Class-Balanced Distillation for Long-Tailed Visual Recognition

Ahmet Iscen
iscen@google.com
Andre Araujo
andrearaujo@google.com
Boqing Gong
bgong@google.com
Cordelia Schmid
cordelias@google.com

Google Research

Abstract

Real-world imagery is often characterized by a significant imbalance of the number of images per class, leading to long-tailed distributions. An effective and simple approach to long-tailed visual recognition is to learn feature representations and a classifier separately, with instance and class-balanced sampling, respectively. In this work, we introduce a new framework, by making the key observation that a feature representation learned with instance sampling is far from optimal in a long-tailed setting. Our main contribution is a new training method, referred to as Class-Balanced Distillation (CBD), that leverages knowledge distillation to enhance feature representations. CBD allows the feature representation to evolve in the second training stage, guided by the teacher learned in the first stage. The second stage uses class-balanced sampling, in order to focus on under-represented classes. This framework can naturally accommodate the usage of multiple teachers, unlocking the information from an ensemble of models to enhance recognition capabilities. Our experiments show that the proposed technique consistently outperforms the state of the art on long-tailed recognition benchmarks such as ImageNet-LT, iNaturalist17 and iNaturalist18.

1 Introduction

Most of the modern computer vision techniques require large amounts of labeled training data in order to learn effective models, *e.g.*, for image classification [21, 36, 49], object detection [22, 48], image retrieval [3, 45, 47] or segmentation [8, 25]. Recently, much research has focused on learning with a smaller number of labels (*e.g.*, few-shot learning [14, 17, 50] or semi-supervised methods [28, 37, 53]), or without any labels (*e.g.*, self-supervision [6, 9, 18]). While these works attempt at reducing the required annotations used for learning, they still tend to make the assumption that the training set is *balanced*, meaning that there exists a similar number of examples per category.

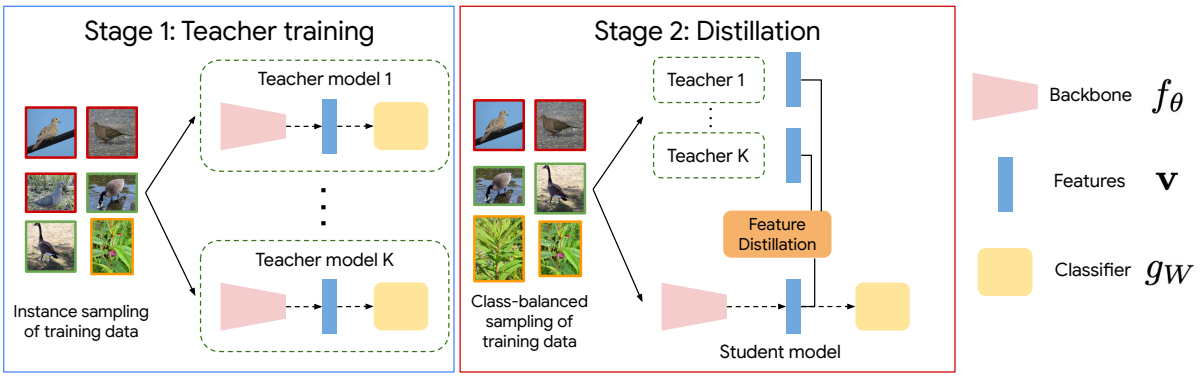

Figure 1: Overview of our Class-Balanced Distillation approach (CBD). In the first stage, we learn one or multiple teacher models with instance sampling. In the second stage, we use class-balanced sampling to distill the features extracted by the teacher model(s) into a student model (right). The backbone is re-trained from scratch with feature distillation and a classification loss in the second stage.

Long-tailed recognition aims to address the real-world setting where a few of the labels are observed with very high frequency (*head*), while most labels appear rarely (*tail*), with a continuum in-between. For example, in natural world datasets like iNaturalist [27], some species are more abundant and easier to photograph than others; similarly, for datasets of human-made and natural landmarks [57], some are much more popular destinations than others. This extreme imbalanced setting makes long-tailed visual recognition a challenging problem, where models often underfit the tail classes. Early works tackle this challenge by different sampling strategies [5, 11] or re-weighting the loss function [12, 34, 39].

