

ACE: Ally Complementary Experts for Solving Long-Tailed Recognition in One-Shot

Jiarui Cai, Yizhou Wang, Jenq-Neng Hwang
University of Washington
Seattle, WA, USA

{jrcai, ywang26, hwang}@uw.edu

Abstract

One-stage long-tailed recognition methods improve the overall performance in a “seesaw” manner, i.e., either sacrifice the head’s accuracy for better tail classification or elevate the head’s accuracy even higher but ignore the tail. Existing algorithms bypass such trade-off by a multi-stage training process: pre-training on imbalanced set and fine-tuning on balanced set. Though achieving promising performance, not only are they sensitive to the generalizability of the pre-trained model, but also not easily integrated into other computer vision tasks like detection and segmentation, where pre-training of classifiers solely is not applicable. In this paper, we propose a one-stage long-tailed recognition scheme, ally complementary experts (ACE), where the expert is the most knowledgeable specialist in a subset that dominates its training, and is complementary to other experts in the less-seen categories without being disturbed by what it has never seen. We design a distribution-adaptive optimizer to adjust the learning pace of each expert to avoid over-fitting. Without special bells and whistles, the vanilla ACE outperforms the current one-stage SOTA method by 3~10% on CIFAR10-LT, CIFAR100-LT, ImageNet-LT and iNaturalist datasets. It is also shown to be the first one to break the “seesaw” trade-off by improving the accuracy of the majority and minority categories simultaneously in only one stage. Code and trained models are at <https://github.com/jrcai/ACE>.

1. Introduction

Object recognition is one of the most essential and substantial applications in computer vision. However, the performances of the state-of-the-art object recognition methods have limited capability on classifying real-world entities, which are skewed-distributed in a long-tailed manner naturally. Mostly driven by artificially-balanced datasets [4, 13], current models are dominated by the sample-rich

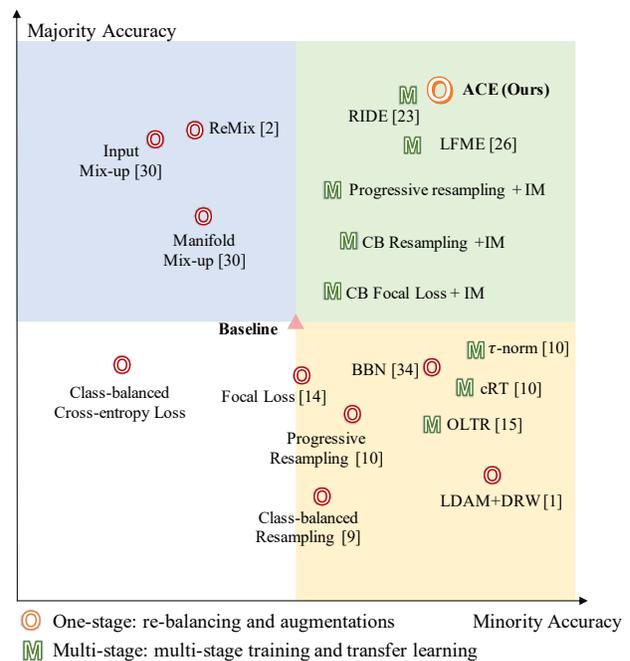


Figure 1. Performance of representative long-tailed recognition methods in terms of majority and minority classes compared to the baseline model (a ResNet). The results indicate most re-balancing methods improve the performance of minority categories by sacrificing that of the majority even with two-stage training (quadrant IV). Data augmentations are effective on the heads but slightly hurt the tails (quadrant II). The proposed ACE is the first one-stage SOTA method that improves the majority and minority simultaneously. Statistics for this figure are listed in the supplementary materials.

classes and lose sight of the tails when adapting to long-tailed sets. Facing up to the reality, scarce as the tail categories are, they are of the same or even higher significance than the heads in various fields, such as biological species identification [20], disease classification [21] and web-spam message detection [33]. This long-lasting bottleneck signif-

icantly restricts classification-related computer vision tasks into practical use, including detection [19, 25, 28] and instance segmentation [22, 29].

To ensure a well-accepted recognition capability over all categories, a tail-sensitive classifier becomes necessary. Existing solutions fall in three categories: one-stage [8, 24], two-stage with pre-training [10, 1], and multi-stage multi-expert frameworks [26, 23]. The one-stage algorithms follow a straightforward idea to address the imbalance of training set by re-balancing, including re-sampling [10] and re-weighting [1, 3, 31]. Despite the promotion of the tails, balancing techniques show an obvious “seesaw” phenomenon (Figure 1), that the accuracy of majority classes is sacrificed, indicating the under-representation of the heads. This raises a new concern that reducing the heads’ accuracy might lead to more serious consequences. Taking the animal identification system as an example, some species are much richer in population than the endangered ones. Increasing the recognition accuracy of the snow leopards has little chance to be verified as they are rarely seen; on the contrary, failing to precisely classify two bird kinds can easily result in a misunderstanding of the local ecology.

Literature in the recent years [10, 23, 26, 32] handles the issue in a roundabout way: firstly train the feature extractor (backbone) with the whole imbalanced set for generalizable representation learning, then re-adjust the classifier by re-sampled data or build diverse experts for various tasks in cascading stages. Further improving the performance as they are, however, the general idea still holds old wine in a new bottle by making new trade-offs. To re-balance the data distribution, heavily relying on the well-adjusted pre-trained model and re-balancing skills make the frameworks sensitive to hyper-parameters and hard to find a sweet point. More importantly, the accumulated training steps make the multi-stage models redundant and less practical to be integrated with other tasks simultaneously, *e.g.*, detection [22] and segmentation [29]. To guarantee the plug-in and play property, it is thus highly desirable to have a classifier that overcomes the long-tail challenge with only one stage.

