From generalized zero-shot learning to long-tail with class descriptors

Dvir Samuel[1] **Yuval Atzmon**[2] **Gal Chechik**[1,2]
[1]Bar-Ilan University, Ramat Gan, Israel
[2]NVIDIA Research, Tel Aviv, Israel
dvirsamuel@gmail.com, yatzmon@nvidia.com, gal.chechik@biu.ac.il

Abstract

Real-world data is predominantly unbalanced and long-tailed, but deep models struggle to recognize rare classes in the presence of frequent classes. Often, classes can be accompanied by side information like textual descriptions, but it is not fully clear how to use them for learning with unbalanced long-tail data. Such descriptions have been mostly used in (Generalized) Zero-shot learning (ZSL), suggesting that ZSL with class descriptions may also be useful for long-tail distributions.

We describe DRAGON*, a late-fusion architecture for long-tail learning with class descriptors. It learns to (1) correct the bias towards head classes on a sample-by-sample basis; and (2) fuse information from class-descriptions to improve the tail-class accuracy. We also introduce new benchmarks CUB-LT, SUN-LT, AWA-LT for long-tail learning with class-descriptions, building on existing learning-with-attributes datasets and a version of Imagenet-LT with class descriptors.* DRAGON *outperforms state-of-the-art models on the new benchmark. It is also a new SoTA on existing benchmarks for GFSL with class descriptors (GFSL-d) and standard (vision-only) long-tailed learning ImageNet-LT, CIFAR-10, 100, and Places365-LT.*

1. Introduction

Real-world data is predominantly unbalanced, typically following a long-tail distribution. From text data (Zipf's law), through acoustic noise (the 1-over-f rule) to the long-tail distribution of classes in object recognition [45], few classes are frequently observed, while the many remaining ones are rarely encountered.

Long-tail data poses two major challenges to learning: *data paucity* and *data imbalance*. First, at the tail of the distribution, classes are poorly sampled and one has to use few-shot and zero-shot learning techniques. Second, when training a single model for both richly-sampled classes and poorly-sampled classes, the common classes dominate training, and as we show below, this skews prediction confidence towards rich-sampled classes.

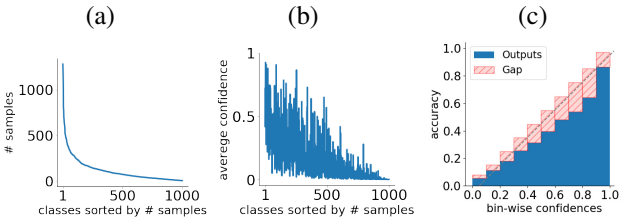

Figure 1: Training with unbalanced data leads to a "familiarity bias", where models are more confident and more over-confident about frequent classes [48, 47, 6]. **(a)** Class distribution of a long-tailed ImageNet [12]. Classes are ordered from left to right by decreasing number of samples. **(b)** When training a ResNet-10 on ImageNet-LT, validation (and test) predictions tend to have low confidence for tail classes. We show the mean output of softmax for each class, conditioned on samples from that class. **(c)** A reliability graph for the model in b. Predictions are grouped based on confidence. The model has larger confidence gaps (pink boxes) for more confident predictions, which usually come from head classes. This result suggests that overconfidence is strongly affected by class frequency, and we can learn to correct it if the number of samples is known.

To address *data paucity* of tail classes, note that visual examples can very often be augmented with class descriptors. Namely, semantic information about classes given as text or attributes [26, 36, 4, 52]. This approach, *learning with class-descriptors* has been studied mostly for zero-shot and generalized zero-shot learning [38, 32, 43, 52, 53]. Here, we propose to adapt it to generalized few-shot learning (GFSL) by fusing information from two modalities. A visual classifier, expected to classify correctly head classes, and a semantic classifier, trained with per-class descriptors and is expected to classify correctly tail classes. We explain the subtleties of GZSL and GFSL in Section 2 (Related work).

To address *data imbalance*, we first note that learning with unbalanced data leads to a **familiarity effect**, where models become biased to favor the more familiar, rich-

sampled classes [48, 6]. Since deep models tend to be overly confident about high-confidence sample predictions [17, 25], they become over-confident about head classes [48, 47, 6]. Figure 1 illustrates this phenomenon. It shows that a model trained on unbalanced data (Figure 1a), has higher confidence for head classes (Figure 1b). It is also an over-estimate of the true accuracy (Figure 1c), especially for head classes. See further analysis in Appendix B. [1]

A natural way to correct the familiarity bias would be to penalize high-frequency classes, either during training using a balanced loss, or post-training [23]. However, it would be a grave mistake to penalize all samples from rich classes, because confidence is sometimes justified, as in the case of "easy" prototypical examples of a class. Indeed, we show below that addressing the familiarity bias benefits from *per-sample debiasing*, going beyond class-based debiasing. To summarize, **model overconfidence is affected by class frequency. It can be estimated by observing the full vector of predictions to correct for overconfidence. Not all samples of a class should be penalized for belonging to a frequent class**.

Importantly, the familiarity effect caused by data imbalance has a crippling effect on model accuracy and on aggregating predictions from multiple modalities. Several approaches attempted calibrating predictions of deep networks to remedy the above biases (see e.g. a survey in [17]) and some became common practice. Unfortunately, the problem is still far from being solved.

We propose to address both the *data-imbalance* and the *data paucity* learning challenges, using a single late-fusion architecture. We describe an easy-to-implement debiasing module that offsets the familiarity effect by learning to predict the magnitude of the bias for any given sample. It further improves learning at the tail by learning to fuse information from visual and semantic modalities. It can easily be reduced to address long-tail learning with a single modality (vision only), where it improves over current baselines.

