Learning From Multiple Experts: Self-paced Knowledge Distillation for Long-tailed Classification

Liuyu Xiang[1], Guiguang Ding[1,(✉)], and Jungong Han[2]

[1] School of Software, Tsinghua University, Beijing, China;
Beijing National Research Center for Information Science and Technology (BNRist)
xiangly17@mails.tsinghua.edu.cn, dinggg@tsinghua.edu.cn
[2] Computer Science Department, Aberystwyth University, SY23 3FL, UK
jungonghan77@gmail.com

Abstract. In real-world scenarios, data tends to exhibit a long-tailed distribution, which increases the difficulty of training deep networks. In this paper, we propose a novel self-paced knowledge distillation framework, termed Learning From Multiple Experts (LFME). Our method is inspired by the observation that networks trained on less imbalanced subsets of the distribution often yield better performances than their jointly-trained counterparts. We refer to these models as 'Experts', and the proposed LFME framework aggregates the knowledge from multiple 'Experts' to learn a unified student model. Specifically, the proposed framework involves two levels of adaptive learning schedules: Self-paced Expert Selection and Curriculum Instance Selection, so that the knowledge is adaptively transferred to the 'Student'. We conduct extensive experiments and demonstrate that our method is able to achieve superior performances compared to state-of-the-art methods. We also show that our method can be easily plugged into state-of-the-art long-tailed classification algorithms for further improvements.

1 Introduction

Deep convolutional neural networks (CNNs) have achieved remarkable success in various computer vision applications such as image classification, object detection and face recognition. Training a CNN typically relies on carefully collected large-scale datasets, such as ImageNet [6] and MS COCO [30] with hundreds of examples for each class. However, collecting such a uniformly distributed dataset in real-world scenarios is usually difficult since the underlying natural data distribution tends to exhibit a long-tailed property with few majority classes (head) and large amount of minority classes (tail) [37, 51, 36]. When deep models are trained under such imbalanced distribution, they are unlikely to achieve the expected performances which necessitates developing relevant algorithms.

Recent approaches tackle this problem mainly from two aspects. First is via re-sampling schemes or cost-sensitive loss functions to alleviate the negative impact of data imbalance. Second is by head-to-tail knowledge transfer, where

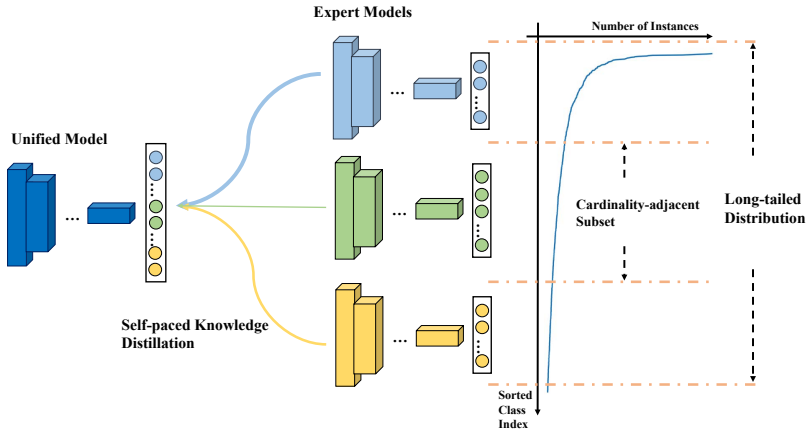

Fig. 1. Schematic illustration of our proposed method.

prior knowledge or induction bias is learned from the richly annotated classes and generalize to the minority ones.

Orthogonal to the above two perspectives, we propose a novel self-paced knowledge distillation method which can be easily plugged into previous methods. Our method is motivated by an interesting observation that learning a more uniform distribution with fewer samples is sometimes easier than learning a long-tailed distribution with more samples [37]. We first introduce four metrics to measure the *'longtailness'* of a long-tailed distribution. We then show that if we sort all the categories according to their cardinality, then splitting the entire long-tailed dataset into subsets will lead to a smaller *longtailness*, which indicates that they suffer a less severe data imbalance problem. Therefore training a CNN on these subsets is expected to perform better than their jointly-trained counterparts. For clarity, we refer to such a subset as **cardinality-adjacent subset**, and the CNN trained on these subsets as **Expert Models**.

Once we acquire the well-trained expert models, they can be utilized as guidance to train a unified student model. If we take a look at human learning process as students, we can conclude two characteristics: 1) the student often takes various courses from easy to hard, 2) as the learning proceeds, the student acquires more knowledge from self-learning than from teachers and he/she may even exceed his/her teachers. Inspired by these findings, we propose a *Learning From Multiple Experts (LFME)* framework with two levels of adaptive learning schemes, termed as self-paced expert selection and curriculum instance selection. Specifically, the self-paced expert selection automatically controls the impact of knowledge distillation from each expert, so that the learned student model will gradually acquire the knowledge from the experts, and finally exceed the expert. The curriculum instance selection, on the other hand, designs a curriculum for the unified model where the training samples are organized from easy to hard, so

that the unified student model will receive a less challenging learning schedule, and gradually learns from easy to hard samples. A schematic illustration of our LFME framework is shown in Fig. 1.

To verify the effectiveness of our proposed framework, we conduct extensive experiments on three benchmark long-tailed classification datasets, and show that our method is able to yield superior performances compared to the state-of-the-art methods. It is worth noting that our method can be easily combined with other state-of-the-art methods and achieve further improvements. Moreover, we conduct extensive ablation studies to verify the contribution of each component.