A very recent trend in this area is to (explicitly or implicitly) decouple the learning of the feature representation and the classifier into two stages [10, 30, 33, 61, 65]. Typically, these methods first train a model with the imbalanced training data in the first stage, then apply additional operations, such as meta-learning instance-wise weights [30] or augmenting the feature representations of under-represented classes [10], while they fine-tune the model in the second stage. Kang *et al.* [33] focus on the sampling strategies used in both stages and suggest that the feature representations are best learned with instance sampling (*i.e.*, each image having the same probability of being sampled during training) in the first stage, while classifiers are best learned with class-balanced sampling (*i.e.*, each class having the same probability of being sampled) in the second stage.

In our work, we propose a simple, flexible, and effective two-stage framework that makes a more aggressive decoupling of the two stages, allowing the second stage to learn a new feature extractor from scratch and the first stage to learn multiple, complementary models. More specifically, we address two key observations that affect the existing approaches. The first observation is that the features learned by the instance sampling in previous works are far from optimal for a long-tailed dataset, which we demonstrate in Section 4. The second observation is that the class-balanced classifier learning improves *tail* classes, but at the expense of penalizing *head* classes.

We approach both shortcomings by class-balanced knowledge distillation [23], which allows the feature representations to continue evolving in the second stage and benefit from different sampling strategies. Figure 1 illustrates the main components of our method. We train an ensemble of teacher models with instance sampling in the first stage. In the sec-

ond stage, we learn a student model with class-balanced sampling while distilling feature representations from the teachers. Compared with the training and fine-tuning strategy, our approach provides flexibility to the first stage, which can enhance the feature representation by ensembling, and a versatile distillation tool to the second stage, which essentially learns how to combine and evolve the features.

Our contributions are the following:

- A novel two-stage learning method, referred to as Class-Balanced Distillation (CBD), which is suitable for long-tailed recognition datasets, simple to implement, and effective in combining the advantages of instance sampling and class-balanced sampling.
- A feature distillation scheme for ensembling teachers, which efficiently combines feature representations of multiple teachers with different characteristics, including different data augmentations, to further improve its efficacy.
- An extensive experimental evaluation of state-of-the-art long-tailed recognition benchmarks, demonstrating that our model outperforms prior arts substantially, with improvements for *both* head and tail classes.

2 Related Work

Long-Tailed Recognition. The need for handling long-tailed datasets has emerged in many applications, including but not limited to image classification [41, 54], face recognition [4, 62, 63, 64], object detection [38, 46], instance segmentation [20, 26, 55], and multi-label learning [46, 58]. This work focuses on long-tailed image classification, but the proposed approach is generic and may benefit other applications.

Some recent approaches decouple representation and classifier learning in deep long-tailed visual recognition [10, 30, 33, 61, 65]. The representation learning stage often employs instance sampling, followed by different classifier learning methods. Kang *et al.*[33] studied several normalization techniques for the linear classifier layer. Jamal *et al.* [30] proposed a meta-learning algorithm to re-weight both classes and instances. Zhou *et al.* [65] employed an annealing factor to transition the learning from representations to a classifier continuously. Chu *et al.* [10] augmented tail classes in the feature space. In contrast, we propose knowledge distillation [23] as an efficient strategy for two-stage learning in long-tailed recognition, allowing the representation to evolve between different stages. Besides, this enables learning from not just one, but an ensemble of teacher model representations.

Xiang *et al.* [60] have explored knowledge distillation in long-tailed classification for a different purpose from ours. The authors split the original long-tailed training set into a subset of more balanced training sets. An *expert* is learned for each subset, and distillation is used to fuse the experts into a single model. In our work, we instead use the entire dataset for training the model and employ distillation to fuse the information from different teachers and sampling strategies into a single model.

Another line of research in long-tailed recognition is to promote the tail classes when training deep models. These works include sampling the tail more frequently than the head [31], re-weighting losses [12, 30, 31], balancing losses [5, 39, 40, 51], and changing the momentum [52]. Convolutional neural networks with memory modules may better represent the tail [41, 66], and one can also transfer knowledge from the head to the tail [35, 56, 62].

Wu *et al*. [59] introduced a taxonomic classifier to avoid making severe errors at the tail. These methods are orthogonal to ours, and they could complement each other.

Knowledge Distillation. Knowledge distillation [1, 23] refers to transferring information from a *teacher model* to a *student model*. It has been used in a variety of machine learning and computer vision tasks, such as image classification [23], object detection [7], semi-supervised learning [53] and few-shot learning [16]. Typically this involves making the output (logits) of student model similar to the teacher model. In this work, we use a variant which transfers information directly at the feature level. Feature distillation has been successfully used in other tasks, such as asymmetric metric learning [2]. It is also shown that feature distillation helps reduce catastrophic forgetting in incremental learning [24, 29] and domain expansion [32]. In our work, we extend feature distillation to the case of multiple teacher models with different data augmentation and sampling.