The paper has three main novel contributions:
(1) A new late-fusion architecture (DRAGON) that learns to fuse predictions based on vision with predictions based class descriptors.
(2) A module that rebalances class predictions across classes on a sample-by-sample basis.
(3) New benchmarks CUB-LT, SUN-LT, and AWA-LT for evaluating long-tail learning with textual class descriptors (LT-d). DRAGON is SoTA on these datasets, and also on ImageNet-LT augmented with class descriptors and on existing two-level benchmarks.

[1] As an interesting side note, studies of human decision making and preference learning show a similar bias towards familiar samples. This effect is widely observed and has been connected to the availability heuristic studied by Tversky and Kahneman [44].

2. Related methods

Long-tail learning: Learning with unbalanced data causes models to favor head classes [6]. Previous efforts to address this effect can be viewed as either algorithmic or data-manipulations approaches.

Algorithmic approaches encourage learning of tail classes using a non-uniform cost per misclassification function. A natural approach is to rescale the loss based on class frequency [19]. [27] proposed to down-weigh the loss of well-classified examples, preventing easy negatives from dominating the loss. [37] dynamically rescaled the cross-entropy loss based on the difficulty to classify a sample. [7] proposed a loss that encourages larger margins for rare classes. [23] decoupled the learning procedure into representation learning and classification and studied four approaches. Among them, LWS L_2-normalizes the last-layer, since the weight magnitude correlates with class cardinality. The effect of this approach is similar to that presented in this paper, but here we apply recalibration dynamically on a sample-by-sample basis.

Data-manipulation approaches aim to flatten long-tail datasets to correct the bias towards majority classes. Popular techniques employ over-sampling of minority classes (more likely to overfit) [10, 18] , under-sampling the majority classes (wastes samples) [13] , or generating samples from the minority classes (can be costly to develop)[5].

Another approach is to transfer meta-level-knowledge from data-rich classes to data-poor classes. [49] gradually transfer hyperparameters from rich classes to poor classes by representing knowledge as trajectories in model space that capture the evolution of parameters with increasing training samples. [29] first learns representations on the unbalanced data and then fine-tunes them using a class-balanced sampling and a memory module.

Learning with class descriptors: Learning with class descriptors is usually applied to zero-shot learning (ZSL) [52, 26, 4], where a classifier is trained to recognize (new) unseen classes based on their semantic description, which can include a natural-language textual description or predefined attributes. In several ZSL studies, attributes detected in a test image are matched with ground-truth attributes of each class, and several studies focused on this matching [26, 4, 42, 56, 55, 8].

A series of papers proposed to learn a shared representation of visual and text features (class-descriptors). As one example, [43] learns such a shared latent manifold using autoencoders and then minimizes the MMD loss between the two domains. Another recent line of work synthesizes feature vectors of unseen classes using generative models like VAE of GAN, and then use them in training a conventional classifier [32, 51, 14, 1, 31, 59, 38, 53]. The major baseline we compare our approach with is CADA-VAE [38], the current SoTA for Generalized FSL with class descrip-

tors. CADA-VAE uses a variational autoencoder that aligns the distributions of image features and semantic (attribute) class embedding in a shared latent space. A recent work, [53], uses a mixture of VAEs and GANs. We could not directly compare with [53] because their FSL protocol deviates from the standard benchmark of [38, 52] by fine-tuning the CNN features. *Without* fine-tuning, their reported metrics for GZSL are similar to CADA-VAE.

Some studies fused information from vision and *per sample* descriptors (e.g., [58]). This is outside the scope of this paper because it may require extensive labeling.

Generalized ZSL (GZSL) and Generalized FSL: GZSL extends ZSL to the scenario where the test data contains both seen and unseen classes [9, 52, 40]. Recently, GZSL extended to Generalized Few-Shot-Learning with class descriptors (GFSL-d), where the unseen classes are augmented with a fixed number of few training samples [38, 43]. Namely, the distribution of samples across classes is a 2-level distribution, with *many* "head" classes and a smaller set of "tail" classes all having the same (small) number of samples per class. Both GZSL and GFSL-d can be viewed as special cases of long-tail learning with class-descriptors, but with a *short*-tailed unnatural distribution.

Most related GZSL approaches are [3, 40, 54]. They use a gating mechanism to weigh the decisions of seen-classes experts and a ZSL expert. The gating module is modeled as an out-of-distribution estimator. The current paper differs from their work by (1) The problem setup is different. Here, all samples are *in-distribution* and the distribution of classes is smooth and long-tail with a much smaller number of head classes. (2) DRAGON architecture first quantifies and corrects the (smooth) familiarity effect. Then it learns how to fuse the debiased decision of the two experts.

Early vs late fusion: When learning from multiple modalities, one often distinguishes between early and late fusion models [28]. Early fusion models combine features from multiple modalities to form a joint representation. Late fusion methods combine decisions of per-modality models [22, 2, 35]. Our approach addresses the long-tail setup, by leveraging the information in the familiarity bias to debias experts predictions.

3. Long-tail learning with class descriptors

We start with a formal definition of the problem of learning over unbalanced distributions with class descriptors.

We are given a training set of n labeled (image) samples: $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where each $\mathbf{x}_i$ is a feature vector and y_i is a label in $\{1, 2, \ldots k\}$. Samples are drawn from a distribution $\mathcal{D} = p(x, y)$ such that the marginal distribution over the classes $p(y)$ is strongly non uniform. For example, $p(y)$ may be exponential $p(y) \sim \exp(-ky)$.

As a second supervision signal, each class y is also accompanied with a *class-description* vector $\mathbf{a}_j$, $j = 1, .., k$,

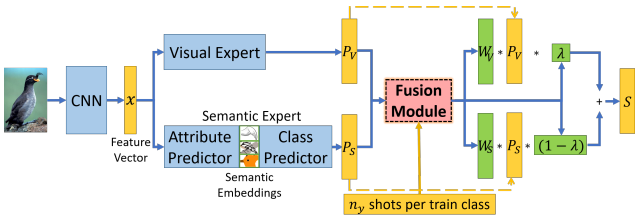

Figure 2: The DRAGON architecture for long-tail learning with class-descriptors. The visual-expert and semantic-expert each outputs a prediction vector fed to a fusion module. The fusion module combines expert predictions and debias them. Blue, network components. Yellow, input to the fusion module. Green, the outputs of the fusion module.

in the form of semantic attributes [26] or natural-language embedding [36, 59, 40]. For example, classes in CUB [46] are annotated with attributes like `Head-color:red`.