Our contributions can be summarized as follows: (1) We introduce four metrics for evaluating the *'longtailness'* of a distribution and further propose a *Learning From Multiple Experts* knowledge distillation framework. (2) We propose two levels of adaptive learning schemes, i.e. model level and instance level, to learn a unified *Student* model. (3) Our proposed method achieves state-of-the-art performances on three benchmark long-tailed classification datasets, and can be easily combined with state-of-art methods for further improvements.

2 Related Work

Long-tailed, Data-imbalanced Learning. The long-tailed learning problem has been comprehensively studied due to the prevalence of data imbalance problem [17, 37]. Most previous methods tackle this problem using either re-sampling, re-weighting or 'head-to-tail' knowledge transfer. Re-sampling methods either adopt over-sampling on tail classes [3, 16] or use under-sampling [9, 44, 22] on head classes. On the other hand, various cost-sensitive loss functions have been proposed in the literature to re-weight majority and minority instances [26, 51, 8, 29, 20, 2, 5]. Among them, Range Loss [51] minimizes the range of each class to enhance the learning towards face recognition with long-tail while Focal Loss [29] down-weights the loss assigned to well-classified examples to deal with class imbalance in object detection. Label-Distribution-Aware Margin Loss (LDAM) [2], on the other hand, encourages minority classes to have larger margins.

Researchers also try to employ head-to-tail knowledge transfer for data imbalance. In [45, 46] a transformation from minority classes to majority classes regressors/classifiers is learned progressively while in [31] a meta embedding equipped with a feature memory is proposed for such knowledge transfer.

Few-shot learning methods [10, 15, 47] also try to generalize knowledge from a richly annotated dataset to a low-shot dataset. This is often achieved by training a meta-learner [12, 48] from the many-shot classes and then generalize to new few-shot classes. However, different from few-shot learning algorithms, we mainly focus on learning a continuous spectrum of data distribution jointly, rather than focus solely on the few-shot classes.

Knowledge Distillation. The idea of knowledge distillation was first introduced in [19] for the purpose of model compression where a student network is trained to mimic the behavior of a teacher network so that the knowledge is compressed to the compact student network. Then the distillation target is further

extended to hidden layer features [41] and visual attention [50], where attention map from the teacher model is transferred to the student. Apart from distilling for model compression, knowledge distillation is also proved to be effective when the teacher and the student have identical architecture. i.e. self-distillation [49, 11]. Knowledge distillation is also applied in other areas such as continual learning [28, 39], semi-supervised learning [32, 7] and neural style transfer [21].

Curriculum and Self-paced Learning. The basic idea of curriculum learning is to organize samples or tasks in ascending order of difficulty, and it has been widely adopted for weakly supervised learning [27, 24, 14] and reinforcement learning[43, 33, 34]. Apart from designing easy-to-hard curriculums based on prior knowledge, efforts have also been made to incorporate learning process to dynamically adjust the curriculum. In [23] a self-paced curriculum determined by both prior knowledge and learning dynamics is proposed. In [13] a multi-armed bandit algorithm is used to determine a syllabus, where the rate of increase in prediction accuracy and network complexity are utilized as reward signals. In [40], meta learning is employed to assign weights to training samples based on gradient directions.

Table 1. Comparison of entire distribution and subsets under four metrics. Larger values indicate more *longtailness*.

Metric \ Accuracy	I_{Ratio}	I_{KL}	I_{Abs}	I_{Gini}
Entire	256.0	0.707	0.769	0.524
Many-shot	12.8	0.278	0.481	0.322
Medium-shot	4.7	0.122	0.356	0.235
Low-shot	4.0	0.099	0.320	0.209

3 Motivation and Metrics for Evaluating Data Imbalance

The problem we address in this work is to train a CNN on a long-tailed classification task. Our method is inspired by an interesting empirical finding that, training a CNN on a balanced dataset with fewer samples sometimes leads to superior performances than on a long-tailed dataset with more samples. As the experiment in [37] reveals that even when 40% of the positive samples are left out for the representation learning, the object detection performances can still be surprisingly improved a bit due to a more uniform distribution. This observation successfully emphasizes the importance of learning a balanced, uniform distribution. To learn a more balanced distribution, a natural question to ask would be, how to measure the data imbalance. Since almost every manually collected dataset more or less contains various number of samples per class. To this end, we first introduce four metrics for data imbalance measurement.

For a long-tailed dataset, if we denote N, N_i, C to be the total number of samples, number of samples in class i, and number of classes respectively, then the four metrics for measuring data imbalance is introduced as follows:
Imbalance Ratio [42] is defined as the ratio between the largest and the smallest number of samples: $I_{Ratio} = \frac{N_{i_{max}}}{N_{i_{min}}}$

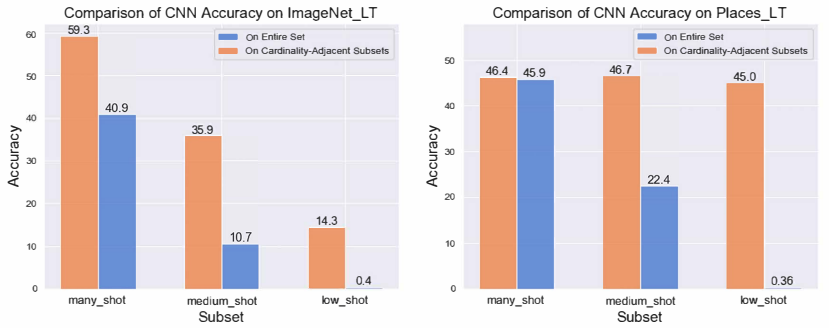

Fig. 2. Comparison of CNN performances trained on cardinality-adjacent subsets and the entire dataset.

