Long-tailed Recognition by Routing Diverse Distribution-Aware Experts

ICLR 2021 Spotlight

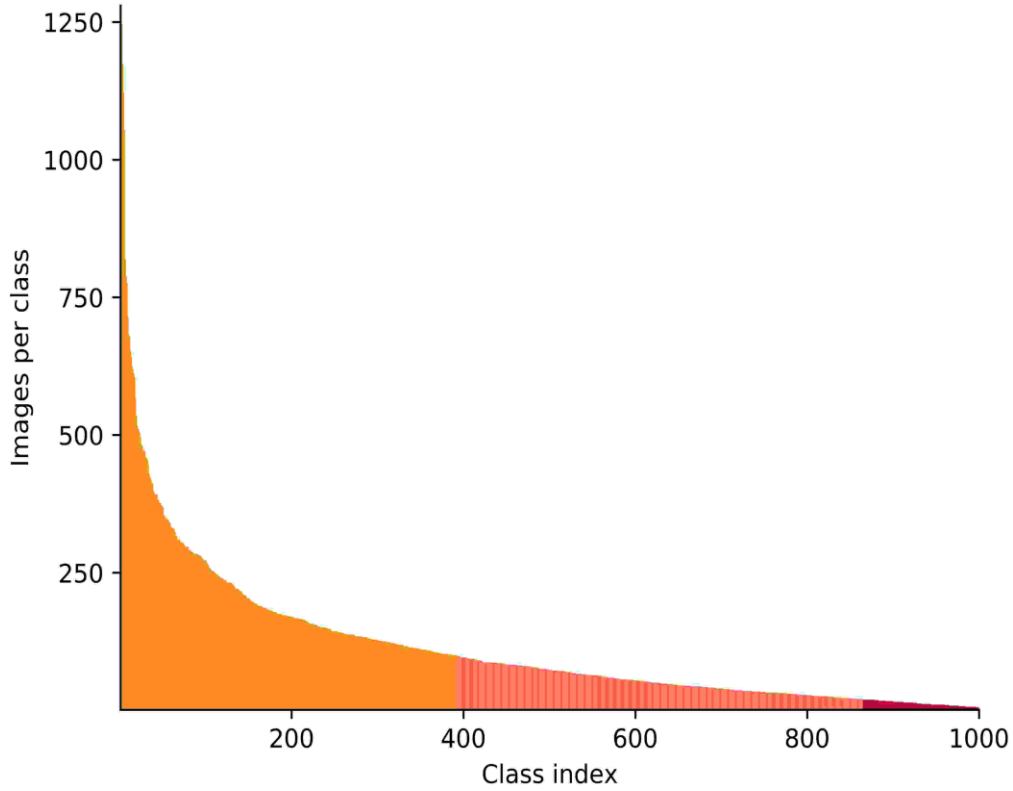
박성현

DAVIAN

Data and Visual Analytics Lab

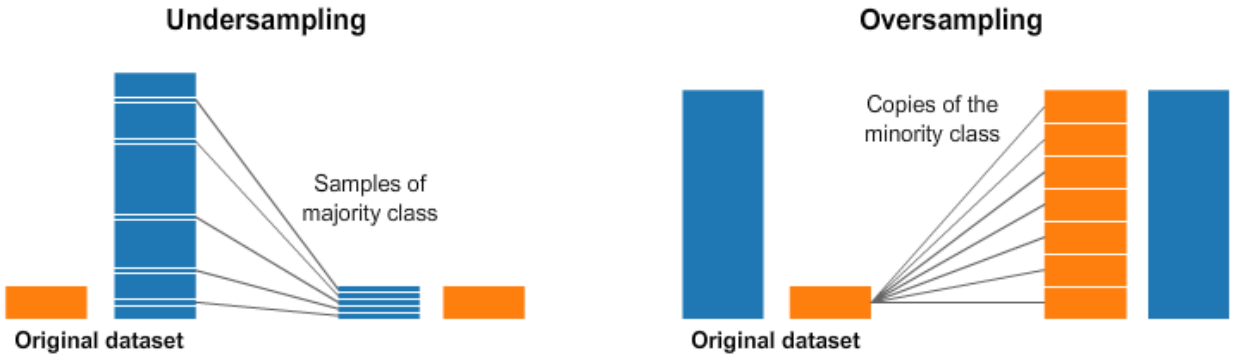
Long-tailed Classification

- ***Real-world datasets*** usually suffer from its expensive data acquisition process and the labeling cost. This commonly leads a dataset to have a "***long-tailed***" label distribution.
- Such class-imbalanced datasets make the standard training of DNN harder to generalize, particularly if one requires a ***class-balanced performance*** metric for a practical reason.

