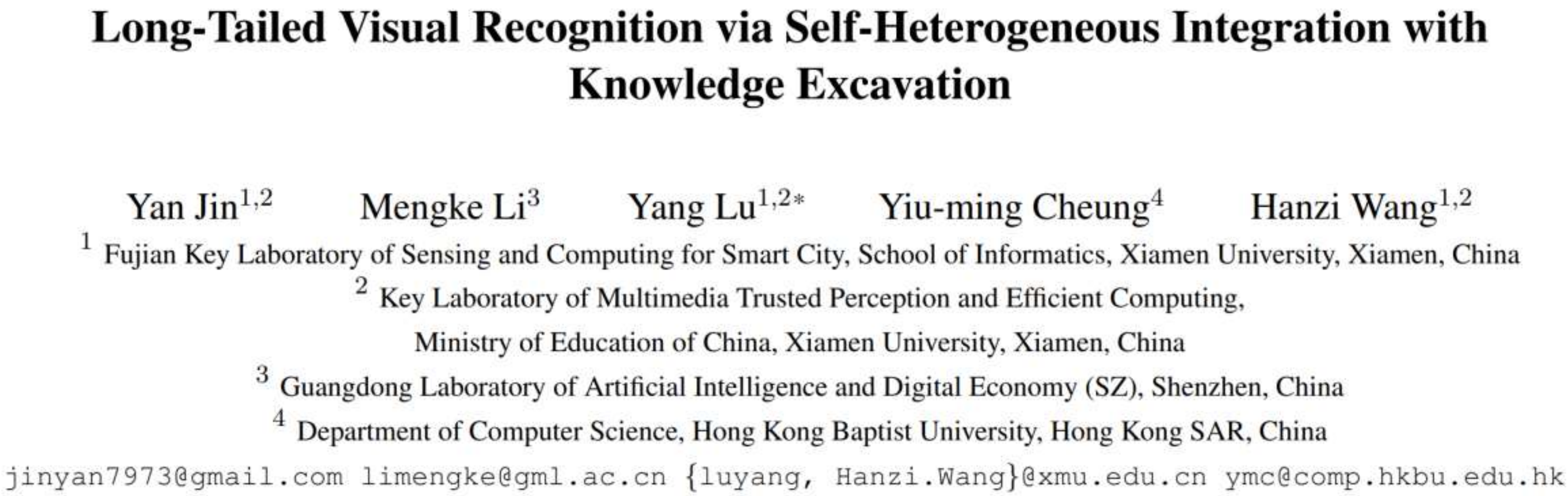


Long-Tailed Visual Recognition via Self-Heterogeneous Integration with Knowledge Excavation

Yan Jin[1,2] Mengke Li[3] Yang Lu[1,2*] Yiu-ming Cheung[4] Hanzi Wang[1,2]

[1] Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, Xiamen, China

[2] Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, Xiamen University, Xiamen, China

[3] Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China

[4] Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China

jinyan7973@gmail.com limengke@gml.ac.cn {luyang, Hanzi.Wang}@xmu.edu.cn ymc@comp.hkbu.edu.hk

CVPR 2023

2023.9.1

Mixture of Experts (MoE): Models trained on individual groups are ensembled together in a multi-expert framework.