Imbalance Divergence is defined as the KL-Divergence between the long-tailed distribution and the uniform distribution:

$$I_{KL} = D(P||Q) = \sum_i p_i log \frac{p_i}{q_i}$$

where $p_i = \frac{N_i}{N}$ is the class probability for class i, and $q_i = \frac{1}{C}$ denotes the uniform probability.
Imbalance Absolute Deviation [4] is defined as the sum of aboslute distance between each long-tailed and uniform probability:

$$I_{Abs} = \sum_i |\frac{1}{C} - \frac{N_i}{N}|$$

Gini Coefficient is defined as

$$I_{Gini} = \frac{\sum_{i=1}^{C}(2i - C - 1)N_i}{C\sum_{i=1}^{C} N_i}$$

where i is the class index when all classes are sorted by cardinality in ascending order.

The last three metrics all measure the distance between the current long-tailed distribution and a uniform distribution. For all four metrics, smaller values indicate a more uniform distribution. Having these metrics at hand, we show that for a long-tailed dataset, if we sort the classes by their cardinality, i.e. number of samples, then a subset of the classes with adjacent cardinality will become

less long-tailed under these imbalance measurements. As Table 2 shows that, if we split the long-tailed benchmark dataset ImageNet-LT into three splits (many-shot, medium-shot, few-shot) according to class cardinality (following [31]), then all four metrics become smaller, which indicates that these subsets become less imbalanced than the original. Then the CNNs trained on these subsets are expected to perform better than their jointly-trained counterparts. We verify this assumption on two long-tailed benchmark datasets ImageNet-LT and Places-LT [31]. As shown in Fig. 2 that CNNs trained on these subsets outperform the joint model by a large margin. This empirical result also accords with our intuition that a continuous spectrum subset of adjacent classes is more balanced in terms of cardinality, and the learning process involves less interference between the majority and the minority.

4 The LFME framework

4.1 Overview

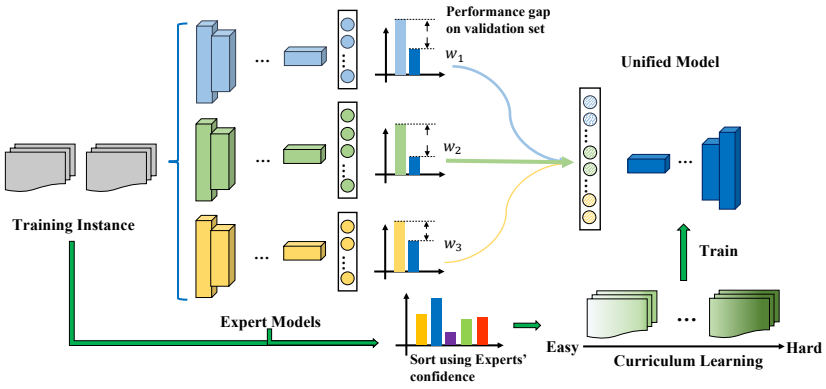

Fig. 3. Overview of the LFME framework.

In this section, we describe the proposed LFME framework in detail. Formally, given a long-tailed dataset D with C classes, we split the entire set of categories into L cardinality-adjacent subsets $S_1, S_2, ..., S_L$ using $L-1$ thresholds $T_1, T_2, ..., T_{L-1}$ such that $S_i = \{c|T_i \leq N_c \leq T_{i+1}\}$, where N_c is class c's cardinality. Then we train L expert models on each of the cardinality-adjacent subset and denote them as $\mathcal{M}_{E_1}, \mathcal{M}_{E_2}, ..., \mathcal{M}_{E_L}$. These expert models serve to 1) provide output logits' distribution for knowledge transfer 2) provide output confidence as instance difficulty cues. These information enables us to develop self-paced and curriculum learning schemes from both model level and instance level. From the perspective of **self-paced expert selection**, we adopt a weighted knowledge distillation between the output logits from the expert models and the student

model. As the learning proceeds, the student will gradually approach the experts' performances. In such cases, we do not want the experts to limit the learning process of the student. To achieve this goal, we introduce a self-paced weighted scheme based on the performance gap on the validation set between the expert models and the student model. As the student model acquires knowledge from both data and the expert models, the importance weight of knowledge distillation will gradually decrease, and finally the unified student model is able to achieve comparable or even superior performance compared to the experts. From the perspective of **curriculum instance selection**, given the confidence scores from the Expert models, we re-organize the training data from easy to hard, i.e. from low-confidence to high-confidence. Then we exploit a soft weighted instance selection scheme to conduct such curriculum, so that easy samples are trained first, then harder samples are added to the training set gradually. This progressive learning curriculum has proved to be beneficial for training deep models [1]. Finally, with the two levels of self-paced and curriculum learning schemes, the knowledge from the expert models will be gradually transferred to the unified student model. An overview of the LFME framework is shown in Fig 3.