At test time, a new set of m test samples $\{\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m}\}$ is given from the same distribution $\mathcal{D}$. We wish to predict their correct classes.

4. Our approach

Our approach is based on two observations: (1) Semantic descriptions of classes are easy to collect and can be very useful for tail (low-shot) classes, because they allow models to recognize classes even with few or no training samples [26, 52, 4, 56]. [56] (Figure 4) shows visualization of the class descriptors: Classes form compact clusters and are near their corresponding class descriptors. This allows a model to make correct classifications, based on class descriptors, even with few samples or no samples at all. For more details see Appendix C. (2) The average prediction confidence over samples of a class is correlated with the number of training samples of that class (Figure 1).

Our architecture leverages these observations and learns to (1) Combine predictions of two expert classifiers: A conventional visual expert which is more accurate at head classes and a semantic expert which excels at tail classes; (2) Reweigh the scores of each class prediction, taking into account the number of training samples for that class.

4.1. The architecture

The DRAGON architecture follows two design considerations: *modularity* and *low-complexity*. First, modularity; DRAGON allows to plug-in existing multi-modal experts, each trained for its own modality. Below we show experiments with language-based experts and a visual expert, but other modalities can be considered (e.g., mesh, depth, motion, or multi-spectral information). Second, limiting the model to have a small number of parameters is important because tail classes only have few training samples and the model must perform well at the tail.

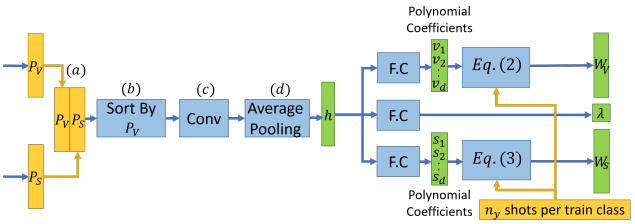

Figure 3: Architecture of the fusion-module for long-tail learning with class-descriptors. In blue, network components. In yellow, inputs to the fusion-module and in green, activations or outputs of the fusion-module. The inputs P_V denote the softmax prediction vector of the *Visual Expert*, and P_S that of the *Semantic Expert*. The outputs W_V, W_S and λ are used in Eq. (1) for re-weighting the inputs. See Section 4.2 for more details.

Our general architecture (Figure 2) takes a late fusion approach. It consists of two experts modules: A visual expert and a semantic expert. Each expert outputs a prediction vector which is fed to a fusion module. The fusion module combines the expert predictions and learns to debias the familiarity effect, by weighing the experts and re-scaling their class predictions.

4.2. A fusion-module

The fusion module takes as input the prediction vectors of two experts p_V, p_S for a given image, and a vector containing the number of training samples per class. It has three outputs: $\lambda \in (0,1)$ is a scalar to trade-off the visual expert against the semantic expert. $\mathbf{w}_V \in (0,1)^k$ is a vector that weighs the predictions of the visual expert. Similarly, $\mathbf{w}_S$ weighs the semantic expert predictions. Given these three outputs, a debiased score is computed for a class y:

$$S(y) \quad = \quad \lambda \mathbf{w}_V(y) p_V(y) + (1-\lambda)\mathbf{w}_S(y) p_S(y). \quad (1)$$

Figure 3 describes the architecture of the fusion-module. It has two main parts. The first part maps the prediction scores to a meaningful joint space, by first aligning the prediction of both classifiers, and then sorting according to confidence.

In more detail, the first part has four steps. (a) Stacking together the predictions of two experts to a $\mathcal{Y} \times 2$ vector. This makes the following convolution meaningful across the 2 experts axis. (b) To make convolution meaningful also along the classes axis, and since classes are categorical, we reorder classes by their prediction score according to one of the experts. Section 4.3 explains the rationale of this reordering . (c) Now that predictions are sorted, feed the sorted scores to a $N_{filters} \times 2 \times 2$ convolutional network. (d) Follow with an average-pooling layer (per class), yielding a $(\mathcal{Y}-1)$-dimensional vector $\mathbf{h}$.

The goal of the second part is simply to predict a debiasing coefficient for each prediction, namely, learn a function from n_y to $(0,1)^k$. We know from Figure 1 that the bias is inversely related to the number of samples. Aiming for a simple model, we train a polynomial regression that takes as input the number of samples and outputs a debiasing weight $(w_V(y))$. The coefficients of this polynomial $v_0, ..., v_{d-1}$ are learned as a deep function over $\mathbf{h}$. Similarly for $f_S(y)$.

More formally, let n_y be the number of training samples of class y, and let $\bar{n}_y = n_y / \max_y n_y$ be the normalized counts, then we have

$$\mathbf{w}_V(y) = \sigma\Big[\sum_{j=0}^{d-1} v_j(\mathbf{h})\bar{n}_y^j\Big], \ \mathbf{w}_S(y) = \sigma\Big[\sum_{j=0}^{d-1} s_j(\mathbf{h})\bar{n}_y^j\Big], \quad (2)$$

where d is the polynomial degree and σ denotes a sigmoid that ensures that the resulting scale is in $[0,1]$.

Finally, the fusion module also predicts the trade-off scalar λ to control the relative weight of the visual and semantic experts. This is achieved using a fully connected layer over $\mathbf{h}$. Section 9 analyzes the contribution of each component of the approach with an ablation study.

4.3. Architecture design decision

DRAGON is designed with a small number of model parameters so it can improve predictions at the tail, where very few samples are available.