The hankerings of overcoming current long-tail challenges make us look more profoundly to the human intelligence. When human-beings make hard classification choices, saying diagnosis of diseases, it is advantageous to involve specialists’ insights who are well-aware of their own fields. Moreover, for the rare diseases, panel discussion and consultation are indispensable to exclude interfering potentials. Similarly, in the long-tailed issue, we are inspired to design a group of experts with *complementary* skills: (1) they share elementary knowledge from the most diverse data source; (2) they are professional at splits of data respectively, and aware of what they do not specialize in; (3) opinions from the experienced experts (who see more data) are incorporated to complement the judgment from junior

experts (who see less) for optimal decision.

Following the idea, we propose the Ally Complementary Experts (ACE) for one-stage long-tailed recognition. ACE is a multi-expert structure where experts are trained in parallel with a shared backbone. The experts are assigned with diverse but overlapping imbalanced subsets, to benefit from specialization in the dominating part. We also introduce a distribution-adaptive optimizer that controls the update of each expert according to the volume of its training set. Finally, the outputs of all experts are re-scaled and aggregated by data splits. ACE is trained end-to-end without any pre-training or staged-training.

We evaluate ACE on various widely-used long-tailed datasets, including CIFAR10-LT, CIFAR100-LT [3], ImageNet-LT [15] and iNaturalist2018 [20] extensively with various experimental settings. Our method becomes the new SOTA among all one-stage long-tailed recognition methods with by 3-10% accuracy gain and is the first one that improves performance on all the three frequency groups (many-shot, medium-shot and few-shot). ACE also surpasses several multi-stage methods [10, 11, 15, 26] by a large margin.

2. Related Works

Methods for long-tailed recognition can be mainly grouped into three types: (1) readjustment of the data distribution; (2) two-stage training and transfer learning; (3) multi-expert/branch frameworks.

2.1. On the Data: Re-balancing and Augmentations

Re-balancing consists of under-sampling of the head classes, over-sampling of the tail classes and re-weighting of the loss function by the frequency or importance of the samples [9, 14, 3, 1]. Naive re-sampling in a class-balanced manner [8, 24] can easily overfit on the sample-few classes, either constructing a less imbalanced distribution by square-root sampling [17] or adjusting from instance-balanced to class-balanced sampling progressively [1, 3, 34] is a more stable and promising alternative.

Besides, strong data augmentations, which compensate for the insufficiency of data and improves the model’s generalizability, could increase the diversity of the training set. Mixup [30] along with its long-tailed variant re-balanced Mix-up (ReMix) [2]; and tail classes synthesis [32] are representative methods. However, the above algorithms commonly sacrifice the tails for the heads, or vice versa (Figure 1). The reason is the contradiction between representation learning and classifier learning, *i.e.*, instanced-based (bias) sampling learns the most generalizable representations while the unbiased classifier is less likely to overfit the re-sampled set.

Method	Data for Experts	Relationship of Experts	Number of Training Stages	Majority Gain	Minority Gain
LFME [26]	non-overlapping splits	independent	2	+	+
RIDE [23]	same full set	competing and complementary	3	++	+
ACE (Ours)	overlapping splits	supportive and complementary	1	+	++

Table 1. Comparisons between the proposed method with two SOTA multi-expert networks.

2.2. On the Representation: Two-stage Training and Transfer Learning

The methods in the second category migrate the learned knowledge from the heads to tails by two-stage training or memory-based transfer learning. Deferred re-balancing by re-sampling (DRS) and re-weighting (DRW) scheme [1] train the classifier layers with re-balancing after obtaining good representation on the imbalanced set at the first stage. Kang *et al.* [10] propose τ -norm and learnable weight scaling (LWS) to re-balance the decision boundaries of classifiers in the parameter domain. OLTR [15] and inflated episodic memory (IEM) [35] utilize memory banks for prototype learning and knowledge transfer among classes. However, the use of re-balancing can still hurt the accuracy of heads, and the inevitable extra memory consumption potentially limits the deployment on large-scale datasets.

2.3. Ensemble Methods: Multi-expert Networks

The recent trend on multi-expert or multi-branch networks shows the strong potential to address the long-tailed issue by treating the relatively balanced sub-groups separately. BBN [34], which assigns two branches with normal and reversed sampling, respectively, incooperates a cumulative learning strategy to adjust the bilateral training. BBN merges the two-stage methods into one, but still suffers from the same drawbacks of slight degradation of head’s accuracy. LFME [26] and RIDE [23] are multi-expert architectures that learn diverse classifiers in parallel, combining with knowledge distillation and distribution-aware expert selection. The main difference between our proposal and these two state-of-the-art methods are summarized in Table 1. Though achieving impressive performance, both of them suffer from extensive hyper-parameter tuning to balance the multiple optimization functions. More importantly, the multi-stage training requirement makes them difficult to be integrated into other tasks, like detection and segmentation.