4.2 Self-paced Expert Selection

Once we acquire the well-trained expert models, we feed the training data and obtain their output predictions. Then we employ knowledge distillation as an extra supervision signal to the student model. Specifically, for the expert $\mathcal{M}_{E_l}$ trained on l-th cardinality-adjacent subset S_l, if we denote $\mathbf{z}^{(l)}, \hat{\mathbf{z}}$ to be the output logits of the current expert model and the current student model respectively, then the knowledge distillation loss for expert $\mathcal{M}_{E_l}$ is given by:

$$L_{KD_l} = -H(\tau(\mathbf{z}^{(l)}), \tau(\hat{\mathbf{z}}^{(l)})) = -\sum_{i=1}^{|S_l|} \tau(z_i^{(l)}) \log(\tau(\hat{z_i}^{(l)}))$$

where $\hat{\mathbf{z}}^{(l)} = \hat{\mathbf{z}}_{c \in S_l}$ is the student logits for classes in S_l and

$$\tau(z_i^{(l)}) = \frac{\exp(z_i^{(l)}/T)}{\sum_j \exp(z_j^{(l)}/T)}, \quad \tau(\hat{z_i}^{(l)}) = \frac{\exp(\hat{z_i}^{(l)}/T)}{\sum_j \exp(\hat{z_j}^{(l)}/T)}$$

are the output probabilities using Softmax with temperature T. T is usually set to be greater than 1 to increase the weight for smaller probabilities. In this way, for each expert model $\mathcal{M}_{E_l}$ we have its knowledge distillation loss to transfer its knowledge to the student model, and there are L losses in total, corresponding to L experts trained on L cardinality-adjacent subsets. The most straightforward way to aggregate these losses would be simply summing them up. However, this could be problematic, since as the learning process goes on, the student model's performance will gradually approach or even exceed the expert's. In such cases, we do not want the expert models become performance ceilings for the student model, and we wish to gradually weaken the guidance from the experts as the data-driven learning proceeds.

To achieve this goal, we propose a Self-paced Expert Selection scheme based on the performance gap between the student and the experts. In the experiments, we use the Top-1 Accuracy on the validation set after each training epoch as the measurement for performance gap. If we denote the Top-1 Accuracy of the student model $\mathcal{M}$ and the expert model $\mathcal{M}_{E_l}$ at epoch e to be $Acc_{\mathcal{M}}$ and Acc_{E_l} respectively, then a weighting scheme is defined as follows:

$$w_l = \begin{cases} 1.0 & if Acc_{\mathcal{M}} \leq \alpha Acc_{E_l} \\ \dfrac{Acc_{E_l} - Acc_{\mathcal{M}}}{Acc_{E_l}(1-\alpha)} & if Acc_{\mathcal{M}} > \alpha Acc_{E_l} \end{cases}$$

and w_l is updated at the end of each epoch. The weight scheduling threshold α controls the knowledge distillation to switch from ordinary to a self-paced decaying schedule. With the self-paced weight scheduling weight w_l, the knowledge transfer from the experts is automatically controlled by the student model's performance. The final knowledge distillation loss is the automatic weighted sum of knowledge distillation loss from all expert models:

$$L_{KD} = \sum_{l=1}^{L} w_l L_{KD_l}$$

4.3 Curriculum Instance Selection

Following the spirit of curriculum learning which mimics the human learning process, three questions need to be answered: 1) how to evaluate the difficulty of each instance, 2) how to *select* or *unselect* a sample, 3) how to design a curriculum so that samples are organized from easy to hard.

For the first question, we use the expert models' output confidence for each instance as an indication for instance difficulty. Given a training instance (x_i, y_i), suppose its ground-truth class y_i falls into the l-th subset S_l, i.e. $y_i \in S_l$, then we take the corresponding l-th expert model and use its output prediction for class y_i as confidence score, denoted as p_i. In this way, we can obtain the confidence score for all the instances in the training set.

For the second question, we adopt a soft selection method for instance selection. For instance (x_i, y_i), we replace the cross entropy loss with a weighted version:

$$L_{CE} = \sum_{i=1}^{N} v_i^{(k)} L_{CE}(x_i)$$

where $v_i^{(k)} \in [0, 1]$ is the selection weight at k-th epoch. A higher value of v_i (close to 1) indicates a soft selection of i-th instance, while a smaller value indicates a soft unselection of that instance.

Finally, to answer the third question, we design an automatic curriculum to determine the value of $v_i^{(k)}$, so that the instances are selected from easy to hard. The simplest approach is to sort the instances using their confidence score p_i obtained by the expert models. However, different from traditional curriculum

learning scenarios, our long-tailed classification problem involves both many-shot and low-shot categories, where low-shot instances tend to have lower confidence scores than many-shot instances. When sorted by the confidence score, the low-shot samples tend to be classified as hard examples and are not selected at first, which we do not wish to happen. To deal with such scenarios, instead of sorting across the whole training set, we sort instances according to their confidence scores **within each cardinality subset**. To be more specific, given the expert output confidence, v_i^k should be determined by three factors 1) the expert confidence p_i, 2) current epoch k, 3) the cardinality-adjacent subset S_l the i-th instance belongs to. Since the whole dataset is long-tailed, while we select samples from easy to hard, we also wish to select as uniform as possible across all subsets at the beginning of the training, and gradually add more hard samples as the epoch increases. In other words, at the first epoch we wish to select all the samples in the subset with lowest shots $\mathcal{S}_{min}$ (i.e. classes in $\mathcal{S}_{min}$ have the smallest number of samples) and same amount of samples in other subsets, and gradually add more samples until all the samples in all subsets are selected in the last epoch.