Debiasing based on all expert predictions. To achieve the above goals, DRAGON implicitly learns how frequent is a class of a given sample. Namely, when the model receives a new test sample, it predicts if it is from a head class, a tail class, or somewhere between, and adjusts the confidence of the experts accordingly. To do this, it has been shown in the context of zero-shot learning that profile of confidence values are a good predictor if a sample comes from a seen or unseen class [3]. DRAGON generalizes this idea to long-tail distributions. To do this it takes as input all class predictions from both experts (Figure 3a).

Using order statistics over predictions. To process expert prediction, we point out that order statistics over the prediction vector – the maximum confidence, 2^{nd} max, etc...– provides a strong signal about confidence calibration. Using the maximum of a vector is a very common operator in deep learning, known as max pooling. Here however, there is additional important information the gap between subsequent order statistics, like $\max - 2^{nd}\max$. As an intuitive example, a maximal prediction of 0.6 should be interpreted differently if the 2^{nd} max is 0.4 or 0.1.

Order statistics can be easily computed by sorting the vector of predictions (Figure 3b). Sorting also increases the sample efficiency for learning, because later layers have each order statistic located at a fixed position in their input regardless of class. The function learned over order-statistics gaps is therefore shared across all classes.

The 2×2 convolution (Figure 3c) works well with the

sorted expert predictions. Its filters capture two signals: (1) the confidence gaps between the two experts for each class; and (2) the confidence gaps between order-statistics for each expert alone.

5. Experiments

We evaluate DRAGON in three unbalanced benchmark scenarios. (1) *"Smooth-Tail"*, the long-tailed distribution of classes decays smoothly (Figure 4) and each class is accompanied with textual class descriptors. (2) *"Two-Level"*, the distribution has a step-shape as in [38]; Most classes have many samples and the rest have few samples (Figure 5). (3) *"Vision-only"*, a long-tail setup, as in Smooth-Tail, but without class descriptors.

We compare DRAGON with SoTA approaches on standard benchmarks for each of these three scenarios. See Appendix E for implementation details.

5.1. Overview of main results

For Smooth-Tail distributed data, we evaluated DRAGON on four benchmarks that we created from existing datasets. We added textual descriptors to ImageNet-LT [29], and generated long-tail versions of CUB, SUN and AWA. Dragon outperforms all baselines on various metrics.

For Two-Level distributed data, DRAGON surpasses the current SoTA [38], tested using their experimental setup.

For Vision-only long-tail data, we tested the calibration component of DRAGON (without fusion). It achieves a new SoTA on ImageNet-LT [29], Places365-LT [29], Unbalanced CIFAR-10/100 [7] and comparable results on iNaturalist2018 [20]. Details and results in Appendix A .

6. Smooth-Tail distribution

6.1. Datasets

To evaluate long-tail learning with class descriptors, we created benchmark datasets in two ways. First, we created long-tail versions of existing learning-with-class-descriptors benchmarks. Second, we augmented existing long-tail benchmark (ImageNet-LT) with class descriptors.

Specifically, we created new long-tail variants of the 3 main learning-with-class-descriptors benchmarks: CUB [46], SUN [33] and AWA [26], illustrated in Figure 4. We ranked classes by the number of samples in each class after assigning tail classes to be consistent with those in the Two-Level benchmark [51, 38] (See Appendix E.2 for more details). We then computed a frequency level for each class following an exponentially decaying function of the form $f(class) = ab^{-rank(class)}$. a and b were selected such that the first class has the maximum number of samples, and the last class has 2 or 3 samples depending on the dataset. We then drew a random subset of samples from each class based

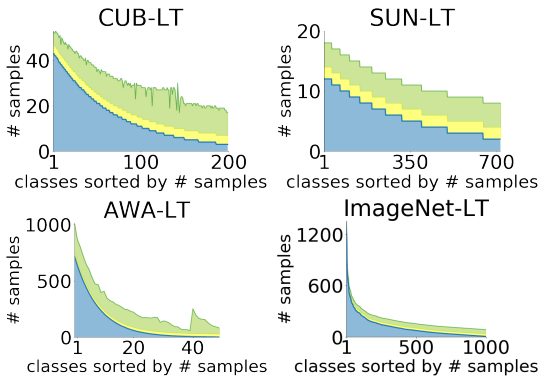

Figure 4: Long-tailed versions of CUB, SUN, AWA and ImageNet. Number of samples for the training, validation and test sets are shown respectively by blue, yellow and green.

	CUB-LT	SUN-LT	AWA-LT
TRAIN # SAMPLES	2,945	4,084	6,713
VAL # SAMPLES	600	1,434	1,250
TEST # SAMPLES	2,348	2,868	6,092
TRAIN SET PROPERTIES			
MAX # SAMPLES	43	12	720
MIN # SAMPLES	3	2	2
MEAN # SAMPLES	14.725	5.696	123.460
MEDIAN # SAMPLES	11	5	35

Table 1: Properties of CUB-LT, SUN-LT and AWA-LT.

on their assigned frequency $f(class)$. To create the validation set, we randomly drew a constant number of samples per class, while keeping an overall size of 20% of the training set. See dataset statistics in Table 1.

As a second type of benchmark, we used the existing long-tailed ImageNet [29] and augmented it with class descriptors. Specifically, we used the word2vec embeddings provided by [8], which are widely used in the literature . Their word embeddings were created by training a skip-gram language model on a Wikipedia corpus to extract a 500-dimensional word vector for each class.

Together, this process yielded the following datasets:

1. **CUB-LT**, based on [46], consists of 2,945 training visual images of 200 bird species. Each species is described by 312 attributes (like tail-pattern:solid, wing-color:black). Classes have between 43 and 3 images per class.
2. **SUN-LT**, based on [33], consists of 4,084 training images, from 717 visual scene types and 102 attributes (like material:rock, function:eating, surface:glossy). Classes have between 12 and 2 images per class.
3. **AWA-LT**, based on [26], consists of 6,713 training images of 50 animal classes and 85 attributes (like texture:furry, or color:black). Classes have between 720 and 2 images per class.
4. **ImageNet-LT-d**, based on [29], consists of 115.8K images from 1000 categories with 1280 to 5 images per class.