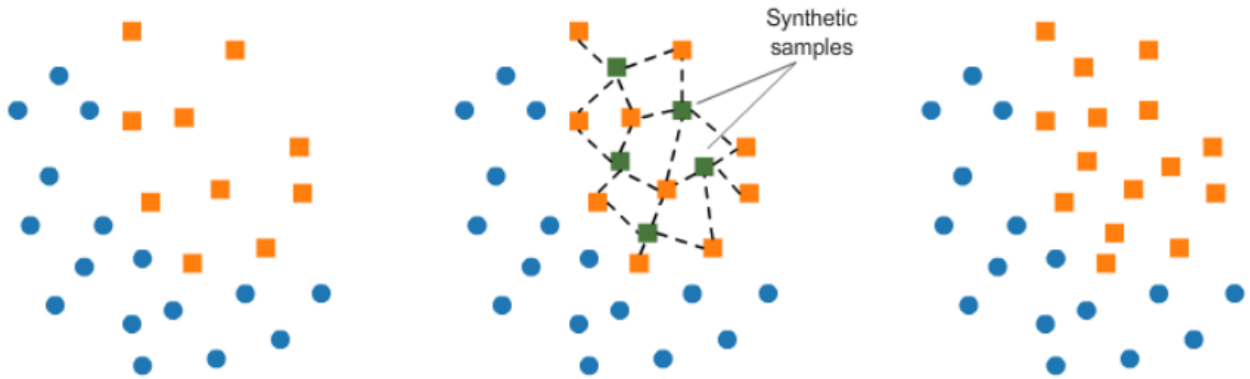


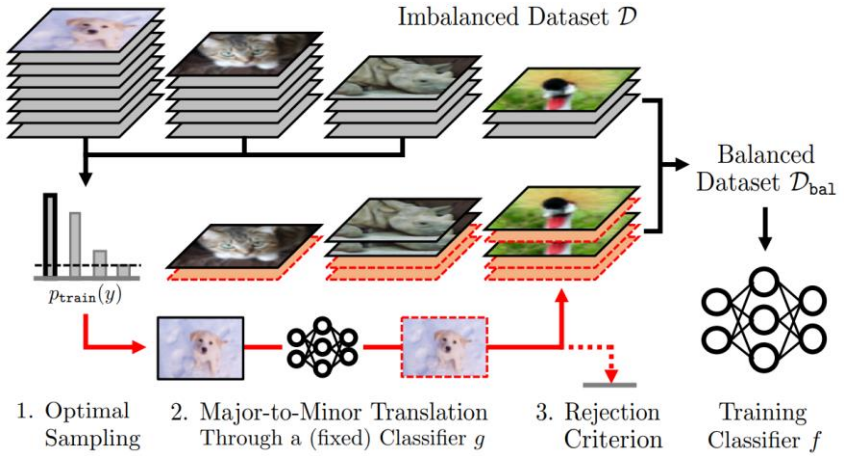
Long-tailed Classification: Re-sampling

- ***Re-sampling*** the dataset so that the expected sampling distribution during training can be balanced.



