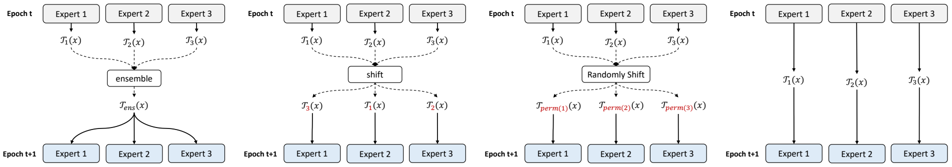
Predictive Consistency Learning for Long-Tailed Recognition (Supplementary Material)

Nan Kang[1,2]
nan.kang@vipl.ict.ac.cn
Hong Chang[1,2]
changhong@ict.ac.cn
Bingpeng Ma[2]
bpma@ucas.ac.cn
Shutao Bai[1,2]
shutao.bai@vipl.ict.ac.cn
Shiguang Shan[1,2,3]
sgshan@ict.ac.cn
Xilin Chen[1,2]
xlchen@ict.ac.cn

[1] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS
Beijing, 100190, China
[2] University of Chinese Academy of Sciences
Beijing, 100049, China
[3] CAS Center for Excellence in Brain Science and Intelligence Technology,
Shanghai, 200031, China

1 Learning with Ensemble

Our method can serve as a plug-and-play module for existing methods. Here we incorporate our method into the ensemble method [2]. When combined with multiple experts, there could be multiple practical ways to construct $T(x)$: (1) **Ensemble**: using the ensemble of all experts. (2) **Shift**: learning from shifted (nearby) experts' predictions. (3) **RandShift**: learning from randomly shifted experts, *i.e*. shifting the experts randomly for each iteration. (4) **Independent**: training each expert independently without interactions among experts. We demonstrate the framework of these strategies in Fig. 1.



(a) Ensemble (b) Shift (c) RandShift (d) Independent

Figure 1: Demonstration of different possible strategies of generating $\mathcal{T}(x)$ for RIDE.

In Tab. 1, we compare the performance of different strategies on CIFAR-100-LT and ImageNet-LT with the original RIDE. The results show that our method achieves better performance than the original RIDE with cross-entropy loss, which again proves the effectiveness of our method.

Notably, we observe that the first three strategies yield sub-optimal results, which is somewhat counter-intuitive. Despite the better performance of the ensemble predictions in (1), it does not lead to better improvement for PCL. We attribute this to the diverse predictions generated by different experts. As discussed in Sec. 3.2, for tail classes with high predictive uncertainty, the predictions of nearby experts may exhibit inconsistencies with the current expert, thereby negatively impacting the probabilistic estimation of $\hat{p}(y|x)$.

Table 1: Comparison of different strategies of generating $\mathcal{T}(x)$ for RIDE.

Strategy	RIDE-CE	Ens	Shf	RShf	Ind
CIFAR-100-LT	49.5	50.2	50.6	50.6	**50.8**
ImageNet-LT (90 epochs)	54.1	54.4	54.3	54.4	**54.8**

2 More Results

2.1 Test Time Distribution Shift

We further compare our method with baselines on different datasets under various training distributions and test-time distribution shifts. In this comparison, we employ two different post-hoc correction methods, namely the *additive* correction (PC-Softmax) [1] and the *multiplicative* correction (CDT) [3]. It is worth noting that the original CDT is not a post-hoc method. Instead, it introduces a class-dependent temperatures a_c to the loss:

$$\mathcal{L}_{CDT}(f_\theta(x), i) = -\log \frac{e^{f_\theta(x)[i]/a_i}}{\sum_c e^{f_\theta(x)[c]/a_c}}, \tag{1}$$

where $a_y = (N_1/N_y)^\tau$ and τ is a hyper-parameter. To convert CDT into a post-hoc method, we consider the $p_t(y)$ and introduce the $a_c^t = (M_{max}/M_c)^\tau$ temperature for target distribution. Then, we reverse the sign of the temperatures and apply it to the test predictions similar to PC-Softmax. Consequently, the post-hoc version of CDT is given by:

$$\arg\max_c f_\theta(x)[c] \cdot a_c / a_c^t. \tag{2}$$

The results are demonstrated in Fig. 2. As shown, PCL improves both the multiplicative correction and additive correction over all test distributions. Specifically, the advantage of PCL over CE is even larger under large label distribution shift (backward regions). Note that multiplicative correction is generally less effective than additive correction, indicating that additive correction is a better solution for post-hoc correction for long-tailed classification.