3 Method

3.1 Classifier Training

Problem Formulation. We are given a set of n instances (images) $X := \{x_1, \ldots, x_n\}$. Each image is labeled according to $Y := \{y_1, \ldots, y_n\}$ with $y_i \in C$, where $C := \{1, \ldots, c\}$ is a label set for c classes. Let C_j denote the subset of instances labeled as class j, and $n_j = |C_j|$ its cardinality. In this paper, the training set follows a long-tailed distribution. Despite the training set imbalance, the goal is to accurately recognize all classes, so we use a balanced test set to evaluate the classifier.

Model. The learned model (typically a convolutional neural network) takes an input image and outputs class confidence scores. We denote the model by $\phi_{\theta,W} : \mathcal{X} \to \mathbb{R}^c$. It contains two components, corresponding to the learnable parameters θ and W, respectively: 1) a *feature extractor*, mapping each instance x_i to a descriptor $\mathbf{v}_i := f_\theta(x_i) \in \mathbb{R}^d$; 2) a *classifier*, typically consisting of a fully connected layer which output *logits* $\mathbf{z}_i := g_W(\mathbf{v}_i) \in \mathbb{R}^c$, denoting the class confidence scores.

In this work, we model g_W as a *cosine classifier* [17, 43], where the feature descriptors and classifier weights are ℓ_2-normalized before the prediction. Its output becomes $\mathbf{z}_i := \gamma \overline{W}^T \overline{\mathbf{v}}_i$, where $\overline{\mathbf{a}}$ is the ℓ_2-normalized version of $\mathbf{a}$, and γ is a scaling hyper-parameter. For simplicity, we omit the extra notation for ℓ_2-normalization and refer to $\mathbf{v}_i$ and W as the ℓ_2-normalized versions for the rest of this paper.

Training. The model parameters θ and W are typically learned by minimizing the loss of the model's predictions over the training set X:

$$L(X, Y; \theta, W) := \sum_{i=1}^{n} \ell(\sigma(\mathbf{z}_i), y_i), \tag{1}$$

where $\mathbf{z}_i = \phi_{\theta,W}(x_i)$ is the output of the model, $\sigma(.)$ is the softmax activation function, and $\ell(.)$ is the cross-entropy loss function.

3.2 Sampling and Two-Stage Training

In the context of long-tailed problems, different sampling strategies have been used to adjust the data distribution at the training time. We briefly review two sampling methods, which are utilized in this work.

Instance sampling attributes each instance $x_i \in X$ with the same probability to a mini-batch. Hence, the instances from the head classes are sampled more frequently than those from the tail classes due to the long-tailed nature of the dataset, making the model prone to underfitting tail classes. Formally, let us denote by p_j the probability of sampling an instance from class j. Under instance sampling, $p_j = n_j/n$.

Class-balanced sampling addresses the class imbalance by equalizing p_j across classes. Under this strategy, each class has the same probability of being selected, *i.e.*, $p_j = 1/c$ for all $j = 1,\ldots,c$. Even though this strategy balances the data distribution, it also under-utilizes the examples from the head classes. Tail classes are sampled much more frequently compared to head classes. As a result, the model tends to overfit the tail classes and exhibits sub-optimal performance.

Two-stage approaches recently show improved performance for long-tailed recognition [10, 30, 33, 61, 65]. We briefly review a few methods in this section; please see Section 2 for a more thorough review.

Classifier Re-Training (cRT) learns the two components of the model $\phi_{\theta,W}$ with different sampling strategies [33]. The feature extractor f_θ is first trained with instance sampling and then frozen, followed by learning the classifier g_W with class-balanced sampling. The authors argue that the first stage produces generalizable features, while the second stage makes the classifier less biased.

Fine-tuning trains the model $\phi_{\theta,W}$ with instance sampling in the first stage. Then the the entire model $\phi_{\theta,W}$ is fine-tuned with class-balanced sampling, using a small learning rate for some number of epochs. The class-balanced sampling is vital for promoting the classifier's performance on the tail classes.