3. Proposed Methodology

3.1. The ACE Framework

The architecture of the proposed Ally Complementary Experts (ACE) network is shown in Figure 2. Followed a

shared backbone, multiple experts are branched out with individual learnable blocks and a prediction layer. A distribution-aware planner distributes diverse but overlapping category splits for each expert, including target categories (TC) and interfering categories (IC). These experts complement each other from three aspects: (1) the dominating categories in their TCs are different, so that the predictions have their own strengths; (2) the TCs are overlapping, especially on sample-few categories, thus the predictions support each other; (3) the experts learn to suppress the output of IC so that they will not bring ambiguity in the categories that have never been seen. To further accommodate the disparity in data, a distribution-adaptive optimizer is designed to guide the experts to update at their own paces. We use classification loss L_{cls} and complement loss L_{com} to train the model end-to-end in only one-shot. Finally, the predictions from the experts are aggregated by averaging over the re-scaled logits in each data split.

3.2. Distribution-aware Planner

The experimental fact that classifiers tend to have better performance on the majority categories than the minority on an imbalanced set is considered a drawback and avoided by existing methods. However, if each split’s prediction is obtained from a classifier that biases on it, we could expect accuracy gains everywhere. Therefore, we design a distribution-aware planner to assign each expert with a subset of the training set, which is also imbalanced and dominated by different splits, respectively. Formally, the process is as follows,

Given a training set $\mathbb{D} = \{X; Y\}$, where X denotes the data and Y denotes the corresponding class labels, with C categories in total, for K experts $\mathbb{E} = \{\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_K\}$. Each \mathcal{E}_i is assigned subset categories \mathcal{C}_i , where C is assumed to be a multiple of K for simplicity of discussion and without loss of generality, $|\mathcal{C}_1 \cup \mathcal{C}_2 \cup \dots \cup \mathcal{C}_K| = C$ and $\mathcal{C}_i \cap \mathcal{C}_j \neq \emptyset$.

Similar to the spirit of re-balancing, sample-few categories should be more exposed. Therefore, the i -th expert

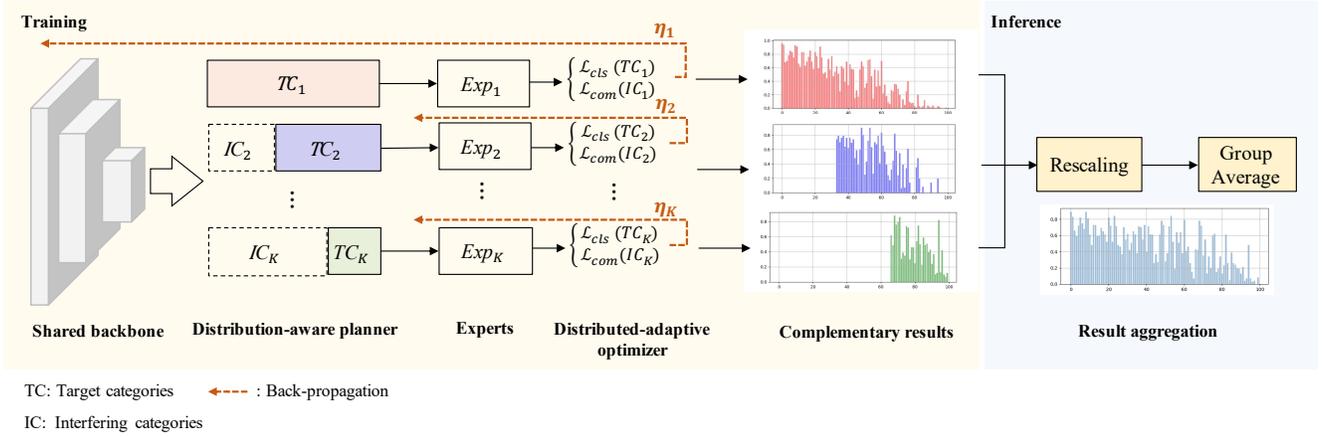


Figure 2. Network architecture of ACE. There are four components: (1) a shared backbone for representation learning; (2) a distribution-aware planner assigns diverse target categories (TC) and interfering categories to each expert, respectively; (3) a group of experts that learns to identify the TC with classification loss L_{cls} and eliminate their effect on IC with complementary loss L_{com} ; (4) a distribution-adaptive loss that adjust learning pace η of each expert for simultaneous convergence. By allying complementary experts (ACE) in a group average manner, the aggregated prediction compromises the merits of all experts.

\mathcal{E}_i is assigned target \mathcal{C}_i and interfering classes $\tilde{\mathcal{C}}_i$

$$\mathcal{C}_i = \left\{ \frac{C}{K}(i-1) + 1, \frac{C}{K}(i-1) + 2, \frac{C}{K}(i-1) + 3, \dots, C \right\},$$

$$\tilde{\mathcal{C}}_i = \left\{ 1, 2, \dots, \frac{C}{K}(i-1) \right\}. \quad (1)$$

For a randomly-sampled mini-batch of training data $\mathcal{B} \subset \mathbb{D}$, \mathcal{E}_i uses the corresponding sub-batch $\mathcal{B}_i = \{(x, y) : (x, y) \in \mathcal{B}, y \in \mathcal{C}_i\}$. In this case, there is always an expert be presented for training with all the samples, and the smaller the class, the more experts are presented. Besides, with this data split mechanism, the medium-shot or few-shot classes have chances to dominate an expert, thus eliminating the bias towards the sample-rich classes. The network degenerates to a plain classifier if $K = 1$.