2.2 Expected Predictions

In Fig. 1 of the main text, we illustrate the mismatch between the label distribution $p_t(y)$ and average prediction $\mathbb{E}_x[\hat{p}_t(y|x;\theta)]$. In this analysis, we quantify the mismatch with the KL divergence between them. As depicted in Tab. 2, the KL divergence gradually increases as the test distribution shift becomes larger (from left to right). This indicates that the approximation becomes less accurate due to more severe label distribution shift. Compared to

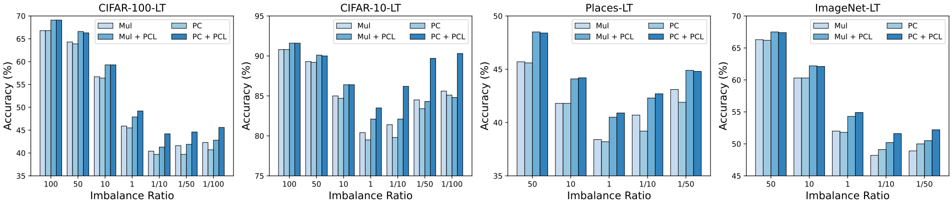


Figure 2: Comparison of recognition accuracy on test time shifted long-tailed datasets.

the baseline, our method achieves much lower KL divergence, indicating that our method produces better predictions.

Table 2: The KL divergence between $\mathbb{E}_x[\hat{p}_t(y|x;\theta)]$ and $p_t(y)$ on test time shifted datasets with PC-Softmax.

CIFAR-10-LT	Forward					Uniform	Backward				
Imbalance ratio	50	25	10	5	2	1	2	5	10	25	50
CE $\times 10^{-2}$	0.25	0.34	0.55	0.77	1.13	1.44	1.85	2.42	2.83	3.12	3.19
Ours $\times 10^{-2}$	**0.13**	**0.13**	**0.15**	**0.17**	**0.19**	**0.20**	**0.24**	**0.30**	**0.34**	**0.39**	**0.43**

CIFAR-100-LT	Forward					Uniform	Backward				
Imbalance ratio	50	25	10	5	2	1	2	5	10	25	50
CE $\times 10^{-2}$	1.32	1.46	1.78	2.19	2.91	3.51	4.37	5.51	6.27	7.05	7.49
Ours $\times 10^{-2}$	**0.42**	**0.42**	**0.43**	**0.45**	**0.52**	**0.60**	**0.76**	**0.99**	**1.16**	**1.35**	**1.45**

ImageNet-LT	Forward					Uniform	Backward				
Imbalance ratio	50	25	10	5	2	1	2	5	10	25	50
CE $\times 10^{-2}$	2.46	2.43	2.43	2.49	2.62	2.77	3.15	3.65	3.97	4.29	4.49
Ours $\times 10^{-2}$	**1.06**	**1.02**	**1.03**	**1.09**	**1.21**	**1.35**	**1.59**	**1.85**	**2.00**	**2.14**	**2.17**

Places-LT	Forward					Uniform	Backward				
Imbalance ratio	50	25	10	5	2	1	2	5	10	25	50
CE $\times 10^{-2}$	1.11	1.15	1.18	1.23	1.32	1.41	1.49	1.64	1.73	1.88	1.98
Ours $\times 10^{-2}$	**0.66**	**0.69**	**0.70**	**0.73**	**0.80**	**0.85**	**0.97**	**1.12**	**1.22**	**1.42**	**1.49**

3 Details of Hyperparameters

In Tab. 3, we summarize the hyperparameters and implementation details for different datasets. Besides, the EMA factor β is set to 0.999 for PCL and 1.0 for ComPCL. The value of coverage level γ depends on the degree of compression, which represents a tradeoff between accuracy and compression ratio. We simply set γ to 0.95 in the experiments.

Table 3: Summary of the hyperparameters and implementation details used in our method.

Dataset	ImageNet-LT	Places-LT	CIFAR-10-LT	CIFAR-100-LT
Imbalance ratio	256	996	$\{10, 50, 100\}$	$\{10, 50, 100\}$
Network				
Backbone	ResNet-50	ResNet-152	ResNet-32	ResNet32
Classifier	Cosine	Cosine	Linear	Linear
Pretrain	-	ImageNet	-	-
Training				
Epochs	90 / 200	30	200	200
Batch Size	256	128	256	256
Learning Rate	0.1	0.001 & 0.1	0.2	0.2
LR Schedule	Cosine	Cosine	Step	Step
Weight Decay Factor	5×10^{-4}			
Method				
α	0.7	0.4	$\{0.8, 0.9, 0.9\}$	$\{0.7, 0.8, 0.9\}$
λ	0.3	0.1	$\{0.4, 0.4, 0.5\}$	$\{0.6, 0.6, 0.6\}$

References

[1] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6626–6636, 2021.

[2] 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.

[3] Han-Jia Ye, Hong-You Chen, De-Chuan Zhan, and Wei-Lun Chao. Identifying and compensating for feature deviation in imbalanced deep learning. *arXiv preprint arXiv:2001.01385*, 2020.