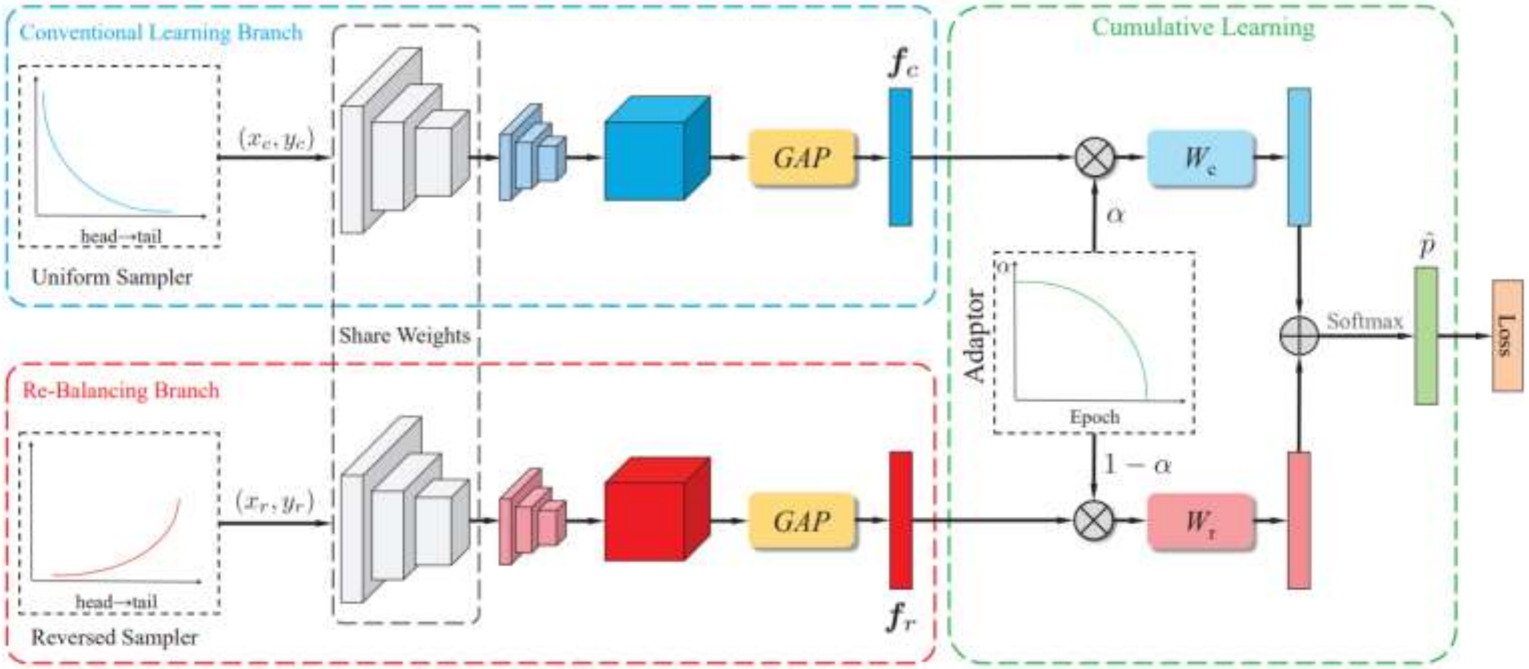


Figure 3. Framework of our Bilateral-Branch Network (BBN). It consists of three key components: 1) The *conventional learning branch* takes input data from a uniform sampler, which is responsible for learning universal patterns of original distributions. While, 2) the *re-balancing branch* takes inputs from a reversed sampler and is designed for modeling the tail data. The output feature vectors $\boldsymbol{f}_c$ and $\boldsymbol{f}_r$ of two branches are aggregated by 3) our *cumulative learning strategy* for computing training losses. "*GAP*" is short for global average pooling.

BBN: Bilateral-Branch Network with Cumulative Learning for Long-Tailed Visual Recognition

Boyan Zhou[1] Quan Cui[1,2] Xiu-Shen Wei[1*] Zhao-Min Chen[1,3]
[1]Megvii Technology [2]Waseda University [3]Nanjing University

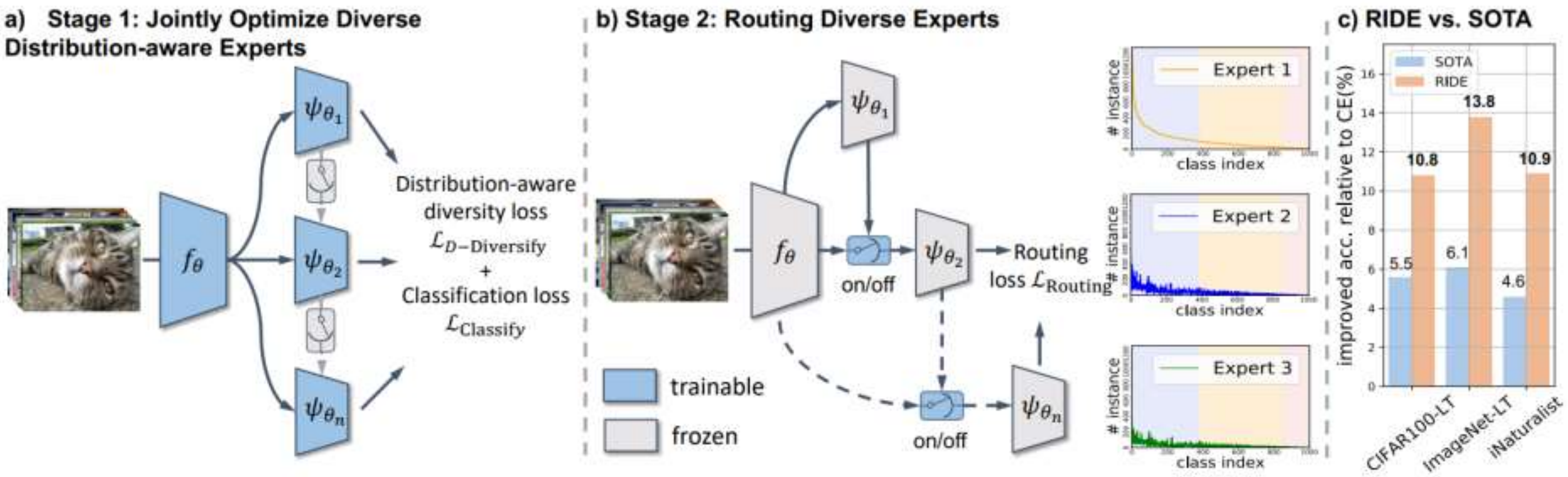

a) Stage 1: Jointly Optimize Diverse Distribution-aware Experts
f_θ
ψ_{θ_1}
ψ_{θ_2}
ψ_{θ_n}
Distribution-aware diversity loss $\mathcal{L}_{D-\text{Diversify}}$
+
Classification loss $\mathcal{L}_{\text{Classify}}$
b) Stage 2: Routing Diverse Experts
on/off
on/off
Routing loss $\mathcal{L}_{\text{Routing}}$
trainable
frozen
Expert 1
Expert 2
Expert 3
# instance
class index
c) RIDE vs. SOTA
SOTA
RIDE
improved acc. relative to CE(%)
5.5
10.8
6.1
13.8
4.6
10.9
0
2
4
6
8
10
12
14
16
CIFAR100-LT
ImageNet-LT
iNaturalist