To achieve this goal, if we denote $N_{\mathcal{S}_l} = \frac{1}{|\mathcal{S}_l|}\sum_{i=1}^{|\mathcal{S}_l|} N_i$ as the average shot (average number of samples per class) in subset $\mathcal{S}_l$, then $v_i^{(k)}$ is determined by $p_i \frac{N_{\mathcal{S}_{min}}}{N_{\mathcal{S}_l}}$ at epoch 1, and grows gradually to 1 at the last epoch. Then at epoch 1, each subset softly selects its $N_{\mathcal{S}_{min}}$ easiest samples, and harder samples are gradually softly added to the training process. Formally, we use a monotonically increasing function f as scheduling function, so that $v_i^{(k)}$ will gradually grow from $p_i \frac{N_{\mathcal{S}_{min}}}{N_{\mathcal{S}_l}}$ to 1. For simplicity, we choose the linear function in the experiments and f is defined as

$$f(v_i^k) = (1 - v_i^{(1)})\frac{e}{E} + v_i^{(1)}$$

where $v_i^{(1)} = p_i \frac{N_{\mathcal{S}_l}}{N_{\mathcal{S}_{min}}}$ is the initial soft selection weight at epoch 1, and e, E are the current epoch and the total number of epochs respectively. It is worth noting that the scheduling function f can also be any convex or concave function as long as it is monotonically increasing within $[1, E]$. The impact of choosing different f is further analyzed in the experimental section. A schematic illustration of w_l and v_i can be found in Fig 4.

4.4 Training

With the Self-paced Expert Selection and Curriculum Instance Selection, we obtain the final loss function:

$$L = \sum_{i=1}^{N} v_i L_{CE}(x_i, y_i) + \sum_{l=1}^{L}\sum_{i=1}^{N} w_l L_{KD_l}(\mathcal{M}, \mathcal{M}_{Exp}; x_i)$$

where N, L are the number of training instances and number of experts respectively, and v_i, w_i controls the two levels of adaptive learning schedules. In

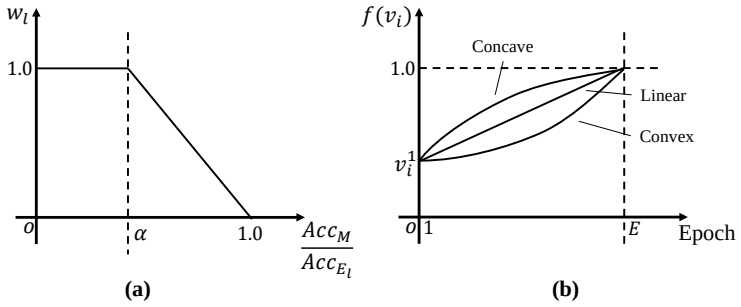

Fig. 4. Weight scheduling function for (a) model level selection w_l, and (b) instance level selection v_i.

practice, we first train expert models using ordinary Instance-level Random Sampling, where each instance is sampled with equal probability. We then train the whole LFME using Class-level Random Sampling adopted in [31, 25], where each class is sampled with equal number of samples and probability.

5 Experiments

5.1 Experimental Settings

Dataset We evaluate our proposed method on three benchmark long-tailed classification datasets: ImageNet-LT, Places-LT proposed in [31] and CIFAR100-LT proposed in [2]. ImageNet-LT is created by sampling a subset of ImageNet [6] following the Pareto distribution with power value $\alpha = 6$. It contains 1000 categories with class cardinality ranging from 5 to 1280. Places-LT is created similarly from Places dataset [52] and contains 365 categories with class cardinality ranging from 5 to 4980. CIFAR100-LT is created with exponential decay imbalance and controllable imbalance ratio.

Baselines For the first two datasets, similar to [31], our baseline methods include three re-weighting methods: Lifted Loss [35], Focal Loss [29] and Range Loss [51], and one SOTA few-shot learning method FSLwF [12], as well as the recent SOTA method OLTR [31]. For the CIFAR100-LT dataset, we mainly compare with the SOTA method LDAM proposed in [2].

Implementation Details For the first two datasets, we choose the number of cardinality-adjacent subsets $L = 3$ with thresholds $\{20, 100\}$ following the splits in [31]. We refer to these subsets as many, medium and few-shot subsets. For CIFAR100-LT, We equally split the 100 classes into two subsets: many and few-shot. We use PyTorch [38] to implement all experiments. For the first two datasets, we first train the experts using SGD for 120 epochs. Then the LFME model is trained with SGD with momentum 0.9, weight decay 0.0005 for 90 epochs, batch size 256, learning rate 0.1 and divide by 0.1 every 40 epochs. We use ResNet-10 [18] training from scratch for ImageNet-LT and ImageNet

pretrained Resnet-152 for Places-LT. During training, class-balanced sampling is adopted. For the CIFAR100-LT experiments, we first train the experts for 200 epochs and then train the LFME model using SGD with momentum 0.9, weight decay 2×10^{-4}, batch size 128, epochs 200, initial learning rate 0.1 and decay by 0.01 at 160 and 180 epochs, as well as deferred class-balanced sampling, same as [2]. The backbone network is ResNet-32. The distillation temperature T is set to 2, and the expert weight scheduling threshold α is set to 0.6 during the experiments.