- ***Synthesizing*** elements for the minority class (SMOTE, M2m)



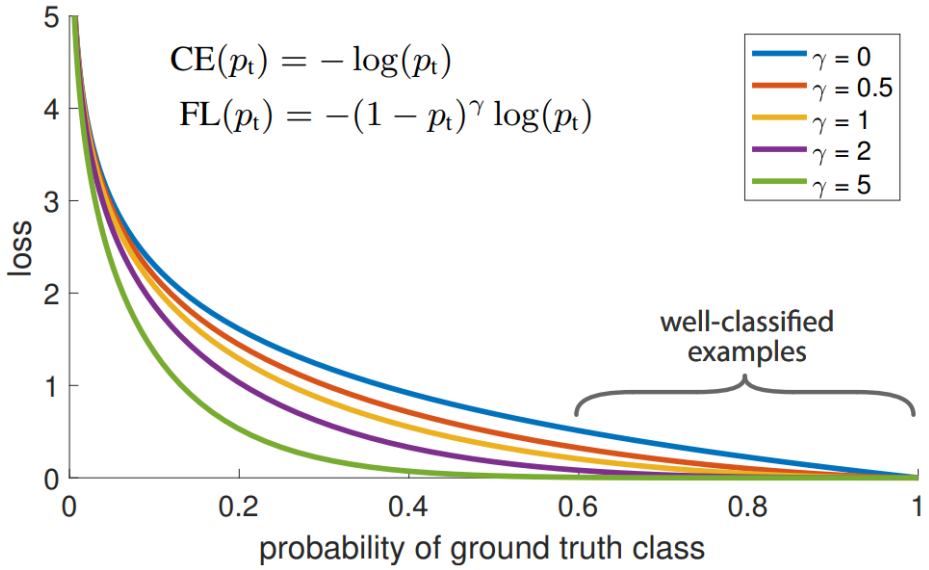


Kim et al, "M2m: Imbalanced Classification via Major-to-minor Translation", CVPR 2020

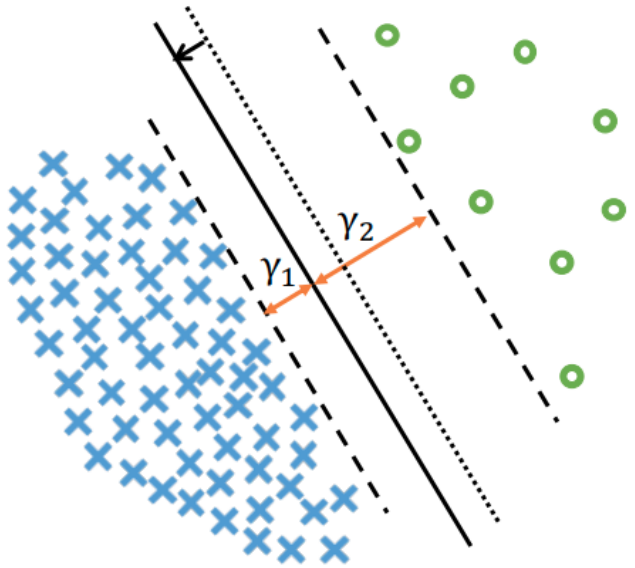
Long-tailed Classification: Re-weighting

- ***Re-weighting*** the loss function by a factor inversely proportional to the sample frequency in a class-wise manner.
- E.g., Range loss, Focal loss, Class-balanced loss, Label distribution-aware loss

- Focal Loss



- LDAM Loss



Tsung-Yi Lin et al, "Focal Loss for Dense Object Detection", ICCV 2017

Motivation

- Analyze the long-tail classifier's performance change in terms of ***bias and variance analysis***.
- The prediction error of model h on instance x with output Y varies with the training data D.
- The ***model bias*** measures the ***accuracy*** of the prediction with respect to the true value, the ***variance*** of the method measures the ***stability*** of the prediction, and the ***irreducible error*** measures the ***precision*** of the prediction and is ***irrelevant*** to the model h.

$$\text{Error}(x; h) = E[(h(x; D) - Y)^2] = \text{Bias}(h)^2 + \text{Variance}(h) + \text{irreducible error}.$$

Motivation

1. ***Mean accuracy*** : All the long-tail methods increase the overall, medium-shot, and few-shot accuracies, but these previous methods all decrease the many-shot accuracy. Our method increases accuracies on all splits.
2. ***Model bias*** : All the long-tail methods reduce the overall, medium-shot, and few-shot bias. The reduction tends to be greater for the tail classes. Our method decreases bias more than other methods on the tail classes.
3. ***Model variance*** : All the current long-tail methods increase the overall, many-shot, medium-shot, and few-shot variance, except cRT has a slight reduction for medium-shot. Our method reduces variances throughout the class spectrum.