Figure 2: RIDE learns experts and their router in two stages. **a)** We first jointly optimize multiple experts with individual classification losses and mutual distribution-aware diversity losses. **b)** We then train a router that dynamically assigns *ambiguous* samples to additional experts on an as-needed basis. The distribution of instances seen by each expert shows that head instances need fewer experts and the imbalance between classes gets reduced for later experts. At the test time, we collect the logits of assigned experts to make a final decision. **c)** RIDE outperforms SOTA methods (i.e. LFME (Xiang et al., 2020) for CIFAR100-LT, LWS (Kang et al., 2020) for ImageNet-LT and BBN (Zhou et al., 2020) for iNaturalist) on all the benchmarks.

Long-tailed Recognition by Routing Diverse Distribution-Aware Experts

Xudong Wang[1], **Long Lian**[1], **Zhongqi Miao**[1], **Ziwei Liu**[2], **Stella X. Yu**[1]
[1]UC Berkeley / ICSI, [2]Nanyang Technological University
{xdwang,longlian,zhongqi.miao,stellayu}@berkeley.edu
ziwei.liu@ntu.edu.sg

知识蒸馏（Knowledge Distillation）

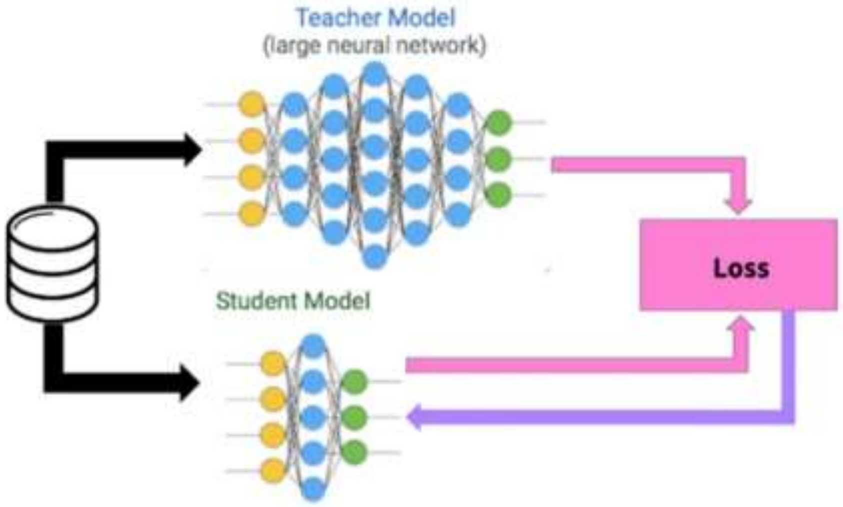
知识蒸馏的概念由Hinton在Distilling the Knowledge in a Neural Network中提出，目的是把 一个大模型或者多个模型集成学到的知识迁移到另一个轻量级模型上。

Knowledge Distillation，简称KD，顾名思义，就是将已经训练好的模型包含的知识(Knowledge)，蒸馏(Distill)提取到另一个模型里面去。

简而言之，就是模型压缩的一种方法，是**一种基于“教师-学生网络思想”的训练方法。**

Teacher Model和Student Model

知识蒸馏采取Teacher-Student模式：将复杂且大的模型作为Teacher，Student模型结构较为简单，用Teacher来辅助Student模型的训练，Teacher学习能力强，可以将它学到的知识迁移给学习能力相对弱的Student模型，以此来增强Student模型的泛化能力。复杂笨重但是效果好的Teacher模型不上线，就单纯是个导师角色，真正部署上线进行预测任务的是灵活轻巧的Student小模型。



Problem: Mixture of Experts (MoE) with the same model depth,
different classes may have different preferences with different depths.