Discussion. Instance sampling produces better feature representations compared to other sampling strategies [33]. However, the model's classifier is biased towards the head classes. Two-stage methods leverage instance and class-balanced sampling separately to find the right balance between the two sampling strategies. Classifier Re-Training learns the feature representations with instance and the classifier with class-balanced sampling, in this order [33]. While being simple and efficient, it has at least two shortcomings: (1) the feature representations tend to mostly focus on the head classes due to the instance sampling in the first stage; (2) the second-stage, class-balanced classifier learning, could overcompensate tail classes, leading to reduced performance for the head classes.

3.3 Class-Balanced Distillation (CBD)

To overcome the shortcomings in existing two-stage methods, we enhance the two-stage learning for long-tailed recognition by improving both (1) the feature representations for tail classes and (2) the classifier for head classes. We leverage distillation [23] to do so. Figure 1 illustrates our overall approach. In the first stage, we use instance sampling to train a teacher model $\widehat{\phi}_{\widehat{\theta},\widehat{W}}$. In the second stage, we adopt class-balanced sampling and yet learn our student model $\phi_{\theta,W}$ from scratch by adding a feature distillation loss.

The feature distillation loss encourages the feature extractor f_θ of the student to heed the teacher's feature extractor. It also amends the student's feature extractor to facilitate the classifier g_W. It reuses but does not fully inherit the first-stage's knowledge, leaving room

for improvement with the class-balanced training. The loss objective from Eq. (1) becomes:

$$L(X,Y;\theta,W) := \sum_{i=1}^{n} (1-\alpha)\cdot \ell\left(\sigma(\mathbf{z}_i), y_i\right) + \alpha \cdot \left(\beta \ell_F\left(\mathbf{v}_i, \widehat{\mathbf{v}}_i\right)\right), \tag{2}$$

where $\widehat{\mathbf{v}}_i = \widehat{f}_{\widehat{\theta}}(x_i)$ is the feature descriptor produced by the teacher model, and $\ell_F(\mathbf{v},\mathbf{x}) = 1-\cos(\mathbf{v},\mathbf{x})$ tries to minimize the cosine distance between two feature descriptors. The hyper-parameter α controls the amount of distillation compared to the cross entropy loss, and β is a scaling parameter.

Feature-Level vs. Classifier-Level Distillations. Note that our objective function differs from the common knowledge distillation [1, 23], which is applied to to the classifier level rather than the feature level:

$$L(X,Y;\theta,W) := \sum_{i=1}^{n} (1-\alpha)\cdot \ell\left(\sigma(\mathbf{z}_i), y_i\right) + \alpha \cdot T^2 \cdot \ell\left(\sigma(\mathbf{z}_i/T), \sigma(\widehat{\mathbf{z}}_i/T)\right), \tag{3}$$

where $\widehat{\mathbf{z}}_i = \widehat{\phi}_{\widehat{\theta},\widehat{W}}(x_i)$ is the teacher model's output, and T is the temperature parameter used for distillation [23].

We experimentally show that the feature-level distillation is advantageous over the conventional classifier-level distillation. In the context of long-tailed recognition, the teacher's classifier is highly biased towards the head classes. By distilling only at the feature level (Eq. (2)), we encourage the student to heed the teacher's feature extraction mechanism, not the classification function, to avoid learning a classifier that is significantly biased to the head.

Distilling Ensemble of Teachers. Unlike the existing two-stage methods which learn a classifier (e.g., by cRT) or fine-tune the model, it is straightforward to use the proposed CBD to further transfer knowledge from multiple teacher models. The resulting student model, in this case, tends to have stronger regularization properties and reduced over-fitting [23].

To enable such capabilities, we train different teacher models with different characteristics. More specifically, we train two types of teacher models with different data augmentations. The *Standard* model relies on standard data-augmentation transformations during training, such as random crop and flip. The *Data Augmentation* model uses additional data transformations, such as color jitter and Gaussian noise ($\sigma = 0.01$) in addition to random crop and flip. When training multiple models of the same type, we start from different initial random seeds. Different initial random seeds affect the initialization of the model parameters as well as the order of classes sampled during the training. Regardless of the teacher model type, the *standard* model is always used when training the student model in the second stage, according to our preliminary experiments.