Table 2. Long-tailed classification results on ImageNet-LT and Place-LT. * denotes reproduced results, other results are from [31].

Acc. / Method	ImageNet-LT				Places-LT			
	Many	Med.	Few	All	Many	Med.	Few	All
Plain Model	40.9	10.7	0.4	20.9	**45.9**	22.4	0.36	27.2
Lifted Loss[35]	35.8	30.4	17.9	30.8	41.1	35.4	24	35.2
Focal Loss[29]	36.4	29.9	16.0	30.5	41.1	34.8	22.4	34.6
Range Loss[51]	35.8	30.3	17.6	30.7	41.1	35.4	23.2	35.1
FSLwF [12]	40.9	22.1	15.0	28.4	43.9	29.9	**29.5**	34.9
OLTR [31]	43.2	35.1	18.5	35.6	44.7	37.0	25.3	35.9
OLTR [31]*	40.7	33.2	17.4	33.8	42.2	38.1	17.8	35.3
Ours	**47.1**	35.0	17.5	37.2	38.4	39.1	21.7	35.2
Ours + Focal Loss	46.7	35.8	17.3	37.3	37.0	**39.6**	23.0	35.2
Ours + OLTR	47.0	**37.9**	**19.2**	**38.8**	39.3	**39.6**	24.2	**36.2**

5.2 Main Results on Long-tailed Classification Benchmarks

Table 2 shows the long-tailed classification results on ImageNet-LT and Places-LT dataset. As can be found that our method is able to achieve superior or at least comparable results to the state-of-the-art methods such as OLTR. We found that many-shot categories benefit most from our LFME framework, while few-shot classes also demonstrate improvements and perform similarly with the re-weighting methods. Moreover, we also demonstrate that our method can be easily incorporated with other state-of-the-art methods, and we show the result of LFME+Focal Loss and LFME+OLTR (where LFME is added in the second stage of OLTR). We observe that both methods benefit from our expert model on all three subsets, and the combination of our method and OLTR outperforms previous methods by a large margin. It is also worth noting that our expert models are trained using vanilla CNNs, and utilizing other techniques will further lead to superior expert models, and assumably, superior student model.

To further demonstrate the statistical significance of the proposed method, we conduct experiments on CIFAR100-LT with imbalance ratio 100. The results in Table 3 show that LFME is able to achieve comparable performances with

the SOTA method LDAM [2] and combining them will further improve LDAM on both many and few-shot subsets.

Table 3. Results on CIFAR100-LT.

Methods	Many	Few	All
Plain CNN	59.0	18.2	38.6
Ours	59.0	25.5	42.3
LDAM [2]	58.8	26.1	42.4
Ours+LDAM	**59.5**	**28.0**	**43.8**

Table 4. Effect of different scheduling functions.

Schedule	Many	Medium	Few	All
Linear	47.1	**35.0**	**17.5**	**37.2**
Convex	47.2	34.6	16.7	36.9
Concave	**47.5**	34.7	17.0	37.1

5.3 Ablation Study

Effectiveness of Each Component

We evaluate each part of our method and the result is shown in Table 5. We compare with the following variants: 1) Instance-level Random Sampling (Ins.Samp.), where each instance is sampled with equal probability. 2) Instance-level Random Sampling + Ordinary Knowledge distillation (Ins.Samp.+KD), where non-self-paced version knowledge distillation from the experts is added, i.e. $w_l = 1.0$. 3) Class-level Random Sampling (Cls.Samp.), where each class is sampled with equal number of samples and probability. 4) Class-level Random Sampling + Ordinary Knowledge Distillation (Cls.Samp.+KD). 5) Class-level Random Sampling + Knowledge Distillation + Self-paced Expert Selection (Cls.Samp.+ KD + SpES). 6) Curriculum Instance Selection + Self-paced Expert Selection (CurIS+KD+SpES), which constitute our LFME framework.

From the results, we come up with the following observations: first, compared to the instance-random sampling, the adopted class-level random sampling is able to largely improve the few-shot performance while also decrease the many-shot performance slightly, since it samples more few-shot and less many-shot instances. Second, the introduction of knowledge distillation from experts can significantly improve the results, as it brings **11.3%** and **3.2%** to the Instance-level Sampling and Class-level Sampling baselines. However, while knowledge distillation brings improvements for many-shot classes, it will also decrease the few-shot accuracy slightly. Third, the Self-paced Expert Selection improves the knowledge distillation on all three subsets. It removes the performance ceiling brought from the experts and allows

Table 5. Effectiveness of each component.

Method \ Accuracy	Many	Med.	Few	All
Ins.Samp.	40.9	10.7	0.4	20.9
Ins.Samp.+KD	55.7	22.2	0.02	32.2
Cls.Samp.	38.8	32.3	17.0	32.6
Cls.Samp.+KD	44.9	34.5	15.8	35.8
Cls.Samp.+KD+SpES	46.6	35.8	16.5	37.1
CurIS+KD+SpES	47.1	35.0	17.5	37.2

the student to exceed the experts. As the results show that SpES brings **0.4%** and **1.3%** overall performance gain respectively. Finally, the proposed Curriculum Instance Selection further improves on the few-shot categories with **1.0%** in accuracy, so that the decrease on few-shot subset caused by the knowledge distillation is compensated.