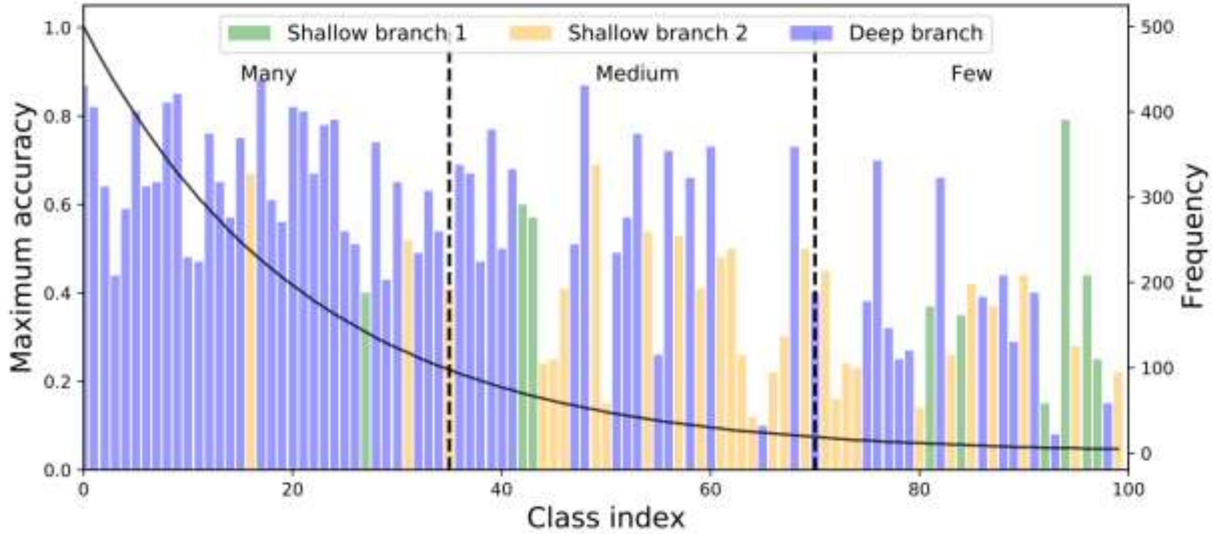


Figure 1. Comparison of test accuracy of a ResNet-32 model with two shallow branches and a deep branch. The model is jointly trained on CIFAR100-LT with an imbalance factor of 100. Only the highest accuracy among the three branches is shown for each class.

- We propose Depth-wise Knowledge Fusion (DKF) to encourage feature diversity in knowledge distillation among experts, which releases the potential of the MoE in long-tailed representation learning.
- We propose a novel knowledge distillation strategy DKT for MoE training to address the hardest negative problem for long-tailed data, which further exploits the diverse features fusing enabled by DKF.

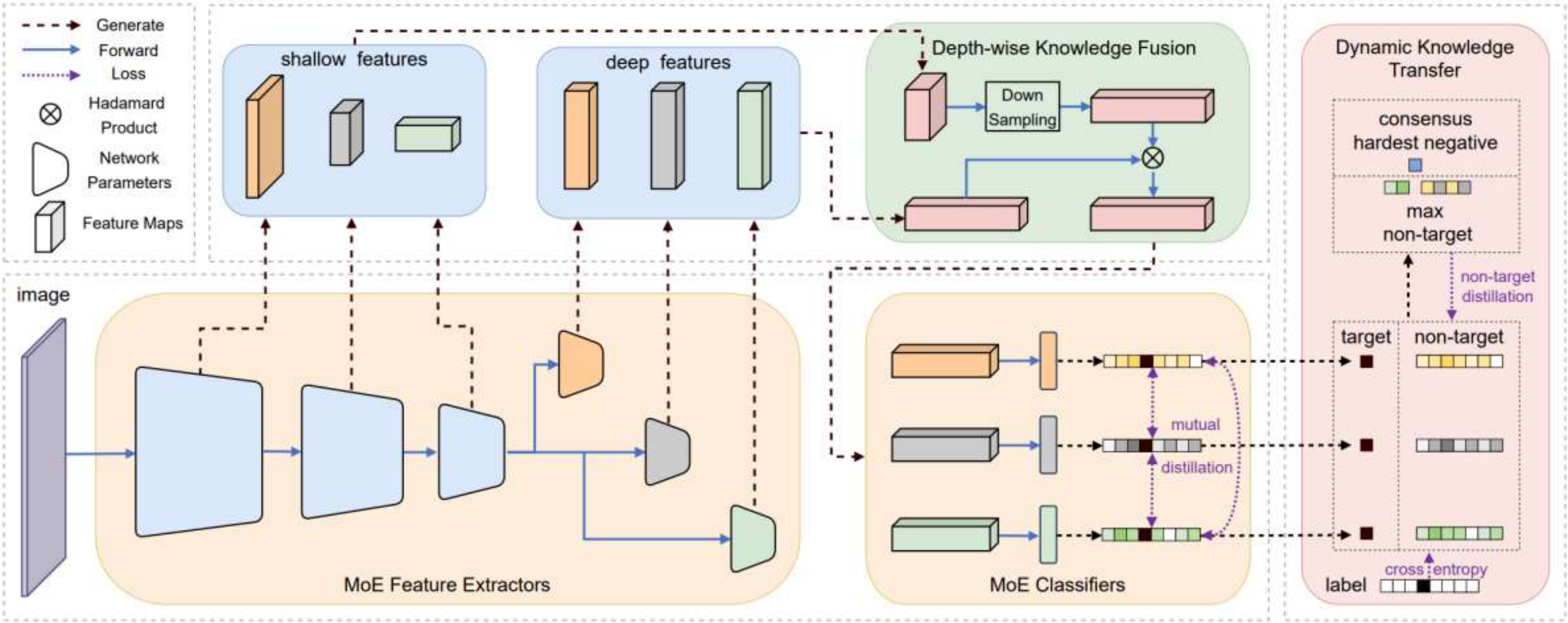
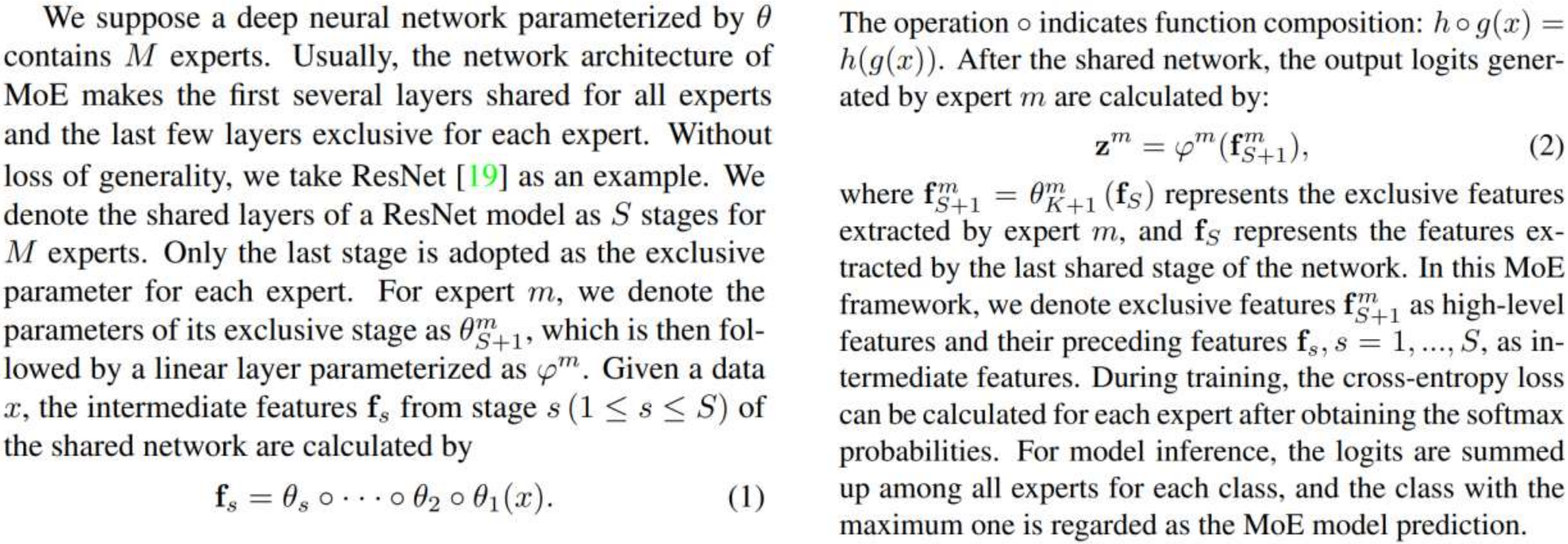