METHOD	CUB-LT		SUN-LT		AWA-LT	
	Acc_{PC}	Acc_{LT}	Acc_{PC}	Acc_{LT}	Acc_{PC}	Acc_{LT}
VISION-ONLY						
CE LOSS* **(VE)**	53.0	65.5	33.7	40.0	73.7	93.4
FOCAL LOSS [27]*	46.4	61.3	30.1	37.4	73.5	91.8
ANCHOR LOSS [37]*	48.3	64.7	28.2	36.2	69.1	93.2
RANGE LOSS [57]*	48.5	65.3	27.9	36.0	68.9	93.5
LDAM LOSS [7]*	50.1	64.1	29.8	36.4	69.1	93.5
CB LWS [23]*	53.1	65.7	33.9	40.2	73.4	93.6
MULTI-MODAL						
LAGO [4]* **(SE)**	54.8	66.0	18.2	18.3	74.0	93.0
CADA-VAE [38]*	48.3	57.4	32.8	35.1	73.5	89.5
LATE FUSION						
MIXTURE	54.8	66.0	34.0	40.3	74.0	93.7
DRAGON (OURS)	**57.8**	**67.7**	**34.8**	**40.4**	**74.1**	**94.1**
DRAGON + BAL. (OURS)	**60.1**	**66.5**	**36.1**	38.5	**76.2**	92.2

IMAGENET-LT-D	RESNET-10	RESNEXT-50			
METHOD	Acc_{PC}	Acc_{MS}	Acc_{MED}	Acc_{FS}	Acc_{PC}
VISION-ONLY					
CE LOSS* **(VE)**	34.8	65.9	37.5	7.7	44.4
FSLWF [16]	28.4	-	-	-	-
FOCAL LOSS [27]	30.5	-	-	-	-
RANGE LOSS [57]	30.7	-	-	-	-
OLTR [29]	35.6	-	-	-	37.7
CB LWS [23]	41.4	60.2	47.2	30.3	49.9
MULTI-MODAL					
DEM [56]* **(SE)**	18.1	16.5	13.0	**50.8**	19.5
CADA-VAE [38]*	42.1	57.4	43.7	27.0	49.3
LATE FUSION					
MIXTURE	40.7	63.8	36.3	23.9	45.1
SHARMA ET AL. [39]	39.2	-	-	-	-
DRAGON (OURS)	**43.1**	**66.0**	**38.3**	47.6	**51.2**
DRAGON + BAL. (OURS)	**46.5**	62.0	**47.4**	50.2	**53.5**

Table 2: **Smooth-tail distribution:** Rows with * denote results reproduced by us. The rest were taken from [23, 29]. VE and SE refer to the *visual-expert* and *semantic-expert* that were used to train DRAGON. Bal. refers to training DRAGON with a balanced loss. **Left:** Comparing DRAGON with baselines on the long-tailed versions of datasets with attributes. We report Per-Class Accuracy Acc_{PC} and Long-Tail Accuracy Acc_{LT}. **Right:** Comparing DRAGON with baselines on the long-tailed ImageNet with word embeddings.

We use Word2Vec [30] class embedding features provided by [8], as textual descriptors.

6.2. Training scheme

The familiarity effect is substantial in the validation and test data, but not in the training data, where models may actually become more confident on rare classes. We observed this effect in CUB-LT, SUN-LT, and AWA-LT. Since we wish to train the fusion module using data that exhibits the familiarity bias, we hold-out a subset of the training data and use it to simulate the response of experts to test samples. Note that in large-scale datasets, like ImageNet-LT-d, no hold-out set is needed and DRAGON is trained on the training set. There, the familiarity bias is also present in the training data, as the models did not overfit the tail classes. Appendix D illustrates this effect in more detail.

6.3. Baselines and variants

We compared DRAGON with long-tail learning and unbalanced data approaches: **Focal Loss [27]**, **Anchor Loss [37]**, **Range Loss [57]**, and **LDAM Loss [7]** are loss manipulation approaches for long-tail distributions. **FSLwF [16]**, **OLTR [29]** and **Classifier-Balancing (CB) [23]** are algorithmic approaches in the long-tail learning benchmarks. **Mixture** and **Class Balanced Experts [39]** are late fusion approaches. **Mixture** resembles mixture-of-experts (MoE) [21] without EM optimization. It fuses the raw outputs of the two experts by a gating module. As with standard MoE models, the gating module is trained with visual-features as inputs.

For CUB-LT, SUN-LT, and AWA-LT, the visual expert was a linear layer over a pre-trained ResNet from [52, 50], which we trained with a balanced xent loss (CE Loss). The semantic expert was LAGO [4]. For ImageNet-LT-d, we followed [23] and set the visual expert to be ResNet-10 or ResNeXt-50. The semantic expert was DEM [56].

6.4. Evaluation metrics and protocol

Evaluation Protocol: The experiments for CUB-LT, SUN-LT, and AWA-LT follow the standard protocols set by [38, 52, 50], including their ResNet-101 features. Their split ensures that **none of the test classes appear in the training data used to train the ResNet-101 model**. For ImageNet-LT-d we used the protocols in [29, 23] with the pre-trained ResNet-10 and ResNeXt-50 provided by [23]. We extracted their features to train the semantic expert.