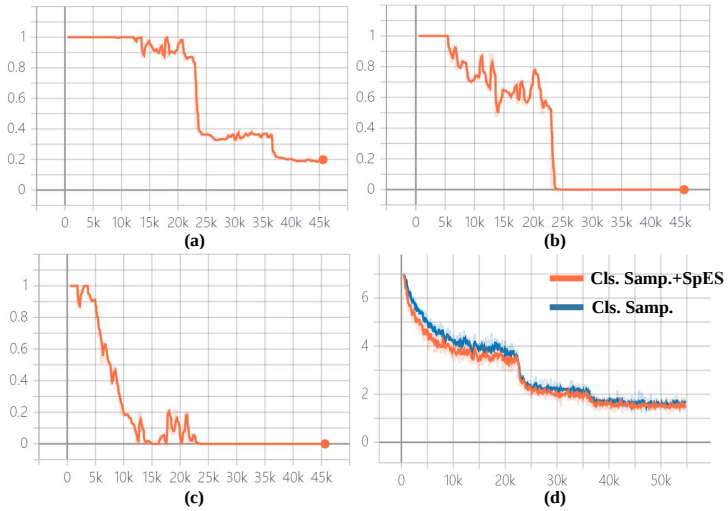

Fig. 5. (a)-(c): Visualization of self-paced expert selection scheduler w_l for many-shot, medium-shot, few-shot expert model. (d): Loss curves before and after adding Self-paced Expert Selection.

Visualization of Self-paced Expert Selection Self-paced expert selection plays an important role in LFME for more efficient and effective knowledge transfer. Fig 5 (a)-(c) gives a visualization of the expert selection weights w_l for many-shot, medium-shot, few-shot model. From the visualization, we observe that w_l serves to automatically control the knowledge transfer, as for many-shot and medium-shot experts the knowledge is consistently distilled, while for few-shot experts, the student instantly exceeds the expert's performance, thus leading to a decay in $w_{fewshot}$. Moreover, we also visualize the impact of Self-paced Expert Selection in terms of cross-entropy loss curves, shown in Fig 5 (d), and we find that it leads to a lower cross-entropy loss, which also verifies the effectiveness of the proposed Self-paced Expert Selection.

Effect of Learning Scheduler We also discuss the impact of different learning schedules for v_i as shown in Table 4. Given the initial instance confidence $v^{(1)}$, we test with the following scheduling functions:

- $f_{Linear} = (1 - v^{(1)})\frac{e}{E} + v^{(1)}$
- $f_{Convex} = 1 - (1 - v^{(1)})\cos(\frac{e}{E}\frac{\pi}{2})$
- $f_{Concave} = (1 - v^{(1)})\frac{\log(1+\frac{e}{E})}{\log 2}$

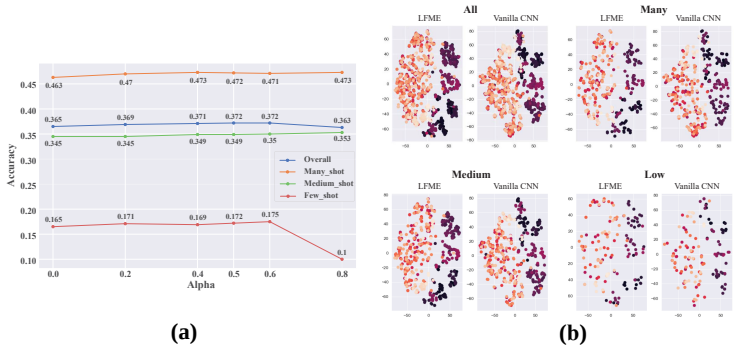

Fig. 6. (a) Sensitivity analysis of α. (b) T-SNE visualization of classification weights.

The result shows that the linear growing function yields the best result, while the concave the convex function f also produce similar performances. The convex function yields the worst performances as it selects fewest instances at the start of the training which may not be beneficial for the training dynamics.

Sensitivity Analysis of Hyperparameter α Fig. 6(a) shows the sensitivity analysis of expert weight scheduling threshold α. From the result, we observe that our model is robust to most α values. When α grows to 1.0, the Self-paced Expert Selection becomes ordinary knowledge distillation, and result in a performance decline.

Visualization of Classification Weights We visualize the classification weights of vanilla CNN and our LFME via T-SNE in Fig 6(b). The results show that our method results in a more structured, compact feature manifold. It shows that without particular re-weighting, our method is also able to produce discriminative feature space and classifiers.

6 Conclusions

In this paper, we propose a Learning From Multiple Experts framework for long-tailed classification problem. By introducing the idea of cardinality-adjacent subset which is less long-tailed, we train several expert models and propose two levels of adaptive learning to distill the knowledge from the expert models to a unified student model. From the extensive experiments and visualizations, we verify the effectiveness of our proposed method as well as each of its component, and show that the LFME framework is able to achieve state-of-the-art performances on the long-tailed classification benchmarks.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No.U1936202, No.61925107). We also thank anonymous reviewers for their constructive comments. Corresponding author: Guiguang Ding.