Figure 2. The structure of the proposed SHIKE. Each expert in MoE fuses the features from its own exclusive layers (deep features) and ones from the shared layers (intermediate features). The fused features are then used for mutual and dynamic knowledge distillation for better model optimization.



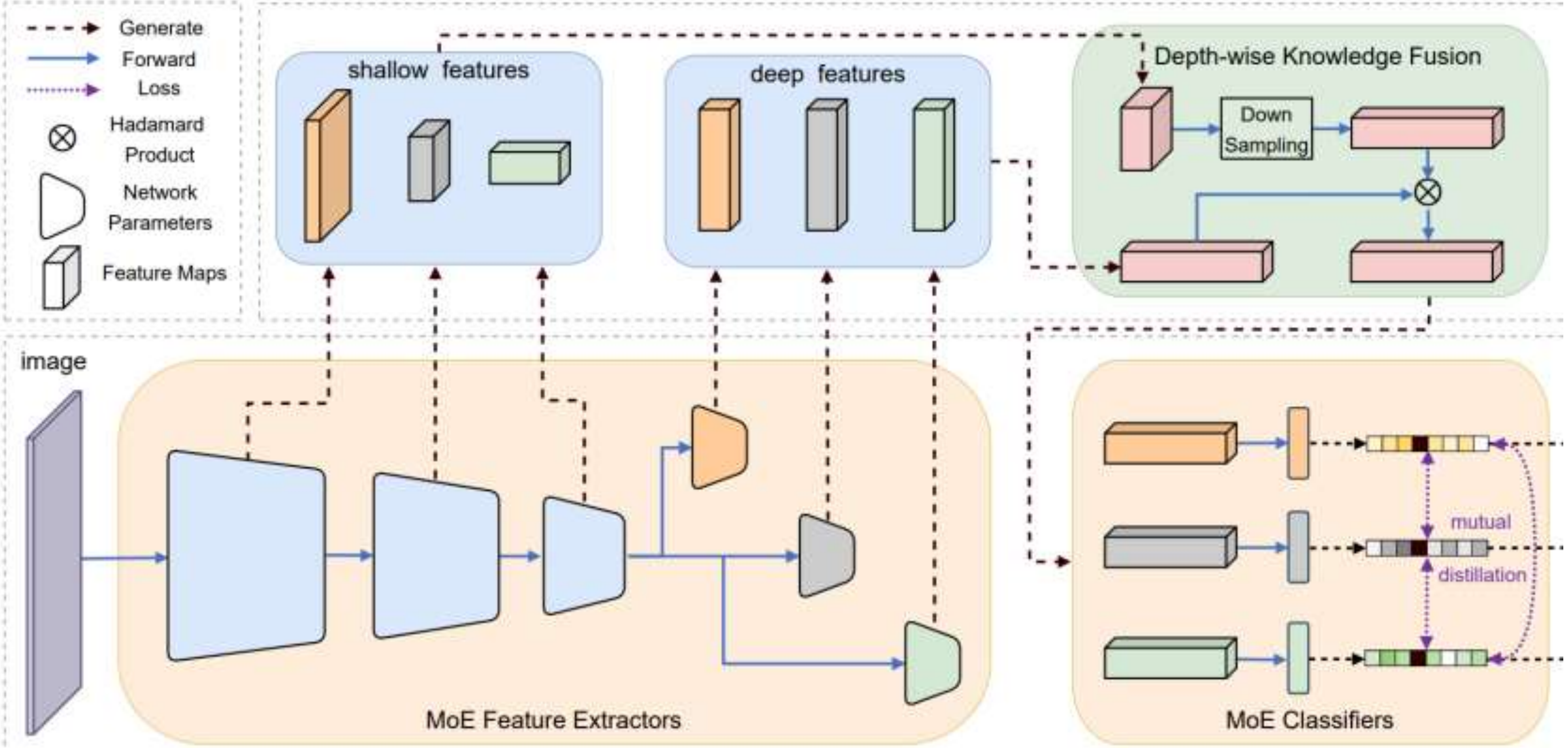
We suppose a deep neural network parameterized by θ contains M experts. Usually, the network architecture of MoE makes the first several layers shared for all experts and the last few layers exclusive for each expert. Without loss of generality, we take ResNet [19] as an example. We denote the shared layers of a ResNet model as S stages for M experts. Only the last stage is adopted as the exclusive parameter for each expert. For expert m, we denote the parameters of its exclusive stage as θ^{m}_{S+1}, which is then followed by a linear layer parameterized as φ^{m}. Given a data x, the intermediate features $\mathbf{f}_s$ from stage $s\,(1 \leq s \leq S)$ of the shared network are calculated by