Similar to the existing methods, we use the ResNet [7] as our backbone. The last residual block is duplicated for each expert, and followed by a learnable weight scaling (LWS) classifier [10]. The output logits (before SoftMax) of \mathcal{E}_i is $\mathbf{z}^i \in \mathbb{R}^{1 \times |C|}$, which are further adjusted to be $\hat{\mathbf{z}}_i$ by the norm of the fully-connected layers' weights to have comparable scales:

$$\hat{\mathbf{z}}_i = \frac{\|\mathbf{w}_i\|^2}{\|\mathbf{w}_1\|^2} \cdot \mathbf{z}_i \quad (2)$$

The set of experts that trained with class c is \mathcal{S}^c , then the output logit of class c is the average among the outputs from \mathcal{S}^c , i.e.,

$$\mathbf{o}^c = \frac{1}{|\mathcal{S}^c|} \sum_{\mathcal{E}_i \in \mathcal{S}^c} \hat{\mathbf{z}}_i \quad (3)$$

SoftMax operation is applied on \mathbf{o} to obtain the classification confidence.

3.3. Objective Functions

Loss functions are applied on each expert separately instead of on the aggregated output \mathbf{o} to avoid a mixture of expert-specific features. We use the cross-entropy loss as the classification loss, with the sub-batch \mathcal{B}_i for \mathcal{E}_i ,

$$L_{cls}^i(\mathcal{B}_i) = - \sum_{\mathcal{C}_i} y \log(\sigma(\mathbf{z}_i)), \quad (4)$$

where $\sigma(\cdot)$ represents the SoftMax operation.

In addition to classifying the assigned targeted class, each expert's response should not affect the other experts on the classes they have never seen, i.e., the interfering categories (IC). For the experts themselves, categories in IC are the main source of confusion as well. By eliminating the effect of IC, the experts work in a complementary manner rather than competitive. Hence, a regularization term to suppress the output of IC is necessary. We define the complement loss L_{com} as

$$L_{com}^i(\mathcal{B}_i) = \sum_{c_j \in \tilde{\mathcal{C}}_i} \|\mathbf{z}_i^{c_j}\|^2. \quad (5)$$

The complement loss, which serves as a soft regularization in the optimization process, minimizes the logits of non-target categories for \mathcal{E}_i so as to put down their effect. Detailed study of the impact of incorporating L_{com} could be found in Sec 4.3.

Overall, the loss function for \mathcal{E}_i is

$$L_{\mathcal{E}_i}(\mathcal{B}_i) = L_{cls}^i + L_{com}^i. \quad (6)$$

3.4. Distributed-adaptive Optimizer

Recall the Linear Scaling Rule [6] for training networks in mini-batches with stochastic gradient descent (SGD) optimizer: *when the minibatch size is multiplied by k , multiply the learning rate by k . All other hyper-parameters (weight decay, momentum, etc.) are kept unchanged.*

By this rule, to avoid over-fitting, the optimizer should be distribution-aware to assign smaller weights to \mathcal{E}_i which is trained with less data. Denoted the base learning rate as η_0 , which is the learning rate for the expert presented with all categories, the i -th expert is trained by,

$$\eta_i = \eta_0 \cdot \frac{\sum_{c \in \mathcal{C}_i} n_c}{\sum_c n_j}, \quad (7)$$

where $\mathcal{N} = \{n_1, n_2, \dots, n_C\}$ are the number of samples in each class and \mathcal{N} is assumed in the descending order.

The loss of \mathcal{E}_1 updates the backbone and parameters of \mathcal{E}_1 , and L_i that $i > 1$ only updates the expert itself. The reason is the errors likely duplicates because of data overlapping, which means the backbone could be corrected multiple times due to the same error. This is similar to the idea of re-weighting methods, as introduced in Section 2.1 that hurt the representation learning. Therefore, only \mathcal{E}_1 updates the backbone.

4. Experiments

4.1. Datasets and Protocols

Generally, in long-tail recognition tasks, the classes are categorized into many (with more than 100 training samples), medium (with 20 ~100 samples) and few (with less than 20 samples) splits [15]. The *imbalance factors* (IFs) of the long-tailed datasets, defined as the frequency of the largest class divided by the smallest class, vary from 10 to over 500 [3, 15, 20].

CIFAR100-LT and CIFAR10-LT [3] are artificially created from the balanced CIFAR dataset [13] by reducing training samples according to an exponential function $n = n_i \mu^i$, where i is the class index, n_i is the original number of samples and $\mu \in (0, 1)$. We experiment with two commonly used IFs, 100 and 50. There are approximately 10K~13K training images and 10K testing images for each split. ResNet-32 is used as the base network, where the last residual block is tripled for the branches to be comparable with other methods. Following [7], for training samples, 4 pixels are padded on each side, following by a 32×32 random crop on the padded image or its horizontal flip. The network is trained by the stochastic gradient descent (SGD) optimizer with a momentum 0.9 for 400 epochs. The base

learning rate is 0.1 and decreases by 0.1 at epoch 320 and 360, respectively. Mixup [30] augmentation is used with α as 0.3, and the last 20 epochs are trained without Mixup.