Evaluation metrics: We evaluated DRAGON on the Smooth-Tail benchmark with the following metrics:
(a) Per-Class Accuracy (Acc_{PC}): Balanced accuracy metric that uniformly averages the accuracy of each class $\frac{1}{k}\sum_{y=1}^{k} Acc(y)$, where $Acc(y)$ is the accuracy of class y.
(b) Long-Tailed Accuracy (Acc_{LT}): Test accuracy, where the distribution over test classes is long-tailed like the training distribution. This is expected to be the typical case in real-world scenarios. See Appendix E.1 for more details.
(c) Many-Shot, **Medium-shot** and **Few-Shot accuracies**: For ImageNet-LT-d, we follow [29] and report accuracy for: Acc_{MS} (>100 training images), Acc_{MED} (20-100 images) and Acc_{FS} (< 20 images).

6.5. Results with smooth-tail distribution

Table 2 (Left) provides the test accuracy for three long-tail benchmark datasets and compares DRAGON to baselines and individual components of the DRAGON model. DRAGON achieves higher accuracy compared with all competing methods, both with respect to class-balanced accu-

CUB-LT	Acc_{PC}	Acc_{LT}
MAX.	57.3	70.8
AVG.	57.0	70.0
PRODUCT	56.9	70.3
MIXTURE	53.7	66.5
DRAGON (OURS)	**60.0**	**70.9**

Table 3: Comparing DRAGON against common late fusion approaches on the validation set of CUB-LT.

racy (Acc_{PC}) and to test-distribution accuracy (Acc_{LT}). Improving Acc_{LT} indicates that DRAGON effectively classifies head classes, which are heavily weighted in Acc_{LT}. At the same time, improving Acc_{PC} indicates that DRAGON also effectively classifies tail classes, which are up-weighted in Acc_{PC}.

Table 2 (Right) provides the Acc_{PC} accuracy for ImageNet-LT-d. We can directly see the benefit of fusion information between modalities - the visual expert excels on many-shot classes, Acc_{MS}, while the semantic expert excels only on few-show classes, Acc_{FS}. DRAGON recalibrate and fuse both experts to excel in all classes.

We further trained DRAGON with a balanced cross-entropy loss (Bal.). This strategy has a synergistic effect with DRAGON (last row in Table 2): It improves tail accuracy Acc_{PC} for all benchmarks while only marginally hurting head accuracy $AccLT$.

Table 3 compares DRAGON against common late fusion strategies, on the validation set of CUB-LT: *AVG* (averaging expert predictions), *Max* (taking the largest prediction), *Product* (multiplying expert predictions) and *Mixture*. We show that those approaches, which late fuse predictions of the two experts, are usually better at head classes (Acc_{LT}) while giving less accurate results for tail classes (Acc_{PC}). DRAGON achieves better results on both metrics because it also calibrates expert predictions. In the ablation study (Table 5) we compare our fusion module to more ablated fusion components.

7. Two-Level (GFSL-d) benchmark

We follow the protocol of [38] on the original CUB, SUN, and AWA (Figure 5), to compare DRAGON in a Two-Level setting.

For those datasets, many-shot classes are kept as in the original train-set, while few-shot classes have an increasing number of shots: 1,2,5,10 and 20 (in SUN up to 10 shots).

7.1. Baselines and variants

We compared DRAGON with **LDAM Loss [7]** and with SoTA multi-modal GFSL-d approaches: **ReViSE [43]**, **CA-VAE [38]**, **DA-VAE [38]** and **CADA-VAE [38]**. Their results were obtained from the authors of [38], while LDAM results were reproduced by us.

The visual expert was a linear layer over a pre-trained ResNet from [52, 50], which we trained with a balanced cross-entropy loss. The semantic expert was LAGO [4].

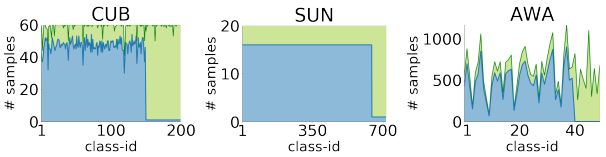

Figure 5: Two-level variants of CUB, SUN and AWA as in [38]. Blue: training set, green: test set.

TWO-LEVEL	CUB				SUN			AWA			
# SHOTS	**1**	**5**	**10**	**20**	**1**	**5**	**10**	**1**	**5**	**10**	**20**
LDAM [7]*	2.4	36.0	52.2	61.5	4.3	26.6	37.0	12.4	41.1	57.0	68.6
REVISE [43]	36.3	44.6	50.9	-	27.4	37.4	40.8	56.1	64.1	67.8	-
CA-VAE [38]	50.6	59.6	62.2	-	37.8	44.2	45.8	64.0	76.6	79.0	-
DA-VAE [38]	49.2	58.8	60.8	-	37.8	43.6	45.1	68.0	75.6	76.8	-
CADA-VAE [38]	55.2	63.0	64.9	66.0	40.6	46.0	47.6	**69.6**	**78.1**	80.2	80.9
CE LOSS* **(VE)**	1.2	30.2	50.2	60.9	1.8	25.1	38.3	11.0	47.8	69.9	73.9
LAGO* **(SE)**	23.0	49.0	58.6	64.8	19.5	25.6	27.8	20.2	59.0	68.7	75.8
DRAGON (OURS)	**55.3**	**63.5**	**67.8**	**69.9**	**41.0**	**46.7**	**48.2**	67.1	76.7	**81.9**	**83.3**

Table 4: **Two-Level distributions:** Comparing DRAGON on Two-Level CUB, SUN and AWA with SoTA GFSL models and baselines and with increasing number of few-shot training samples. Values denote the Harmonic mean Acc_H. VE and SE refer to the *visual-expert* and *semantic-expert* that were used to train DRAGON.

7.2. Evaluation metrics

Following [38], we evaluated the Two-Level benchmark with the Harmonic mean metric (Acc_H): It quantifies the overall performance of head and tail classes, by $Acc_H = 2(Acc_{ms}Acc_{fs})/(Acc_{ms} + Acc_{fs})$. Where, Acc_{ms} is the per-class accuracy over many-shot classes and Acc_{fs} is the per-class accuracy over few-shot classes.