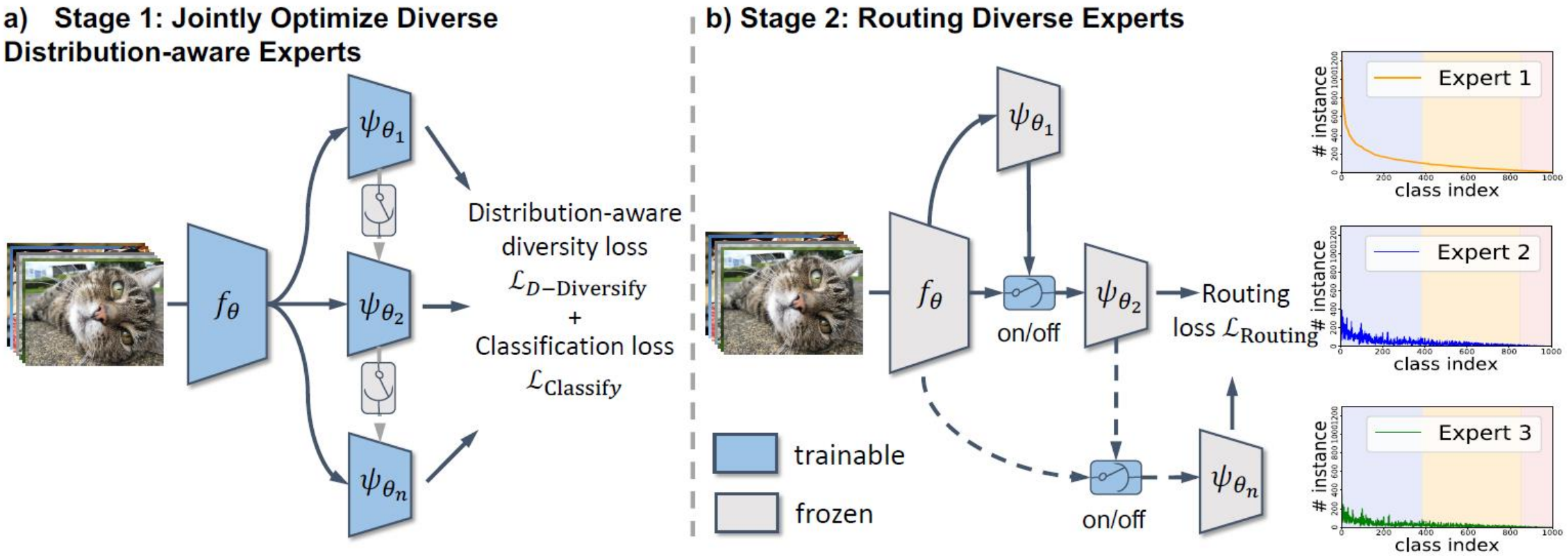
$$\text{Error}(x; h) = E[(h(x; D) - Y)^2] = \text{Bias}(h)^2 + \text{Variance}(h) + \text{irreducible error}.$$

	All			Many-shot			Med-shot			Few-shot		
	acc	bias	var	acc	bias	var	acc	bias	var	acc	bias	var
CE	31.6	0.60	0.47	57.3	0.28	0.35	28.2	0.61	0.51	6.3	0.94	0.57
τ-norm	35.8	0.52	0.49	55.9	0.28	0.37	33.2	0.53	0.52	16.1	0.78	0.60
cRT	36.4	**0.50**	0.50	51.3	0.32	0.41	38.6	**0.44**	0.50	17.0	0.76	0.61
LDAM	34.4	0.53	0.51	55.1	0.28	0.38	31.9	0.53	0.54	13.9	0.81	0.63
RIDE + LDAM	**40.5**	**0.50**	**0.42**	**60.5**	0.28	**0.30**	**38.7**	0.50	**0.44**	**20.1**	**0.74**	**0.52**

(a) Comparisons of the mean accuracy, per-class bias and variance of baseline methods and our RIDE method. Better (worse) metrics than the distribution-unaware cross entropy (CE) reference are marked in green (red).

RoutIng Diverse Experts (RIDE)

- RIDE adopts multiple experts for model variance reduction and employs an additional distribution aware diversity loss for reducing the model bias.