ImageNet-LT [15] is sampled from ImageNet-2012 [4] following the Pareto distribution with the power value $\alpha = 6$. ImageNet-LT contains 115.8K images for 1000 categories, with a maximum of 1280 images per class and a minimum of 5 images per class. Following [15, 10, 26, 32], we use ResNet-10 as the backbone. To be comparable with [10, 23], we also report our results with ResNet-50 and ResNeXt-50 [27]. For data pre-processing, the training samples are resized to 256×256 , then randomly cropped to 224×224 and flipped horizontally with a probability of 0.5; on testing, the aspect ratio of the testing sample is kept by first resizing proportionally its shorter side to 256 then crop 224×224 in the center. The networks are trained by the SGD optimizer with momentum 0.9 for 100 epochs. The base learning rate is 0.1 and decreases by 0.1 at epoch 120 and 160. Mixup augmentation is used with α as 0.3, and the last 20 epochs are trained without Mixup.

iNaturalist2018 [20] is a real-world large-scale dataset for species identification of animals and plants. Following the literature, we use the 2018 version which contains 438K images for over 8K categories, with extremely imbalanced distribution (IF=512) and challenging fine-grained issues. We use ResNet-50 as the backbone, and the same pre-processing and training protocol as ImageNet-LT. Mixup augmentation is used with α as 0.3, and the last 20 epochs are trained without Mixup.

4.2. Performance

Competing methods. Generally, there are two types of the competing methods by whether or not there is a backbone pre-training stage. For one-stage type of methods, re-balancing of the long-tailed dataset is either by resampling, (*e.g.*, class-balanced and progressively-balanced [10]), or reweighting (*e.g.*, focal loss [14], class-balanced focal loss [3], and LDAM [1]). Besides, strong augmentation tricks (*e.g.*, mixup [30], re-balanced mixup [2], tail sample synthesis using class activation maps (CAM) [32])) can also benefit the overall accuracy, especially the heads. Moreover, transfer learning in either image domain (major-to-minor translation [11]) and in feature domain (OLTR [15]) are proved useful. Logit Adjustment [16] encourages a large relative margin between logits of rare versus dominant labels with a one-stage loss. BBN [34] uses a two-branch architecture to combine normal sampling and distribution-reversed sampling progressively, improving the tail’s accuracy in a large margin. The other type is two-stage methods. In the second stage, τ -norm, LWS and cRT [10] retrain or fine-tune the classifier with a balanced dataset or unbiased classifier weights. LFME [26] and RIDE [23] are multi-branch assembled architectures

with knowledge distillation. LFME uses a teacher-student network to train experts on many-/medium-/few-shot splits, while RIDE does not fix the number of branches and uses KL-divergence loss to force them to be experts on different groups.

CIFAR-LT Table 2 shows the proposed ACE performs the best among all one-stage methods and surpasses other multi-stage methods on CIFAR100-LT-100. We outperform the previous one-stage SOTA BBN by 9.0%. Class-wise accuracy gain comparison with representative one-stage long-tailed recognition methods is shown in Figure 3. ACE has significant advantages in medium and few-shot categories. It is also the only method that improves all the groups by a single stage. Table 4 shows the top-1 accuracy on CIFAR10-LT and CIFAR100-LT with imbalance factor 50 and 100.

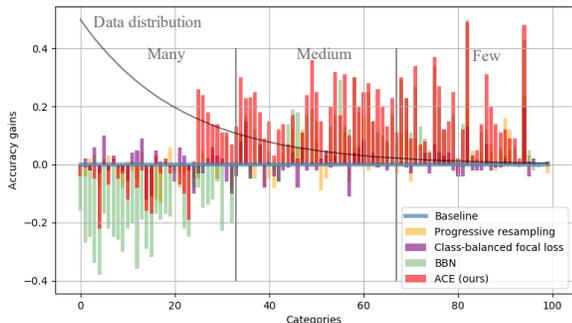


Figure 3. Accuracy gain comparisons between representative one-stage long-tailed recognition methods and baseline. While other methods decrease the majority accuracy, our ACE improves the many, medium and few groups all together.

ImageNet-LT and iNaturalist We also report our performance on ImageNet-LT with various backbone models of ResNet-10, ResNet-50 and ResNeXt-50 as well as iNaturalist-LT with ResNet-50, shown in Table 3. Our method outperforms the BBN by 6.4% (ResNet-50) and 7.3% (ResNeXt-50) on ImageNet-LT and 3.9% on iNaturalist2018, respectively.

4.3. How ACE Works

Complementary of experts. We compare ACE with its two variants to show the effectiveness of its architecture, learning process and the loss function: one is training without L_{com} and the other is the non-complementary architecture, which is called split-specific classifier (SSC). In the latter one, output dimensions of the classifier of \mathcal{E}_i are the same as $|\mathcal{C}_i|$, i.e., $\mathbf{z}^i \in \mathbb{R}^{1 \times |\mathcal{C}_i|}$. In other words, the weights of non-target groups are set to be zero as a hard constraint, instead of learning to suppress them with L_{com} in a soft

regularization manner. Results are in Table 5. Figure 4 shows ACE with L_{com} in the top row, where \mathcal{E}_i learns similar scales over all the data splits and the scales of interfering classes are zeros. Therefore, all trained experts have an approximately equal contribution to the shared splits. We also observe that on the minority splits j of \mathcal{E}_i generates supportive results for \mathcal{E}_j (e.g., \mathcal{E}_1 is peripheral in the few-shot split, and its scales are smaller than those of \mathcal{E}_3 , so it is just a supplementary to \mathcal{E}_3 's output.). As seen from the middle row of Figure 4, by splitting the data to complementary batches, but without L_{com} , all experts compete with each other in the common splits. For example, \mathcal{E}_1 is strong over all categories though it is less accurate in the minority classes compared to \mathcal{E}_3 , \mathcal{E}_1 's scales are still larger than \mathcal{E}_3 's. This explains why ACE without L_{com} has the best performance in the head categories. In the experiments on SSC, where the experts learn to classify \mathcal{C}_i but cannot distinguish the untrained categories, resulting in the obvious dominance of the expert trained with the full set in all splits, making other experts useless.

