computer Vision*, 2012, pp. 172–185.

[10] "iNaturalist_Competition_FGVC_2018.pdf." [Online]. Available: https://www.dropbox.com/s/52nz6qc3zcwqhoa/iNaturalist_Competition_FGVC_2018.pdf?dl=0. [Accessed: 15-Nov-2018].

[11] *iNaturalist competition details. Contribute to visipedia/inat_comp development by creating an account on GitHub*. Visipedia, 2018.

[12] "iNaturalist Challenge at FGVC5." [Online]. Available: https://www.kaggle.com/c/inaturalist-2018. [Accessed: 01-Jul-2018].

[13] G. Van Horn and P. Perona, "The Devil is in the Tails: Fine-grained Classification in the Wild," *ArXiv Prepr. ArXiv170901450*, 2017.

[14] T. Chen, W. Wu, Y. Gao, L. Dong, X. Luo, and L. Lin, "Fine-grained representation learning and recognition by exploiting hierarchical semantic embedding," *ArXiv Prepr. ArXiv180804505*, 2018.

[15] "GBIF." [Online]. Available: https://www.gbif.org/. [Accessed: 15-Nov-2018].
[16] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning.," in *AAAI*, 2017, vol. 4, p. 12.
[17] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1106–1114.
[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
[19] G. Pereyra, G. Tucker, J. Chorowski, \Lukasz Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," *ArXiv Prepr. ArXiv170106548*, 2017.
[20] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *ArXiv Prepr. ArXiv150302531*, 2015.
[21] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie, "Large Scale Fine-Grained Categorization and Domain-Specific Transfer Learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4109–4118.
[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *ArXiv Prepr. ArXiv151203385*, 2015.
[23] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[24] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, 2014, pp. 3320–3328.
[25] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, "Generalization in deep learning," *ArXiv Prepr. ArXiv171005468*, 2017.