RoutIng Diverse Experts (RIDE) : Stage 1

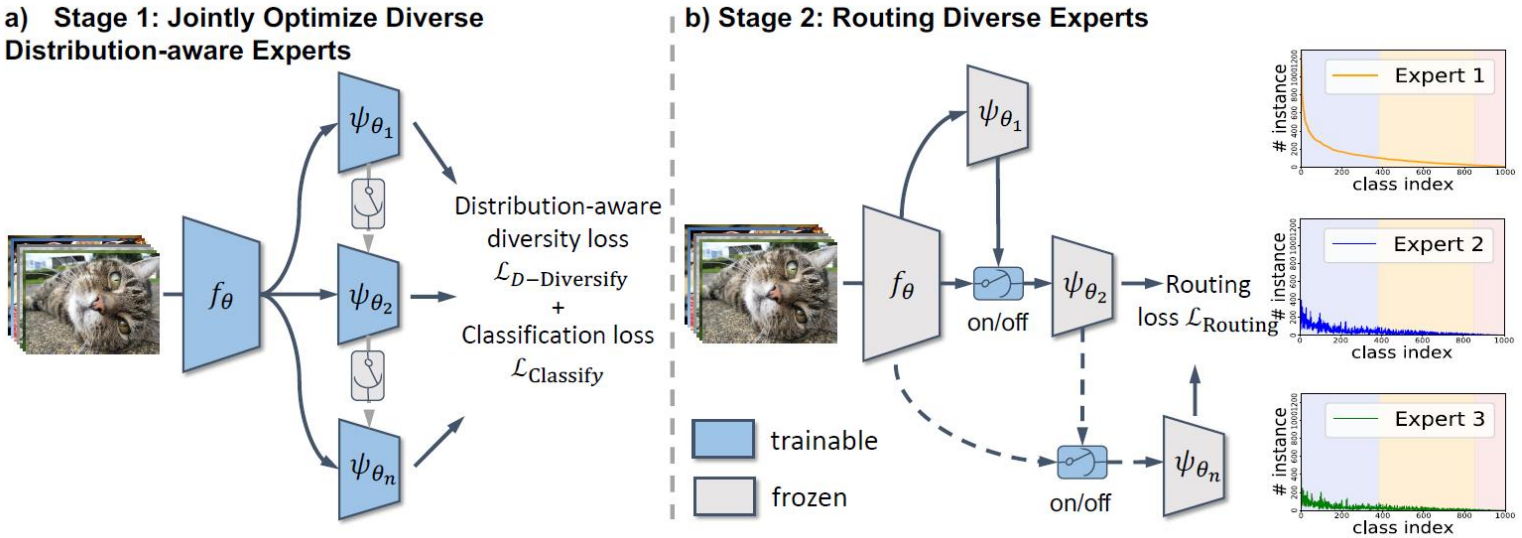
- It is necessary to encourage diversity to tail classes to alleviate the influence of noise.
- Give lower temperature to tail classes, which generates higher probability for the tail classes in distributions that we apply KL Divergence on, encouraging more diversity in tail classes.

$$\mathcal{L}^i_{\text{D-Diversify}} = -\frac{\lambda}{k-1}\sum^{n} D_{\text{KL}}(\phi^i(\vec{x},\vec{T}),\phi^j(\vec{x},\vec{T}))$$

$$\mathcal{L}^i_{\text{Total}} = \mathcal{L}^i_{\text{Classify}}(\phi^i(\vec{x}),y) - \frac{\lambda}{n-1}\sum_{j\neq i}^{n} D_{\text{KL}}(\phi^i(\vec{x},\vec{T}),\phi^j(\vec{x},\vec{T}))$$

$$\phi^i(\vec{x},\vec{T}) = \text{softmax}(\psi_{\theta_i}(f_\theta(\vec{x}))/\vec{T}),$$

$$T_i = \eta\psi_i + \eta(1-\max(\Psi)); \Psi = \{\psi_1,...,\psi_C\} = \{\gamma\cdot C\cdot\frac{n_i}{\sum_{k=1}^{C} n_k} + (1-\gamma)\}_{i=1}^{C}$$