Results here are inspiring: different from most exiting works that try to eliminate the bias, we utilize it. The data re-balancing is embedded in the data assignment to ensure more exposure of the minority. The individual back-propagation of each expert will not hurt the representation learning. Therefore, L_{com} decouples the representation learning and classifier training in one stage.

Effectiveness of distribution-aware optimizer. The distribution-aware optimizer controls the learning speed of each expert with various data assignments. In this section, we compare the linear scaling rule with the square-root scaling [12] and a uniform optimizer. [12] indicates when multiplying the batch size by S , one should multiply the learning rate by \sqrt{S} to keep the variance in the gradient expectation constant. For a uniform optimizer, all the experts share the same η , i.e.,

$$\eta_i^{sqr} = \eta_0 \cdot \sqrt{\frac{\sum_{c \in \mathcal{C}_i} n_c}{\sum_c n_j}}, \eta_i^{uni} = \eta_0 \quad (8)$$

The training will be more sensitive to the variance of data with a larger learning rate. For the experts trained by minority splits, we have $\eta_i^{uni} \gg \eta_i^{sqr} > \eta_i^{linear}$. The comparison of the results is shown in Table 6. All three schemes produce better results than baseline. η_i^{uni} promotes the higher improvements in the majority categories, while significantly decreases the tails. The reason is several experts converge too early and thus not effective due to over-fitting. η_i^{sqr} and η_i^{linear} show similar performance, while η_i^{linear} is better in medium and few-shot splits. By comparing η_i^{sqr} and η_i^{linear} , we observe learning rate is not the principal reason for accuracy booms. We conclude that selecting a proper optimization scheme with respect to

Type	Method	Multi-experts	Accuracy			
			All	Many	Medium	Few
One-Stage	Baseline (ResNet-32)		38.3	65.2	37.1	9.1
	CB resampling [9]§		36.0 (-1.7)	59.0 (-6.2)	35.4 (-1.7)	10.9 (+1.8)
	Focal loss [14]		37.4 (-0.9)	64.3 (-0.9)	37.4 (+0.3)	7.1 (-2.0)
	CB Focal loss [3]§		38.7 (+0.4)	65.0 (-0.2)	37.6 (+0.5)	10.3 (+1.2)
	Progressive [10]		39.4 (+1.1)	63.3 (-1.9)	38.8 (+1.7)	13.1 (+4.0)
	ReMix [2]		40.9 (+2.6)	69.6 (+4.4)	40.7 (+3.0)	8.8 (-0.3)
	Mixup [30]		41.2 (+2.9)	70.7 (+5.5)	40.4 (+3.3)	8.8 (-0.3)
	BBN [34]	✓	39.4 (+1.1)	47.2 (-18.0)	49.4 (+12.3)	19.8 (+10.7)
	Logit Adjustment [16]		43.9 (+5.6)	-	-	-
	ACE (3 experts)	✓	49.4 (+11.1)	66.1 (+0.9)	55.7 (+18.6)	23.5 (+14.4)
ACE (4 experts)	✓	49.6 (+11.3)	66.3 (+1.1)	52.8 (+15.7)	27.2 (+18.1)	
Multi-Stage	τ -norm [10]		43.2	65.7	43.6	17.3
	cRT [10]		43.3	64.0	44.8	18.1
	LDAM+DRW [1]		42.0	61.5	41.7	20.2
	LDAM+LFME [26]	✓	43.8	-	-	-
	LDAM+M2m [11]		43.5	-	-	-
	CAM [32]	✓	47.8	-	-	-
	RIDE [23] (2 experts)	✓	47.0	67.9	48.4	21.8
	RIDE [23] (3 experts)	✓	48.0	68.1	49.2	23.9
	RIDE [23] (4 experts)	✓	49.1	69.3	49.3	26.0

Table 2. Top-1 accuracy on CIFAR100-LT-100. (-) shows comparison to the baseline, where **increase** and **decrease** are represented by color. Our ACE is the only one-stage method with performance gain on all groups and of the best over all categories. §: CB represents class-balanced.

Method	ImageNet-LT			iNaturalist
	Res10	Res50	ResX50	Res50
Baseline	20.9	41.6	44.4	66.1
FSLwF [5]	28.4	-	-	-
Range Loss [31]	30.7	-	-	-
Lifted Loss [18]	30.8	-	-	-
Focal loss [14]	30.5	-	-	60.3
CB Focal loss [3]	-	-	-	61.1
BBN [34]	-	48.3	49.3	68.0
Logit Adj.[16]	-	51.1	-	66.4
ACE (3 experts)	44.0	54.7	56.6	72.9
OLTR [15]	34.1	-	46.3	63.9
NCM [10]	35.5	44.3	47.3	-
LDAM+DRW [1]	36.0	-	-	68.0
cRT [10]	41.8	47.3	49.5	65.2
τ -norm [10]	40.6	46.7	49.4	65.6
LWS [10]	41.4	47.7	49.9	65.9
CAM [32]	43.1	-	-	70.9
LFME [26]	38.8	-	-	-
RIDE [23]†	-	54.4	55.9	71.4
RIDE [23]‡	-	54.9	56.4	72.2

Table 3. Top-1 accuracy on ImageNetLT and iNaturalist2018. Detailed results on each group are listed in the supplementary materials. Overall, it shows the multi-expert/branch architecture outperforms the re-balancing methods. Our ACE has consistent performance gain comparing with other one-stage methods with multiple backbones, and is comparable with multi-stage methods. †:2 experts, ‡:3 experts.