7.3. Results with two-level distribution

Table 4 compares DRAGON with SoTA baselines on the Two-Level setup. Our model wins in CUB and SUN on all shots but loses on AWA for fewer than 10 samples. Furthermore, DRAGON gains better results when the number of shots increases in contrast to complex generative models like CADA-VAE [38]. Appendix F.1 provides results for Acc_{fs} and Acc_{ms} with 1,2,5,10,20 shots.

8. Vision-only long-tail learning

The approach presented in this paper focuses on learning from two modalities, vision and language. Learning to remove the bias predictions can also be useful when learning with a single modality, as in standard long-tail learning [23, 7, 27, 29]. Due to a lack of space, results are provided separately in Appendix A.

	Acc_{PC}	Acc_{LT}	$\#params$
F.C.	54.0	67.1	403
F.C. & $1/n_y$ RESCALE	56.7	60.3	403
F.C. & NON-PARAMETRIC RESCALE	58.2	68.0	81,406
CONV. & NON-PARAMETRIC RESCALE	58.7	68.2	40,612
CONV. & SINGLE PARAMETRIC RESCALE	59.0	67.5	613
DRAGON (OURS)	**60.0**	**70.9**	**1,015**

Table 5: Ablation study, comparing different fusion and re-scaling approaches. The results show the contribution of the convolutional backbone and the re-scaling method for the two experts (validation set, CUB-LT).

9. Ablation study

To understand the contribution of individual components of DRAGON, we carried ablation experiments. We report results on the validation set, which were consistent with the test set (Appendix F.2).

Fusion-Module Architecture: Table 5 compares the performance of various components of the fusion-module on CUB-LT. (1) *F.C.:* predicts λ using a fully-connected layer over $\mathbf{p}_V, \mathbf{p}_S$, no re-balancing ($\mathbf{w}_V(y) = \mathbf{w}_S(y) = 1, \forall y$). (2) *F.C. & $1/n_y$ rescale:* learns λ as in *F.C.*, rescales experts predictions by n_y. (3) *F.C. & non-parametric rescale:* learns λ as *F.C.* and rescales both experts predictions by a learned non-parametric weight for each class instead of a polynomial. (4) *Conv. & non-parametric rescale:* like (3), then applies sorting and convolution (Section 4.2). (5) *Conv. & single parametric rescale* replaces the non-parametric re-scaling weights by a single polynomial of parametrized weights. (6) DRAGON is our full approach described in Section 4. The comparison shows that rescaling expert predictions significantly improves Acc_{PC} and that reducing the number of parameters using the convolutional layer is important.

To quantify the contribution of *per-sample* weighting, Table 6 compares it against *per-class* weighting on three long-tail benchmarks: ImageNet-LT, Places365-LT and CIFAR100-LT. To keep the comparison fair, this was done using vision-only ($\lambda = 1$), and sweeping over the same set of hyper parameters. To gain more intuition on how per-sample weighting helps, Fig. 6 plots the per-sample weights of four images from a ImageNet-LT head class (mousetrap). Per-sample weighs more strongly "easy" samples (low entropy) than non-typical samples. This illustrates that per-sample weighting does not penalize "justified" high-confidence predictions if they happen to arrive from a head class. At the same time, per-sample weighting gives more chance to tail classes, reducing the familiarity bias .

Sharing order statistics (sorting): Table 7 quantifies the benefit of sorting expert predictions. As discussed in Section 4.3, sorting enables sharing of information across classes by fixing the input location of each order statistic.

	Places365-LT	**ImageNet-LT**		**CIFAR100-LT**
	ResNet-50	ResNet-10	ResNeXt-50	ResNet-32
CE Loss **(VE)**	30.2	34.8	44.4	38.3
Per-class	36.9	40.0	49.2	40.5
Per-sample	**38.1**	**42.0**	**50.1**	**42.0**

Table 6: Ablation of per-sample weighting on vision-only benchmarks. VE refers to *visual-expert*.

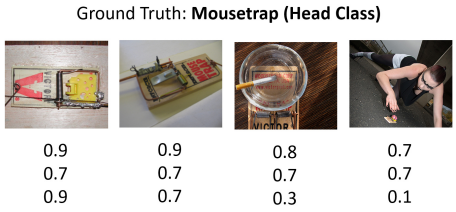

Figure 6: Using per-sample weighting, images typical for the class Mousetrap (softmax has low-entropy) are weighed more strongly than non-typical images (softmax has high-entropy). Per-class weighting reweighs all samples for that class the same (0.7), hurting recognition of typical images.

SORTING	Acc_{PC}	Acc_{LT}
NO SORTING	58.7	68.2
SORTING BY VISUAL EXPERT	**60.0**	**70.9**
SORTING BY SEMANTIC EXPERT	60.0	70.8

Table 7: Ablation study, quantifying the contribution of sorting the fusion-module inputs (validation set, CUB-LT).

10. Conclusion

We discuss two key challenges for learning with long-tail unbalanced data: A "familiarity bias", where models favor head classes, and low accuracy over tail classes due to lack of samples. We address these challenges with DRAGON, a late-fusion architecture for visual recognition that learns with per-class semantic information.

This is achieved by performing per-sample debiasing that is based on the full vectors of predictions from a visual module and a semantic module. It outperforms existing methods on new long-tailed versions of ImageNet, CUB, SUN, and AWA. It further sets new SoTA on a Two-Level benchmark [38].

DRAGON was designed as a late-fusion modular architecture, where one can easily plug-in new pretrained experts as they are developed. As a potential drawback, earlier sharing of information between visual and semantic experts may improve performance even further.

Strongly unbalanced data with a long-tail is ubiquitous in numerous domains and problems. The results in this paper show that a light-weight late-fusion model can be used to address many of the challenges posed by class imbalance.

References

[1] G. Arora, V-K. Verma, A. Mishra, and P. Rai. Generalized zero-shot learning via synthesized examples. In *CVPR*, 2018.