Let $\widehat{\phi}^k_{\widehat{\theta}^k,\widehat{W}^k}$ denote the k-th teacher model. When training the student model $\phi_{\theta,W}$ in the second stage, we combine the knowledge from multiple teachers with the following objective:

$$L(X,Y;\theta,W) := \sum_{i=1}^{n} (1-\alpha)\cdot \ell\left(\sigma(\mathbf{z}_i), y_i\right) + \alpha \cdot \left(\beta \ell_F\left(h(\mathbf{v}_i), \widehat{\mathbf{V}}_i\right)\right), \tag{4}$$

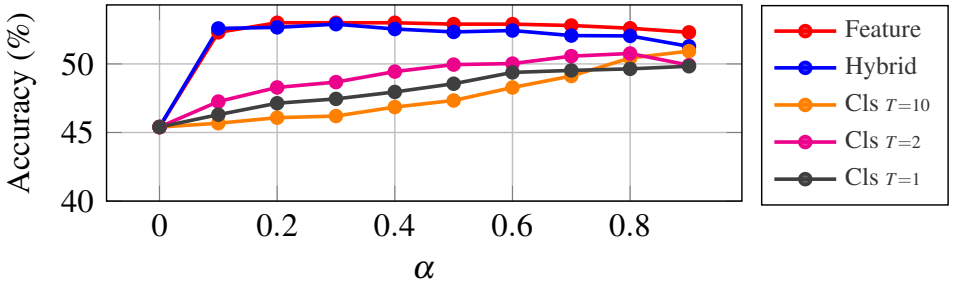

Figure 2: Impact of α in different distillation techniques. Experiments are conducted with ResNet-50 on the ImageNet-LT validation set.

where $\widehat{\mathbf{V}}_i = [\widehat{\mathbf{v}}_i^1, \ldots, \widehat{\mathbf{v}}_i^K]$ concatenates K feature descriptors output by the teacher models, and $h : \mathbb{R}^d \rightarrow \mathbb{R}^{d \cdot K}$ is a linear layer which maps the feature descriptor $\mathbf{v}_i$ to a higher dimensional space where the cosine distance can be computed (the classifier g_W is then stacked on top of $h(\mathbf{v}_i)$). We refer to this variant as CBD_{ENS} in our experiments.

The feature extractors of the teacher models account for the complementary information of the long-tailed training set. By jointly distilling knowledge from them, we transfer the enhanced feature representations to the student feature extractor f_θ, which eases the learning of the classifier g_W.

4 Experiments

4.1 Experimental Setup

Datasets. We experiment with three long-tailed datasets, namely, ImageNet-LT [41], iNaturalist18 [27] and iNaturalist17[54]. Please refer to Section A.1 of the appendix for details of each dataset. Top-1 accuracy is the evaluation metric for all experiments. We also follow the protocol in [41] to report the accuracies for *many-shot* classes (more than 100 images per class), *mid-shot* classes (between 20 and 100 images) and *few-shot* classes (less than 20 images), separately.

Implementation Details. We use the ResNet-{50,152} [21] architectures for ImageNet-LT, and ResNet-{50,101} for iNaturalist17 and iNaturalist18. See Section A.2 of the appendix for training details. The scaling parameter in Eq. (2) is set to $\beta = 100$ based on the accuracy in the ImageNet-LT validation set (see Section A.3 in Appendix). Other parameters, such as α and the number of teacher models K are chosen based on the experiments in Section 4.2.

4.2 Ablation Study

We study the impact of some of the hyper-parameters and components of CBD. All experiments in this section are evaluated on the validation set of ImageNet-LT.

Distillation. We first evaluate different distillation techniques, *i.e.* feature distillation (Eq. (2)) and classification distillation (Eq. (3)) in Figure 2. We report classification distillation with different temperature T values. We also show the impact of the distillation coefficient α in the same figure. This parameter controls the strength of distillation in the loss function, see Eq. (2).

	Vanilla	Data Aug.	Acc. (%)
$K=1$	✓	-	52.7
	-	✓	53.9
$K=2$	✓✓	-	54.2
	-	✓✓	55.5
	✓	✓	56.2
$K=3$	✓	✓✓	56.7
$K=4$	✓✓	✓✓	56.9
$K=5$	✓✓	✓✓✓	56.9

Table 1: **Different ensembles of teachers.** Comprehensive evaluation of different types of K teacher models on the ImageNet-LT validation set with ResNet-50. Each row corresponds to a different ensemble. Multiple ✓refer to multiple models of the same type trained with different random seeds.