Method	CIFAR100-LT		CIFAR10-LT	
	100	50	100	50
Baseline	38.3	42.1	69.8	75.2
Focal loss [14]	37.4	42.4	70.4	75.3
Mixup [30]	39.5	45.0	73.1	77.8
CB Focal loss [3]	38.7	46.2	74.6	79.3
BBN [34]	39.4	47.0	79.8	82.2
Logit Adj.[16]	43.9	-	77.7	-
ACE (3 experts)	49.4	50.7	81.2	84.3
ACE (4 experts)	49.6	51.9	81.4	84.9
LDAM+DRW [1]	42.0	45.1	77.0	79.3
LFME [26]	42.3	-	-	-
LDAM+M2m [11]	43.5	-	79.1	-
CAM [32]	47.8	51.7	80.0	83.6
RIDE [23]	49.1	-	-	-

Table 4. Top-1 accuracy on CIFAR100-LT and CIFAR10-LT with imbalance factor 100 and 50.

the data distribution can effectively benefit the overall performance.

Effectiveness of group average output aggregation.

We compare different aggregation methods of the output logits $\{z^i\}$ from K experts. Four variants of the aggregation methods are shown in Figure 5 (3 experts).

Comparisons between (ACE) and (4) in Table 7 shows that the design of scaling is for preserving the accuracy of the head classes. Computing the maximum by groups over

Methods	All	Many	Medium	Few
ACE (With L_{com})	49.4	66.1	55.7	23.5
Expert 1	41.9	71.2	40.2	10.7
Expert 2	30.7	19.9	53.7	17.7
Expert 3	21.8	0.0	38.7	27.8
Without L_{com}	47.2	71.5	49.4	17.5
Expert 1	42.0	71.0	40.9	10.5
Expert 2	31.1	19.4	53.8	19.4
Expert 3	22.0	0.0	38.8	28.3
With SSC	43.4	65.1	44.4	18.0
Expert 1	41.6	68.2	41.2	12.1
Expert 2	16.0	2.4	26.5	19.9
Expert 3	21.4	0.0	38.6	26.7

Table 5. Overall and many-/medium-/few-shot split top-1 accuracy on CIFAR100-LT-100 of the three model. The results are consistent with our analysis that without complementary loss, the experts are competing, so the results tend to average. Split-specific classifiers (SSC) depends mostly on \mathcal{E}_1 .

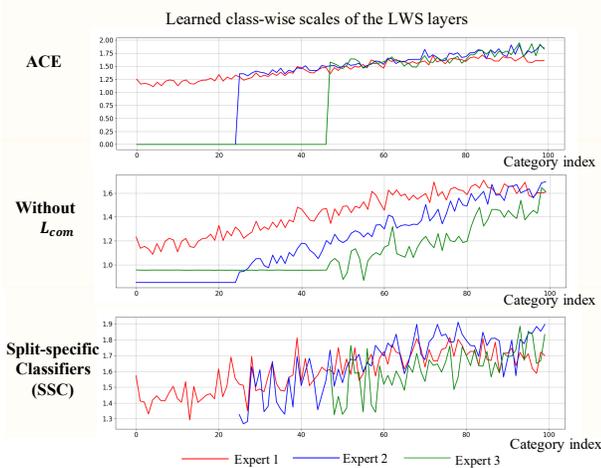


Figure 4. The classifier learnt scales of three model: ACE trained with complementary loss (top), ACE trained without complementary loss (middle), and split-specific classifiers (SSC) (bottom) trained on CIFAR100-LT-100. \mathcal{E}_1 , \mathcal{E}_2 and \mathcal{E}_3 are plotted in red, blue and green colors, respectively. Complementary loss allows the experts work jointly in their common splits. Without the complementary loss, the experts trained with full batch has the largest scales on all splits and competes with the real dominating experts.

Scheme	All	Many	Medium	Few
Linear	49.4	66.1	55.7	23.5
Square-root	49.1	67.1	55.2	22.1
Uniform	41.7	69.7	39.9	10.7

Table 6. Comparisons of learning rate scaling schemes on CIFAR100-LT-100.

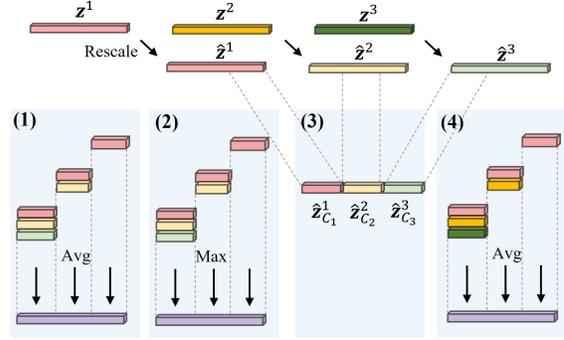


Figure 5. Illustration of variants of output aggregation methods.

the scaled logits (2) also suppresses the performance on the heads, as the experts for small classes are easier to overfit and thus overconfident. Concatenating the result of each dominating group of the experts amplifies the drawbacks of overconfidence, and experts competes each other. Overall, merging multiple experts is a trade-off for one-stage methods, in which all experts are trained from scratch. On the other hand, our ACE balances them by adjusting learning speed and with complementary loss, achieving improvements for all groups.

