Trustworthy Long-Tailed Classification

A. Method

A.1. Discussion of Conflict Factor

The conflict factor C is small with similar beliefs and is big with dissimilar beliefs. Let $\boldsymbol{B}$ be the "belief confusion matrix" computed by: $\boldsymbol{B} = \boldsymbol{b}^1 \boldsymbol{b}^{2^T}$. The conflict factor C can be re-formalized as:

$$\begin{aligned} C &= \sum_{i \neq j} b_i^1 b_j^2 \\ &= \sum_{i,j} b_i^1 b_j^2 - \sum_i b_i^1 b_i^2 \\ &= \sum_{i,j} B_{ij} - trace(\boldsymbol{B}). \end{aligned} \tag{1}$$

Since the sum of uncertainty and all belief masses is a constant: $u + \sum_k b_k = 1$, we have $\sum_{i,j} B_{ij} = \sum_i b_i^1 \sum_j b_j^2 = \sum_i b_i^1 (1 - u^2) = (1 - u^2) \sum_i b_i^1 = (1 - u^1)(1 - u^2)$, and $trace(\boldsymbol{B}) = \boldsymbol{b}^{1^T} \boldsymbol{b}^2$. Therefore, the conflict factor can be written as:

$$C = (1 - u^1)(1 - u^2) - \boldsymbol{b}^{1^T} \boldsymbol{b}^2. \tag{2}$$

It is clear that C is opposite to the inner product $\boldsymbol{b}^{1^T} \boldsymbol{b}^2$ (indicating similarity between two experts) when the uncertainties remain stable.

A.2. Training Process

We summarize the training process in Algorithm 1, including forming the belief masses, computing the uncertainty, and applying the strategy of dynamic expert engagement.

Algorithm 1: The training process of TLC.

Inputs: Long-tailed data $\{\boldsymbol{X}_i, y_i\}_{i=1}^N$;
Initialization: Initialize the M-expert model;
for *epoch = 1,2,...* **do**
  $\mathcal{L} \leftarrow 0$;
  for *m = 1,2,...,M* **do**
    $\boldsymbol{e}^m \leftarrow$ the output of the m-th expert;
    $\boldsymbol{\alpha}^m \leftarrow \boldsymbol{e}^m + 1$;
    $S^m \leftarrow \sum_{k=1}^K \alpha_k^m$;
    $u^m \leftarrow K / S^m$;
    Compute the loss of the m-th expert: $\mathcal{L}^m$;
    Compute the prefix weight: w^m;
    if $w^m > \tau$ **then**
      $\mathcal{L} \leftarrow \mathcal{L} + \mathcal{L}^m$
  end
  Update the parameters of the model with gradient descent on $\mathcal{L}$.
end

B. Experiments

B.1. Datasets

Long-tailed CIFAR. The CIFAR-10-LT and CIFAR-100-LT are formed by sampling a subset from the original CIFAR dataset[1] with exponential distributions [2]. Specifically, for the i-th class, the smapled number of images is $n_i = n\mu^i$, where n is the class volume of each class in the original dataset and $\mu \in (0, 1)$. The imbalance ratio is defined as the relative number of samples in the first class to that of the last class.

Long-tailed ImageNet. The ImageNet-LT dataset[2] is first proposed by [5]. It is sampled from the original ImageNet-2012 [3] over the Pareto distribution with the power value $\alpha = 6$. Overall, it has 115.8K images from 1,000 categories.

B.2. Implementation Details

The initial implementation is based on PyTorch [6] and we will use MindSpore [1] in future work. The backbone for CIFAR-10-LT and CIFAR-100-LT datasets is the ResNet32 [4] with the first block shared across experts. The backbone for ImageNet-LT dataset is the ResNet50 [4] with the first 3 blocks shared across experts. Since the proposed method is general, it can be implemented with other backbones. At training, SGD optimizer is adopted to update the parameters and set the base learning rate as 0.1 for all models. The learning rate will experience a warm-up of the first 5 epochs and a decay of the last 40 epochs. We also apply

[1] https://www.cs.toronto.edu/~kriz/cifar.html

[2] https://drive.google.com/drive/u/0/folders/1j7Nkfe6ZhzKFXePHdsseeeGI877Xu1yf

Table 1. Hyperparameter specifications.

Dataset	Learning Rate	Training Epochs	Warm-up Epochs	η	τ	λ_{div}	Batch Size	Base Model	Optimizer
CIFAR-10-LT	0.1	200	5	0.1	0.52	0.01	128	ResNet32	SGD
CIFAR-100-LT	0.1	200	5	0.2	0.54	0.01	128	ResNet32	SGD
ImageNet-LT	0.1	100	5	0.5	0.42	0.05	256	ResNet50	SGD

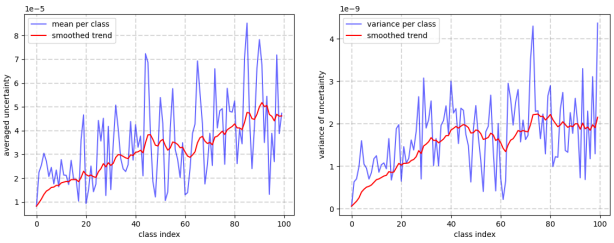
Table 2. Computational cost comparison at training.

Dataset	CIFAR-10-LT		CIFAR-100-LT		ImageNet-LT	
Dynamic expert engagement	×	✓	×	✓	×	✓
Flops	276.45M	**209.25M**	276.45M	**217.52M**	4.53G	**4.17G**
Actually trained parameters	1,855,184	**1,391,504**	1,878,224	**1,408,784**	45,396,032	**35,478,144**

the warm-up strategy [8] in the first 5 epochs and reduce and learning rate in the last 40 and 20 epochs consecutively. We set the number of experts as 2, 3 and 4 in the quantitative evaluations, which is consistent with [7]. We choose the hyperparameters by gird search based on experimental results, in which they are tested and selected with the stride 0.02. The hyperparameter specifications are summarized in Table. 1.

B.3. Empirical Study of Uncertainty

An assumption about Evidence-based Uncertainty is that the uncertainty is low for easy samples and is large for hard samples. Visual support is provided in Fig. 1 by showing the means and variances of uncertainties in different classes. It is clear that: i) tailed classes with averagely more hard samples have larger uncertainties, and ii) uncertainties in tailed classes are distributed more dispersedly, which implies more detected hard samples.



(a) Means (b) Variances

Figure 1. Uncertainty statistics per class.

B.4. Discussion of Computational Cost

The proposed strategy of dynamic expert engagement is marked by reducing redundant experts for easy samples at training. Although the number of experts at training has been discussed in the ablation study, we provide direct comparison of computational cost in Table. 2 for clearness. We use *torchstat*[3] to count the Flops and actually trained parameters. The results show that the proposed TLC does perform more efficiently at training.

References

[1] Lei Chen. *Deep Learning and Practice with MindSpore*. Springer Nature, 2021. 1

[2] 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. 1

[3] 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

[4] 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. 1

[5] 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. 1

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 1

[7] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2020. 2

[8] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei

[3] https://github.com/Swall0w/torchstat

Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020. 2