$$\mathbf{f}_s = \theta_s \circ \cdots \circ \theta_2 \circ \theta_1(x). \tag{1}$$

The operation $\circ$ indicates function composition: $h \circ g(x) = h(g(x))$. After the shared network, the output logits generated by expert m are calculated by:

$$\mathbf{z}^{m} = \varphi^{m}(\mathbf{f}^{m}_{S+1}), \tag{2}$$

where $\mathbf{f}^{m}_{S+1} = \theta^{m}_{K+1}(\mathbf{f}_S)$ represents the exclusive features extracted by expert m, and $\mathbf{f}_S$ represents the features extracted by the last shared stage of the network. In this MoE framework, we denote exclusive features $\mathbf{f}^{m}_{S+1}$ as high-level features and their preceding features $\mathbf{f}_s, s = 1, ..., S$, as intermediate features. During training, the cross-entropy loss can be calculated for each expert after obtaining the softmax probabilities. For model inference, the logits are summed up among all experts for each class, and the class with the maximum one is regarded as the MoE model prediction.



3.2. Depth-Wise Knowledge Fusion

intermediate features among $\mathbf{f}_s$, $s = 1, ..., S$ to each expert. As different intermediate features have different sizes, one expert cannot simply concatenate or multiply them with the assigned features $\mathbf{f}_s$ directly. Therefore, we add several convolution layers for downsampling according to the depth of the intermediate features, achieving feature alignment between $\mathbf{f}_s$ and high-level features $\mathbf{f}^m_{S+1}$ extracted by expert m. Suppose the intermediate features after alignment are $\hat{\mathbf{f}}_s$. In DKF, we propose to fuse the intermediate features with high-level features by multiplication and then transform them into logits by φ^m:

$$\mathbf{z}^m = \varphi^m \left(\hat{\mathbf{f}}_s \otimes \mathbf{f}^m_{S+1} \right), \quad (3)$$

To fully use the diverse features in DKF, we can apply knowledge distillation between any two experts to make them learn from each other. As each expert in MoE often has the same architecture located in the deepest position of the network, it can be guaranteed that each expert can play the role of either a teacher or a student. This enables mutual knowledge distillation between any two experts and provides a perfect opportunity for experts to aggregate different depths of knowledge:

$$\mathcal{L}_{mu} = \sum_{j=1}^{M} \sum_{k \neq j}^{M} KL(\mathbf{p}^j | \mathbf{p}^k), \tag{4}$$

where $\mathbf{p}^j$ and $\mathbf{p}^k$ represent the softmax probabilities of class j and k, respectively. This guarantees the knowledge transferring comprehensively between any two experts.

KL divergence: $$D_{\mathrm{KL}}(\mathbf{p} \| \mathbf{q}) = \sum_{k=1}^{c} \mathbf{p}_k \log\left(\frac{\mathbf{p}_k}{\mathbf{q}_k}\right) \tag{11}$$