Aggregation	All	Many	Medium	Few
Group Avg w/ scaling (ACE)	49.4	66.1	55.7	23.5
Group Max (2)	43.4	47.5	54.2	26.5
Group Concat (3)	37.7	30.3	50.2	22.9
Group Avg w/o scaling (4)	46.7	49.5	53.0	36.5

Table 7. Ablation study on aggregation of the outputs on CIFAR100-LT-100.

5. Conclusion

In this paper, extensive experiments on existing long-tailed recognition algorithms reveal the contradiction between biased representation learning and unbiased classifier learning. We proposed a multi-expert network that optimizes the two in a uniform network. Complementary constraints in data and objective function are applied to suppress the effects of non-targeted groups and promote both of the dominating and minority groups. Besides, a distribution-adaptive optimization scheme helps to adjust the learning paces of each expert to avoid over-fitting. ACE becomes the new SOTA among all one-stage long-tailed recognition methods with 3~10% accuracy gain, and is the first one that improves performance on all three frequency splits. With the equivalent strong performance to the multi-stage methods, there is great potential to extend well-formulated one-stage ACE to complex computer vision tasks like detection and segmentation.

References

- [1] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arachiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *arXiv preprint arXiv:1906.07413*, 2019. [2](#), [3](#), [5](#), [7](#)
- [2] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: Rebalanced mixup. In *European Conference on Computer Vision*, pages 95–110. Springer, 2020. [2](#), [5](#), [7](#)
- [3] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019. [2](#), [5](#), [7](#)
- [4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. [1](#), [5](#)
- [5] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. [7](#)
- [6] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large mini-batch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. [5](#)
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [4](#), [5](#)
- [8] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. [2](#)
- [9] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002. [2](#), [7](#)
- [10] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019. [2](#), [3](#), [4](#), [5](#), [7](#)
- [11] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13896–13905, 2020. [2](#), [5](#), [7](#)
- [12] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014. [6](#)
- [13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. [1](#), [5](#)
- [14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. [2](#), [5](#), [7](#)
- [15] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. [2](#), [3](#), [5](#), [7](#)
- [16] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020. [5](#), [7](#)
- [17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, 2013. [2](#)
- [18] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4004–4012, 2016. [7](#)
- [19] Jingru Tan, Xin Lu, Gang Zhang, Changqing Yin, and Quanquan Li. Equalization loss v2: A new gradient balance approach for long-tailed object detection. *arXiv preprint arXiv:2012.08548*, 2020. [2](#)
- [20] Grant Van Horn, Oisín 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. [1](#), [2](#), [5](#)
- [21] Chen Wang, Chengyuan Deng, and Suzhen Wang. Imbalance-xgboost: leveraging weighted and focal losses for binary label-imbalanced classification with xgboost. *Pattern Recognition Letters*, 136:190–197, 2020. [1](#)
- [22] 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. In *European conference on computer vision*, pages 728–744. Springer, 2020. [2](#)
- [23] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X Yu. Long-tailed recognition by routing diverse distribution-aware experts. *arXiv preprint arXiv:2010.01809*, 2020. [2](#), [3](#), [5](#), [7](#)
- [24] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 7032–7042, 2017. [2](#)
- [25] Jialian Wu, Liangchen Song, Tiancai Wang, Qian Zhang, and Junsong Yuan. Forest r-cnn: Large-vocabulary long-tailed object detection and instance segmentation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1570–1578, 2020. [2](#)
- [26] Liuyu Xiang, Guiguang Ding, and Jungong Han. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *European Conference on Computer Vision*, pages 247–263. Springer, 2020. [2](#), [3](#), [5](#), [7](#)
- [27] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on*

- computer vision and pattern recognition*, pages 1492–1500, 2017. [5](#)
- [28] Weiping Yu, Taojiannan Yang, and Chen Chen. Towards resolving the challenge of long-tail distribution in uav images for object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3258–3267, 2021. [2](#)
- [29] Yuhang Zang, Chen Huang, and Chen Change Loy. Fasa: Feature augmentation and sampling adaptation for long-tailed instance segmentation. *arXiv preprint arXiv:2102.12867*, 2021. [2](#)
- [30] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. [2](#), [5](#), [7](#)
- [31] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5409–5418, 2017. [2](#), [7](#)
- [32] Yongshun Zhang, Xiu-Shen Wei, Boyan Zhou, and Jianxin Wu. Bag of tricks for long-tailed visual recognition with deep convolutional neural networks. 2021. [2](#), [5](#), [7](#)
- [33] Chensu Zhao, Yang Xin, Xuefeng Li, Yixian Yang, and Yuling Chen. A heterogeneous ensemble learning framework for spam detection in social networks with imbalanced data. *Applied Sciences*, 10(3):936, 2020. [1](#)
- [34] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9719–9728, 2020. [2](#), [3](#), [5](#), [7](#)
- [35] Linchao Zhu and Yi Yang. Inflated episodic memory with region self-attention for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4344–4353, 2020. [3](#)