[2] A. Asvadi, L. Garrote, C. Premebida, P. Peixoto, and U. Nunes. Multimodal vehicle detection: fusing 3d-lidar and color camera data. *Pattern Recognition Letters*, 2018.

[3] Y. Atzmon and G. Chechik. Adaptive confidence smoothing for generalized zero-shot learning. *CVPR*, 2018.

[4] Y. Atzmon and G. Chechik. Probabilistic and-or attribute grouping for zero-shot learning. In *UAI*, 2018.

[5] S. Beery, Y. Liu, D. Morris, J. Piavis, A. Kapoor, M. Meister, and P. Perona. Synthetic examples improve generalization for rare classes. *Preprint arXiv:1904.05916*, 2019.

[6] M. Buda, A. Maki, and M. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 2018.

[7] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *NIPS*, 2019.

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

[9] R. Chao, S. Changpinyo, B. Gong, and Sha F. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ICCV*, 2016.

[10] Nitesh V. Chawla, K. Bowyer, L. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *ArXiv*, 2002.

[11] Y. Cui, M. Jia, T. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. *CVPR*, 2019.

[12] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. *CVPR*, 2009.

[13] Chris Drummond. C 4 . 5 , class imbalance , and cost sensitivity : Why under-sampling beats oversampling. 2003.

[14] R. Felix, V. Kumar, I. Reid, and G. Carneiro. Multi-modal cycle-consistent generalized zero-shot learning. In *ECCV*, 2018.

[15] W Feng, j. Cheng, W. Liu, and H. Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 2018.

[16] S. Gidaris and Komodakis.N. Dynamic few-shot visual learning without forgetting. *CVPR*, 2018.

[17] C. Guo, G. Pleiss, Y. Sun, and K. Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

[18] H. Han, W. Wang, and B. Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *ICIC*, 2005.

[19] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 2009.

[20] G. Horn, O. Aodha, Y. Song, A. Shepard, H. Adam, P. Perona, and S. Belongie. The inaturalist challenge 2017 dataset. *ArXiv*, 2017.

[21] R. Jacobs, S. Nowlan, M. Jordan, and G. Hinton. Adaptive mixtures of local experts. *Neural computation*, 1991.

[22] S. et al. Kahou. Combining modality specific deep neural networks for emotion recognition in video. In *ICMI '13*, 2013.

[23] B. Kang, S. Xie, M. Rohrbach, M. Yan, A. Gordo, J. Feng, and Y. Kalantidis. Decoupling representation and classifier for long-tailed recognition. *ICLR*, 2020.

[24] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *ICLR*, 2015.

[25] M Kull, M. Perelló-Nieto, M. Kängsepp, T. Filho, Hao. Song, and P. Flach. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration. *ArXiv*, 2019.

[26] C.H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[27] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *ICCV*, 2017.

[28] K. Liu, Y. Li, N. Xu, and P. Natarajan. Learn to combine modalities in multimodal deep learning. *arXiv preprint arXiv:1805.11730*, 2018.

[29] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019.

[30] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *ArXiv*, 2013.

[31] A. Mishra, M. Reddy, A. Mittal, and H. A. Murthy. A generative model for zero shot learning using conditional variational autoencoders. In *WACV*, 2018.

[32] A. Pambala, T. Dutta, and S. Biswas. Generative model with semantic embedding and integrated classifier for generalized zero-shot learning. In *WACV*, 2020.

[33] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012.

[34] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*, 1999.

[35] S. Pouyanfar, T. Wang, and S. Chen. A multi-label multimodal deep learning framework for imbalanced data classification. In *MIPR*, 2019.

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

[37] S. Ryou, S. Jeong, and P. Perona. Anchor loss: Modulating loss scale based on prediction difficulty. In *ICCV*, 2019.

[38] E. Schönfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata. Generalized zero-shot learning via aligned variational autoencoders. In *CVPR*, 2019.

[39] S. Sharma, N. Yu, M. Fritz, and B. Schiele. Long-tailed recognition using class-balanced experts. *arXiv preprint arXiv:2004.03706*, 2020.

[40] R. Socher, M. Ganjoo, C.D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013.

[41] H. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. *CVPR*, 2016.

[42] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. HS Torr, and T. M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

[43] Y-H. Tsai, L-K. Huang, and R. Salakhutdinov. Learning robust visual-semantic embeddings. In *ICCV*, 2017.

[44] A. Tversky and D. Kahneman. Availability: A heuristic for judging frequency and probability. *Cognitive psychology*, 1973.

[45] G. Van Horn and P. Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017.

[46] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[47] B. Wallace and I. Dahabreh. Improving class probability estimates for imbalanced data. *Knowledge and Information Systems*, 2013.

[48] B. C. Wallace and I. J. Dahabreh. Class probability estimates are unreliable for imbalanced data (and how to fix them). In *2012 IEEE 12th International Conference on Data Mining*.

[49] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NIPS*, 2017.

[50] Y. Xian, C.H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *TPAMI*, 2018.

[51] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata. Feature generating networks for zero-shot learning. In *CVPR*, 2018.

[52] Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning - the good, the bad and the ugly. In *CVPR*, 2017.

[53] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. F-vaegan-d2: A feature generating framework for any-shot learning. *CVPR*, 2019.

[54] H. Zhang and P. Koniusz. Model selection for generalized zero-shot learning. In *ECCV*, 2018.

[55] Hongguang Zhang and Piotr Koniusz. Zero-shot kernel learning. In *CVPR*, 2018.

[56] L. Zhang, T. Xiang, and S. Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, 2017.

[57] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. Range loss for deep face recognition with long-tailed training data. *ICCV*, 2017.

[58] Z. Zhang, P. Luo, C. Loy, and X. Tang. Learning deep representation for face alignment with auxiliary attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 2016.

[59] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal. A generative adversarial approach for zero-shot learning from noisy texts. In *CVPR*, 2018.