References

1. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th annual international conference on machine learning. pp. 41–48. ACM (2009)
2. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. arXiv preprint arXiv:1906.07413 (2019)
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
4. Collins, E., Rozanov, N., Zhang, B.: Evolutionary data measures: Understanding the difficulty of text classification tasks. arXiv preprint arXiv:1811.01910 (2018)
5. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9268–9277 (2019)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
7. Ding, G., Guo, Y., Chen, K., Chu, C., Han, J., Dai, Q.: Decode: Deep confidence network for robust image classification. IEEE Transactions on Image Processing **28**, 3752–3765 (2019)
8. Dong, Q., Gong, S., Zhu, X.: Class rectification hard mining for imbalanced deep learning. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1851–1860 (2017)
9. Drummond, C., Holte, R.C., et al.: C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: Workshop on learning from imbalanced datasets II. vol. 11, pp. 1–8. Citeseer (2003)
10. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1126–1135. JMLR. org (2017)
11. Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., Anandkumar, A.: Born-again neural networks. In: International Conference on Machine Learning. pp. 1602–1611 (2018)
12. Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4367–4375 (2018)
13. Graves, A., Bellemare, M.G., Menick, J., Munos, R., Kavukcuoglu, K.: Automated curriculum learning for neural networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1311–1320. JMLR. org (2017)
14. Guo, S., Huang, W., Zhang, H., Zhuang, C., Dong, D., Scott, M.R., Huang, D.: Curriculumnet: Weakly supervised learning from large-scale web images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 135–150 (2018)
15. Guo, Y., Ding, G., Han, J., Gao, Y.: Zero-shot learning with transferred samples. IEEE Transactions on Image Processing **26**, 3277–3290 (2017)
16. Han, H., Wang, W.Y., Mao, B.H.: Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: International conference on intelligent computing. pp. 878–887. Springer (2005)
17. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on knowledge and data engineering **21**(9), 1263–1284 (2009)

18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
19. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
20. Huang, C., Li, Y., Chen, C.L., Tang, X.: Deep imbalanced learning for face recognition and attribute prediction. IEEE transactions on pattern analysis and machine intelligence (2019)
21. Huang, Z., Wang, N.: Like what you like: Knowledge distill via neuron selectivity transfer. arXiv preprint arXiv:1707.01219 (2017)
22. Jeatrakul, P., Wong, K.W., Fung, C.C.: Classification of imbalanced data by combining the complementary neural network and smote algorithm. In: International Conference on Neural Information Processing. pp. 152–159. Springer (2010)
23. Jiang, L., Meng, D., Zhao, Q., Shan, S., Hauptmann, A.G.: Self-paced curriculum learning. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
24. Jiang, L., Zhou, Z., Leung, T., Li, L.J., Fei-Fei, L.: Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. arXiv preprint arXiv:1712.05055 (2017)
25. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. arXiv preprint arXiv:1910.09217 (2019)
26. Khan, S.H., Hayat, M., Bennamoun, M., Sohel, F.A., Togneri, R.: Cost-sensitive learning of deep feature representations from imbalanced data. IEEE transactions on neural networks and learning systems **29**(8), 3573–3587 (2017)
27. Li, D., Huang, J.B., Li, Y., Wang, S., Yang, M.H.: Weakly supervised object localization with progressive domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3512–3520 (2016)
28. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017)
29. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
30. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
31. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2537–2546 (2019)
32. Lopez-Paz, D., Bottou, L., Schölkopf, B., Vapnik, V.: Unifying distillation and privileged information. arXiv preprint arXiv:1511.03643 (2015)
33. Narvekar, S.: Curriculum learning in reinforcement learning. In: IJCAI. pp. 5195–5196 (2017)
34. Narvekar, S., Sinapov, J., Stone, P.: Autonomous task sequencing for customized curriculum design in reinforcement learning. In: IJCAI. pp. 2536–2542 (2017)
35. Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4004–4012 (2016)
36. Oksuz, K., Cam, B.C., Kalkan, S., Akbas, E.: Imbalance problems in object detection: A review. arXiv preprint arXiv:1909.00169 (2019)

37. Ouyang, W., Wang, X., Zhang, C., Yang, X.: Factors in finetuning deep model for object detection with long-tail distribution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 864–873 (2016)
38. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
39. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017)
40. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. arXiv preprint arXiv:1803.09050 (2018)
41. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014)
42. Stamatatos, E.: Author identification: Using text sampling to handle the class imbalance problem. Information Processing & Management **44**(2), 790–799 (2008)
43. Svetlik, M., Leonetti, M., Sinapov, J., Shah, R., Walker, N., Stone, P.: Automatic curriculum graph generation for reinforcement learning agents. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
44. Tahir, M.A., Kittler, J., Yan, F.: Inverse random under sampling for class imbalance problem and its application to multi-label classification. Pattern Recognition **45**(10), 3738–3750 (2012)
45. Wang, Y.X., Hebert, M.: Learning to learn: Model regression networks for easy small sample learning. In: European Conference on Computer Vision. pp. 616–634. Springer (2016)
46. Wang, Y.X., Ramanan, D., Hebert, M.: Learning to model the tail. In: Advances in Neural Information Processing Systems. pp. 7029–7039 (2017)
47. Xiang, L., Jin, X., Ding, G., Han, J., Li, L.: Incremental few-shot learning for pedestrian attribute recognition. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. pp. 3912–3918. AAAI Press (2019)
48. Xiang, L., Jin, X., Yi, L., Ding, G.: Adaptive region embedding for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 7314–7321 (2019)
49. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4133–4141 (2017)
50. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016)
51. Zhang, X., Fang, Z., Wen, Y., Li, Z., Qiao, Y.: Range loss for deep face recognition with long-tailed training data. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5409–5418 (2017)
52. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence **40**(6), 1452–1464 (2017)