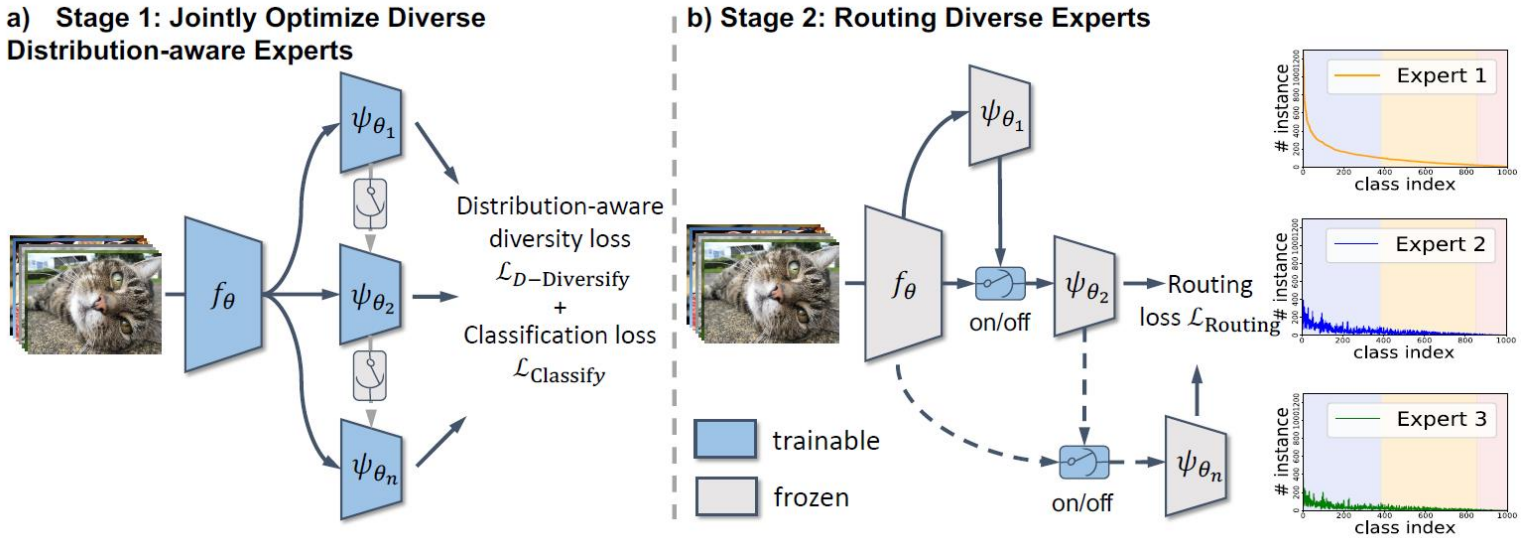


RoutIng Diverse Experts (RIDE) : Stage II

- Take the top z scores of logits, since other logits are generally too small to provide much information about the "ambiguity" of the sample.
- Ground truth y is constructed as: if the current expert does not predict the sample correctly but one of the next expert gives correct prediction, the ground truth is set to 1 (considered as positive), otherwise is 0.

$$\mathcal{L}_{\text{Routing}} = -\omega_{\text{p}} y \log(\frac{1}{1+e^{-y_{\text{ea}}}}) - \omega_{\text{n}}(1-y)\log(1-\frac{1}{1+e^{-y_{\text{ea}}}})$$

$$y_{\text{ea}} = \mathbf{W}_2(\vec{l}_i \oplus \sigma(\mathbf{W}_1 \vec{v}_i))$$



Experiments

Table 1: Top-1 accuracy comparison with state-of-the-arts on **CIFAR100-LT** with an imbalance ratio of 100. Compared with BBN (Zhou et al., 2020) and LFME (Xiang & Ding, 2020), which also contain multiple experts (or branches), RIDE (2 experts) outperforms them by a large margin with fewer GFlops. The relative computation cost (averaged on testing set) with respect to the baseline model and absolute improvements against SOTA (colored in green) are reported. † denotes our reproduced results with released code. ‡ denotes results copied from (Cao et al., 2019).