Figure 2 shows that $T=2$ achieves the highest accuracy for classification distillation. Feature distillation outperforms all variants of classification distillation. It also outperforms a variant (Hybrid) which combines feature and classification distillation ($T=2$) together. Feature distillation is also more stable for different α. This is expected, as the first stage model (instance sampling) produces relatively good features but a sub-optimal classifier. Therefore, it is more beneficial to transfer information directly from the features, rather than the classifier. It is also shown that feature distillation remains relatively stable when $\alpha > 0$. Note that $\alpha = 0$ means that no distillation loss term is used during the training, which is equivalent to class-balanced sampling. We set $\alpha = 0.4$, which gives the top performance in Figure 2, for the remainder of our experiments.

Number of teacher models. We train K teacher models when ensembling is used. The ensemble may contain teacher models of different types, *i.e.* *standard* and *data augmentation*. When using the same type multiple times, *e.g.* two *standard* models, each model is trained with different random seeds to achieve diversity between models. These teacher models are then fused into a single model with distillation – Eq. (4). We refer to this variant of our method as CBD_{ENS}.

Table 1 shows the impact of different number of *standard* and *data augmentation* models when used in an ensemble. We report all combinations for $K=1$ and $K=2$, but only show the variant with the highest accuracy for $K>2$. For $K=1$, the *data augmentation* model achieves a better performance than the *standard* model. Nevertheless, we achieve the best accuracy with some combination of *standard* and *data augmentation* models for $K>1$. The validation accuracy saturates after $K=4$, therefore we use the $K=4$ for CBD_{ENS} for the remainder of our experiments.

4.3 Comparison with Baselines

We compare our method against various baselines. The results are reported on the ImageNet-LT test set. Please refer to Section 3.2 more detailed description of each baseline. For single-stage models, we evaluate *standard* and *data augmentation* models separately with instance and class balanced sampling strategies. For two-stage models, we evaluate *fine-tuning* [1] and

[1]The network is fine-tuned for 10 epochs with 0.01 learning rate in the second stage, which was the best setup for this method on ImageNet-LT

Method	Many-shot	Mid-shot	Few-shot	All
Standard - Instance	66.6	40.4	13.0	46.7
Standard - Class Bal.	60.4	40.0	14.3	44.3
Data Aug. - Instance	66.2	38.6	11.2	45.4
Data Aug. - Class Bal.	58.4	45.2	19.9	46.8
Standard - Fine-tuning	62.8	46.1	24.8	49.6
Standard - Classifier Re-Training	62.9	46.0	25.7	49.8
Data Aug. - Fine-tuning	63.1	48.4	26.9	51.1
Data Aug. - Classifier Re-Training	62.2	47.1	27.8	50.3
Teacher Ensemble	**71.6**	44.4	13.8	50.7
Ours - CBD	65.2	48.0	25.9	51.6
Ours - CBD_{ENS}	**68.5**	**52.7**	**29.2**	**55.6**

Table 2: **Baseline comparison.** Comprehensive evaluation on ImageNet-LT (test set) with the ResNet-50 architecture. The accuracy for many-shot , mid-shot and few-shot classes are reported separately.

classifier-retraining, which is our re-implementation of *cRT* [33] with the cosine classifier. We also evaluate the *data augmentation* version of two-stage baselines, where the first stage is trained with the *data augmentation* model and the second stage is trained with the *standard* model. Finally, we evaluate the *Teacher Ensemble* baseline, which simply takes the average output of teacher models during testing.

Table 2 reports the comparisons against the baselines. We report the accuracy of many-shot, mid-shot, and few-shot classes separately, in addition to the overall accuracy for all classes. When compared to other two-stage models, both CBD and CBD_{ENS} show significant improvements. This confirms that our method is a better option as a two-stage model, even if a single teacher model is used (CBD). Note that the two-stage baselines reduce the accuracy of many-shot classes in the second stage. Ensemble baselines improve the performance for many-shot classes, but show no improvements for mid-shot and few-shot classes. This is not the case for CBD_{ENS} on ImageNet-LT, which shows improvements for *all* class types. We also observe that the *data augmentation* model does not show any significant improvements except for CBD_{ENS}. This demonstrates that our method is capable of combining diverse models in the most effective way.

Longer training of baselines. In order to justify that the improvement is not only due to the longer training, we train the *Standard - Instance model* for two times the number of epochs. This means that the model is trained for 180 epochs on ImageNet-LT and 400 epochs on iNaturalist18, *i.e.* the total number of epochs it takes to train CBD. We obtain 47.1 and 64.7 overall accuracy for ImageNet-LT and iNaturalist18, respectively. When compared to the *Standard - Instance model* on Table 2, the improvement is minimal, which confirms that the improvements of CBD are not due to longer training.