Methods	MFlops	Acc. (%)	Many	Med	Few
Cross Entropy (CE) ‡	69.5 (1.0x)	38.3	-	-	-
Cross Entropy (CE) †	69.5 (1.0x)	39.1	66.1	37.3	10.6
Focal Loss ‡ (Lin et al., 2017)	69.5 (1.0x)	38.4	-	-	-
OLTR (Liu et al., 2019)	-	41.2	61.8	41.4	17.6
LDAM + DRW (Cao et al., 2019)	69.5 (1.0x)	42.0	-	-	-
LDAM + DRW † (Cao et al., 2019)	69.5 (1.0x)	42.0	61.5	41.7	20.2
BBN (Zhou et al., 2020)	74.3 (1.1x)	42.6	-	-	-
τ-norm † (Kang et al., 2020)	69.5 (1.0x)	43.2	65.7	43.6	17.3
cRT † (Kang et al., 2020)	69.5 (1.0x)	43.3	64.0	44.8	18.1
M2m (Kim et al., 2020)	-	43.5	-	-	-
LFME (Xiang & Ding, 2020)	-	43.8	-	-	-
RIDE (2 experts)	**64.8 (0.9x)**	47.0 (+3.2)	67.9	48.4	21.8
RIDE (3 experts)	77.8 (1.1x)	48.0 (+4.2)	68.1	49.2	23.9
RIDE (4 experts)	91.9 (1.3x)	**49.1** (+5.3)	**69.3**	**49.3**	**26.0**

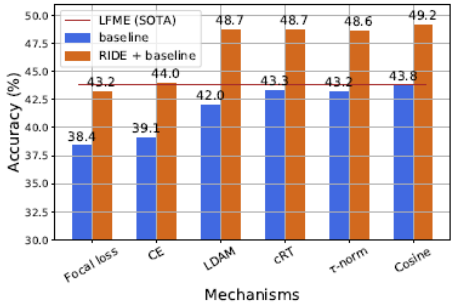


Figure 3: **Extend RIDE to various long-tailed recognition methods.** Consistent improvements can be observed on CIFAR100-LT, which illustrates that the proposed method can be applied to various training mechanisms, either methods that are end-to-end (e.g. LDAM) or require another stage of process (e.g. cRT and τ-norm). By using RIDE, cross-entropy loss (without any re-balancing strategies) can even outperforms current SOTA method LFME. Although higher accuracy can be obtained using distillation, we did not apply it here.

Table 2: Top-1 accuracy comparison with state-of-the-art methods on **ImageNet-LT** (Liu et al., 2019) with ResNet-50 and ResNeXt-50. RIDE achieves consistent performance improvements on various backbones. Results marked with † are copied from (Kang et al., 2020). We compare GFlops against the baseline model. Detailed results on each split are listed in appendix materials.

Methods	ResNet-50		ResNeXt-50	
	GFlops	Acc. (%)	GFlops	Acc. (%)
Cross Entropy (CE)†	4.11 (1.0x)	41.6	4.26 (1.0x)	44.4
OLTR† (Liu et al., 2019)	-	-	-	46.3
NCM (Kang et al., 2020)	4.11 (1.0x)	44.3	4.26 (1.0x)	47.3
τ-norm (Kang et al., 2020)	4.11 (1.0x)	46.7	4.26 (1.0x)	49.4
cRT (Kang et al., 2020)	4.11 (1.0x)	47.3	4.26 (1.0x)	49.6
LWS (Kang et al., 2020)	4.11 (1.0x)	47.7	4.26 (1.0x)	49.9
RIDE (2 experts)	**3.71 (0.9x)**	54.4 (+6.7)	**3.92 (0.9x)**	55.9 (+6.0)
RIDE (3 experts)	4.36 (1.1x)	54.9 (+7.2)	4.69 (1.1x)	56.4 (+6.5)
RIDE (4 experts)	5.15 (1.3x)	**55.4** (+7.7)	5.19 (1.2x)	**56.8** (+6.9)

Experiments

Table 3: Comparison with state-of-the-art methods on **iNaturalist** (Van Horn et al., 2018). RIDE outperforms current SOTA BBN, which also contains multiple "experts", by a large margin on many-shot classes. Results marked with † are from BBN (Zhou et al., 2020) and Decouple (Kang et al., 2020). BBN's results are from the released checkpoint. Relative improvements to SOTA result of each split (colored with gray) are also listed, with the largest boost from few-shot classes.