We also repeat the same procedure for the *Classifier re-Training* baseline, where we train the linear model for 90 (ImageNet-LT) and 200 (iNaturalist18) epochs in the second stage. We obtain 50.1 and 67.2 for ImageNet-LT and iNaturalist18, respectively. When compared to the *Classifier re-Training* model on Table 2, the gains are again minimal. This again confirms that the efficacy of CBD and CBD_{ENS} is not due to the longer training times.

Complexity. CBD requires higher training complexity compared to other baselines. A network is trained from scratch in each stage. We demonstrate that if other baselines (*Instance*

ImageNet-LT		
Method	R-50	R-152
LWS [33]	47.7	50.5
cRT [33]	47.3	50.1
cRT+SSP [61]	51.3	-
Logit Adj. [44]	51.1	52.1
ELF(LDAM) [15]	52.0	-
Ours - CBD	51.6	53.9
Ours - CBD_{ENS}	55.6	**57.7**

iNaturalist18		
Method	R-50	R-101
LWS [33]	69.5	69.7
cRT [33]	68.2	70.7
cRT+SSP [61]	68.1	-
Logit Adj. [44]	68.4	70.8
ELF(LDAM) [15]	69.8	-
Ours - CBD	68.4	70.5
Ours - CBD_{ENS}	**73.6**	**75.3**

iNaturalist17		
Method	R-50	R-101
CB [10]	58.1	60.9
Rethinking CB [30]	59.4	-
Feature Aug. [10]	62.0	65.9
cRT [33]	63.9	65.2
BBN [65]	65.8	-
Ours - CBD	64.6	66.5
Ours - CBD_{ENS}	**69.3**	**71.3**

Table 3: **State-of-the-art comparison.** Comparison of CBD variants against the state of the art with ResNet-50 and ResNet-152.

and *Classifier re-Training*) are given the same amount of training resources, their performance is still lower than CBD. CBD_{ENS} requires training multiple ($K = 4$) teacher models in the first stage, which further increases the training complexity. However, the teacher models do not interact with each other during the training, which means that all teacher models can be trained in parallel, which can significantly improve the overall time for training. Memory consumption does not depend on the scale of the dataset, as it is fixed (e.g. 4 ResNet-50 models) regardless of the size of the dataset. Note that both CBD and CBD_{ENS} require a single model during the inference. Therefore, the test time efficiency remains the same as for all the other baselines.

4.4 Comparison with State of the Art

Table 3 compares CBD and CBD_{ENS} with $K = 4$ teachers to the state of the art on ImageNet-LT, iNaturalist18 and iNaturalist17 datasets, respectively. Our method shows consistent improvement for all datasets with different network architectures. On ImageNet-LT, we observe 3.6% improvement with CBD_{ENS} (ResNet-50) over the prior best. CBD_{ENS} outperforms the state of the art on iNaturalist18 (iNaturalist17) by 3.8% (3.5%) with ResNet-50. Relative improvement is even higher when a larger network is used; we observe 5.5% improvement over state of the art with CBD_{ENS} with ResNet-152 in ImageNet-LT, and 4.5% improvement over state of the art in iNaturalist18 with ResNet101. See Section A.7 of the Appendix for result for each class split separately.

To investigate the compatibility of CBD with existing methods, we also include a variant where the loss function in CBD is replaced by the loss function proposed in the work of Menon *et al.* [44]. On ImageNet-LT, CBD + Logit Adjustment [44] gains 0.6% over CBD, i.e., it obtains 52.2 accuracy, and CBD_{ENS} + [44] improves 0.5% over CBD_{ENS}, i.e., it achieves 56.1 accuracy.

5 Conclusions

In this paper, we have introduced a new two-stage method for long-tailed recognition called CBD. Our approach leverages knowledge distillation to combine information from two sampling strategies. Both the feature representation and the classifier evolve between stages, leading to a more effective model. We thoroughly evaluate the effectiveness of our method by comparing it against baselines and previous work. Our experiments demonstrate that CBD significantly improves the state of the art in long-tailed recognition benchmarks.

References

[1] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *ACM SIGKDD*, 2006.

[2] Mateusz Budnik and Yannis Avrithis. Asymmetric metric learning for knowledge transfer. *arXiv preprint arXiv:2006.16331*, 2020.