Methods	GFlops	All	Many	Medium	Few
CE †	4.14 (1.0x)	61.7	72.2	63.0	57.2
CB-Focal †	4.14 (1.0x)	61.1	-	-	-
OLTR †	4.14 (1.0x)	63.9	59.0	64.1	64.9
LDAM + DRW †	4.14 (1.0x)	64.6	-	-	-
cRT	4.14 (1.0x)	65.2	69.0	66.0	63.2
τ-norm	4.14 (1.0x)	65.6	65.6	65.3	65.9
LWS	4.14 (1.0x)	65.9	65.0	66.3	65.5
BBN	4.36 (1.1x)	66.3	49.4	70.8	65.3
RIDE (2 experts)	**3.67 (0.9x)**	71.4 (+5.1)	70.2 (+1.2)	71.3 (+0.5)	71.7 (+5.8)
RIDE (3 experts)	4.17 (1.0x)	72.2 (+5.9)	70.2 (+1.2)	72.2 (+1.4)	72.7 (+6.8)
RIDE (4 experts)	4.51 (1.1x)	**72.6** (+6.3)	**70.9** (+1.9)	**72.4** (+1.6)	**73.1** (+7.2)

Methods	#expert	$\mathcal{L}_{\text{Diversify}}$	$\mathcal{L}_{\text{D-Diversify}}$	EA	distill	GFlops	Acc. (%)
LDAM + DRW							42.0
RIDE	2	✓				1.1x	45.0 (+3.0)
	2		✓			1.1x	46.6 (+4.6)
	2		✓		✓	1.1x	47.3 (+5.3)
	2		✓	✓	✓	**0.9x**	47.0 (+5.0)
	3		✓	✓	✓	1.1x	48.0 (+6.0)
	4		✓	✓	✓	1.3x	**49.1** (+7.1)

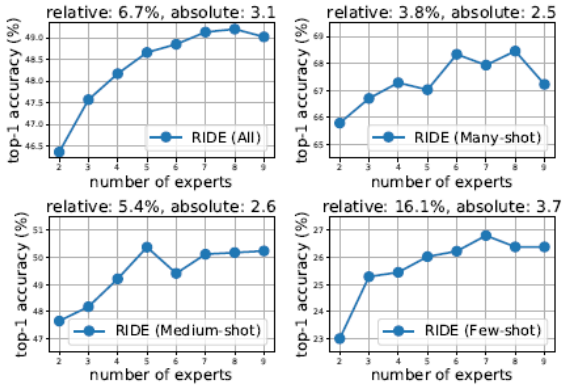


Figure 4: **# experts vs. top-1 accuracy for each split** (All, Many/Medium/Few) of CIFAR100-LT. Compared with the many-shot split, which is 3.8% relatively improved by adding more experts, the few-shot split can get more benefits, that is, a relative improvement of 16.1%.

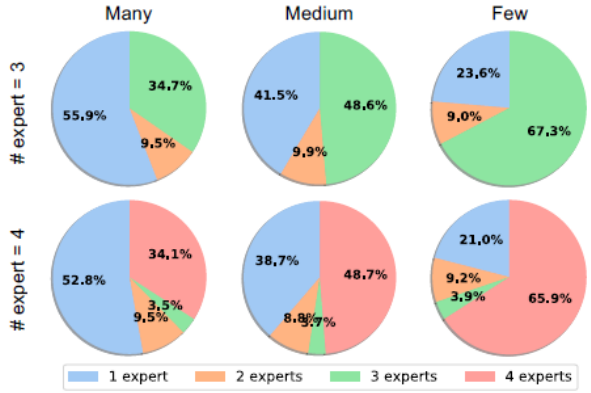


Figure 5: **The proportion of the number of experts** allocated to each split of CIFAR100-LT. For RIDE with 3 or 4 experts, more than half of many-shot instances only require one expert. On the contrary, more than 76% samples of few-shot classes require opinions from additional experts.

Thank you!