[3] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. *ECCV*, 2020.

[4] Dong Cao, Xiangyu Zhu, Xingyu Huang, Jianzhu Guo, and Zhen Lei. Domain balancing: Face recognition on long-tailed domains. In *CVPR*, 2020.

[5] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *NeurIPS*, 2019.

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. *ECCV*, 2018.

[7] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *NeurIPS*, 2017.

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *ICML*, 2020.

[10] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. *ECCV*, 2020.

[11] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *CVPR*, 2018.

[12] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. *CVPR*, 2019.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[14] Matthijs Douze, Arthur Szlam, Bharath Hariharan, and Hervé Jégou. Low-shot learning with large-scale diffusion. In *CVPR*, 2018.

[15] Rahul Duggal, Scott Freitas, Sunny Dhamnani, Duen Horng, Jimeng Sun, et al. Elf: An early-exiting framework for long-tailed classification. *arXiv preprint arXiv:2006.11979*, 2020.

[16] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *ICCV*, 2019.

[17] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018.

[18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 2020.

[19] Samantha Guerriero, Barbara Caputo, and Thomas Mensink. Deepncm: Deep nearest class mean classifiers. *ICLR, Worskhop Track*, 2018.

[20] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *CVPR*, 2017.

[23] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[24] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, 2019.

[25] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *CVPR*, 2018.

[26] Xinting Hu, Yi Jiang, Kaihua Tang, Jingyuan Chen, Chunyan Miao, and Hanwang Zhang. Learning to segment the tail. In *CVPR*, 2020.

[27] iNaturalist 2018 competition dataset. iNaturalist 2018 competition dataset. https://github.com/visipedia/inat_comp/tree/master/2018, 2018.

[28] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *CVPR*, 2019.

[29] Ahmet Iscen, Jeffrey Zhang, Svetlana Lazebnik, and Cordelia Schmid. Memory-efficient incremental learning through feature adaptation. In *ECCV*, 2020.

[30] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *CVPR*, 2020.

[31] Ren Jiawei, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *NeurIPS*, 2020.

[32] Heechul Jung, Jeongwoo Ju, Minju Jung, and Junmo Kim. Less-forgetful learning for domain expansion in deep neural networks. In *AAAI*, 2018.

[33] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *ICLR*, 2020.

[34] Salman Khan, Munawar Hayat, Syed Waqas Zamir, Jianbing Shen, and Ling Shao. Striking the right balance with uncertainty. In *CVPR*, 2019.

[35] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *CVPR*, 2020.

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[37] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

[38] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *CVPR*, 2020.

[39] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[40] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *CVPR*, 2020.

[41] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019.

[42] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[43] Chunjie Luo, Jianfeng Zhan, Xiaohe Xue, Lei Wang, Rui Ren, and Qiang Yang. Cosine normalization: Using cosine similarity instead of dot product in neural networks. In *ICANN*, 2018.

[44] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *ICLR*, 2021.

[45] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, 2017.

[46] Junran Peng, Xingyuan Bu, Ming Sun, Zhaoxiang Zhang, Tieniu Tan, and Junjie Yan. Large-scale object detection in the wild from imbalanced multi-labels. In *CVPR*, 2020.

[47] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018.

[48] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2014.

[50] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

[51] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *CVPR*, 2020.

[52] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *NeurIPS*, 2020.

[53] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.

[54] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist species classification and detection dataset. In *CVPR*, 2018.

[55] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. *arXiv preprint arXiv:2007.11978*, 2020.

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

[57] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *CVPR*, 2020.

[58] Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *ECCV*, 2020.

[59] Tz-Ying Wu, Pedro Morgado, Pei Wang, Chih-Hui Ho, and Nuno Vasconcelos. Solving long-tailed recognition with deep realistic taxonomic classifier. *arXiv preprint arXiv:2007.09898*, 2020.

[60] Liuyu Xiang, Guiguang Ding, and Jungong Han. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *ECCV*, 2020.

[61] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. *NeurIPS*, 2020.

[62] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for face recognition with under-represented data. In *CVPR*, 2019.

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

[64] Yaoyao Zhong, Weihong Deng, Mei Wang, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang. Unequal-training for deep face recognition with long-tailed noisy data. In *CVPR*, 2019.

[65] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *CVPR*, 2020.

[66] Linchao Zhu and Yi Yang. Inflated episodic memory with region self-attention for long-tailed visual recognition. In *CVPR*, 2020.