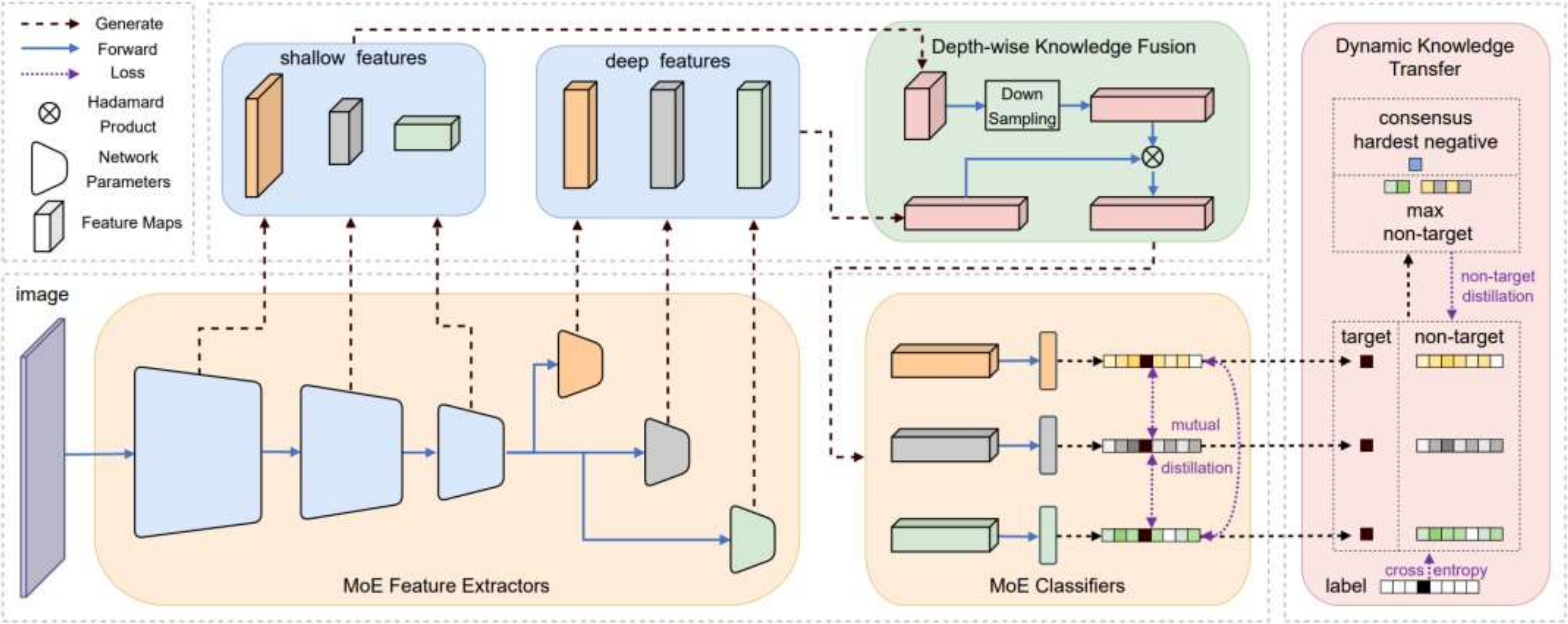
classification by θ_i : $$\mathbf{p}^{(i)}(x, y) = \mathrm{softmax}\left(\left[\frac{\psi_{\theta_i}(f_\theta(x))_1}{T_1} \quad \ldots \quad \frac{\psi_{\theta_i}(f_\theta(x))_c}{T_c}\right]\right). \tag{12}$$

class-wise temperature: $$T_k = \alpha\left(\beta_k + 1 - \max_j \beta_j\right) \tag{13}$$

normalized class size: $$\beta_k = \gamma \cdot \frac{n_k}{\frac{1}{c}\sum_{s=1}^{c} n_s} + (1 - \gamma). \tag{14}$$

3.3. Dynamic Knowledge Transfer

when they share similar semantic features. The non-target classes with high confidence logits are called hard negative classes [34,40]. It is dangerous to conduct knowledge distillation if the experts have a consensus on the hardest negative class, which may be a head class, for the tail class samples because the misleading information may be transferred.



For a sample x with label y, its corresponding output logits of expert m is $\mathbf{z}^m = [z_1^m, z_2^m, ..., z_C^m]$. Following [68], we first decouple the logits into a target logit z_y^m and non-target logits $[z_{I_1}^m, z_{I_2}^m, ..., z_{I_{C-1}}^m]$, where $\mathcal{I} = [I_1, I_2, ..., I_{C-1}]$ stores the index of non-target classes. After logits decoupling, we introduce the non-target set to a new knowledge distillation problem with $C-1$ classes. The average logits of all experts can be calculated for each non-target class $[\bar{z}_{I_1}, \bar{z}_{I_2}, ..., \bar{z}_{I_{C-1}}]$, where

$$\bar{z}_{I_i} = \frac{1}{M} \sum_{m=1}^{M} z_{I_i}^m, \tag{5}$$

for $i = 1, ..., C-1$. We can thus identify $\max_i\{\bar{z}_{I_i}\}$ among all non-target classes as the consensus hardest negative class, which is believed as the hardest negative class by joint prediction of MoE. To effectively suppress the logit of consensus hardest negative class through softmax suppression, a teacher who can comprehensively utilize the non-target knowledge is needed. Specifically, DKT chooses the maximum non-target logit among all experts denoted as $\hat{z}_{I_i}$:

$$\hat{z}_{I_i} = \max_{m=1,...,M} \{z_{I_i}^m\}, \tag{6}$$

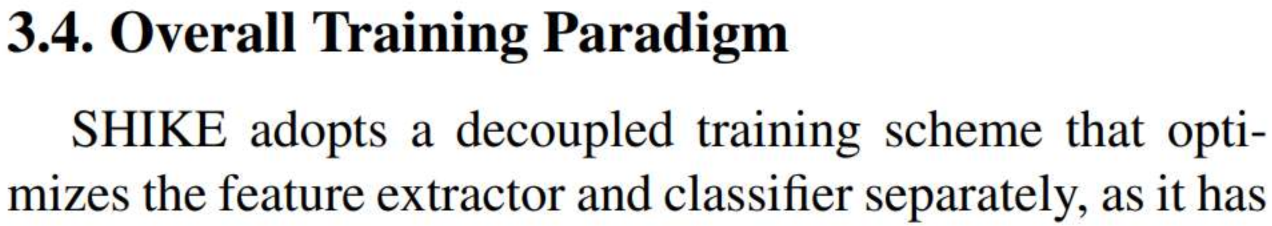
Combining the consensus hardest negative with the maximum non-target logits, we can form a set of non-target logits called grand teacher:

$$z_{I_i}^{\mathcal{T}} = \begin{cases} \bar{z}_{I_i}, & i = \mathrm{argmax}_j\{\bar{z}_{I_j}\}, \\ \hat{z}_{I_i}, & \text{otherwise}, \end{cases} \tag{7}$$

for $i = 1, ..., C-1$. Note that the grand teacher is only for suppressing the hardest negative class within non-target classes while the target class is not involved. After electing

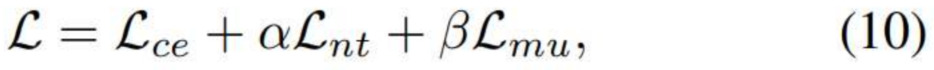
$$\widetilde{p}_{I_i}^{\mathcal{T}} = \frac{\exp(z_{I_i}^{\mathcal{T}})}{\sum_{i=1}^{C-1} \exp(z_{I_i}^{\mathcal{T}})}, \quad \widetilde{p}_{I_i}^{m} = \frac{\exp(z_{I_i}^{m})}{\sum_{i=1}^{C-1} \exp(z_{I_i}^{m})}, \tag{8}$$

$$\mathcal{L}_{nt} = \sum_{m=1}^{M} KL(\widetilde{\mathbf{p}}^{\mathcal{T}} | \widetilde{\mathbf{p}}^{m}). \tag{9}$$

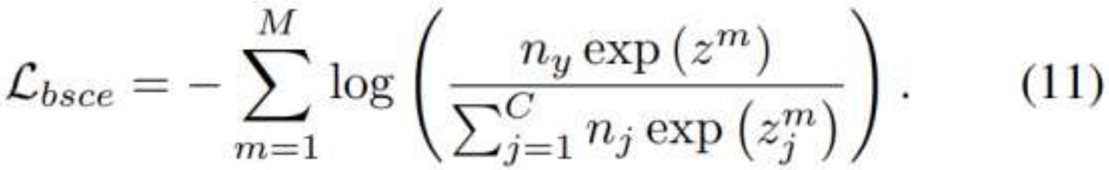


3.4. Overall Training Paradigm

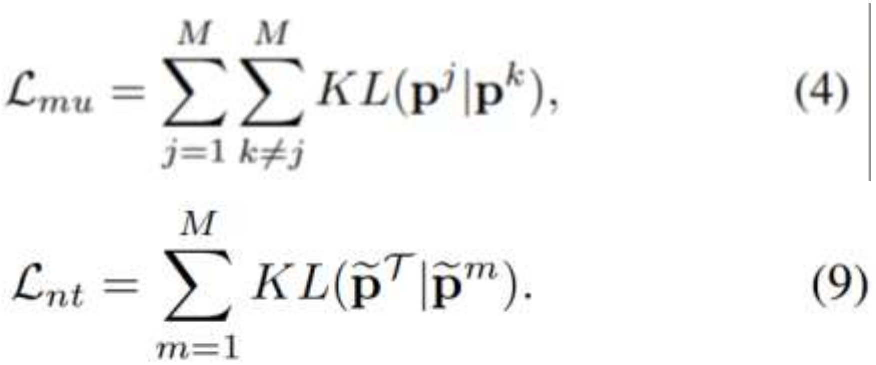
SHIKE adopts a decoupled training scheme that optimizes the feature extractor and classifier separately, as it has



$$\mathcal{L} = \mathcal{L}_{ce} + \alpha \mathcal{L}_{nt} + \beta \mathcal{L}_{mu}, \tag{10}$$



$$\mathcal{L}_{bsce} = -\sum_{m=1}^{M} \log \left(\frac{n_y \exp\left(z^m\right)}{\sum_{j=1}^{C} n_j \exp\left(z_j^m\right)} \right). \tag{11}$$



$$\mathcal{L}_{mu} = \sum_{j=1}^{M} \sum_{k \neq j}^{M} KL(\mathbf{p}^j | \mathbf{p}^k), \tag{4}$$

$$\mathcal{L}_{nt} = \sum_{m=1}^{M} KL(\widetilde{\mathbf{p}}^{\mathcal{T}} | \widetilde{\mathbf{p}}^m). \tag{9}$$

Method	Year	CIFAR100-LT	
		100	50
Single model			
Focal Loss [40]	2017	42.3	-
OLTR [41]	2019	43.4	-
LDAM-DRW [5]	2019	44.4	-
τ-norm [27]	2020	45.4	-
cRT [27]	2020	45.6	-
BALMS [49]	2020	50.7	-
LADE [23]	2021	45.4	50.5
GCL [36]	2022	48.7	53.6
Weight Balancing [2]	2022	53.6	57.7
Contrastive & Hybrid methods			
BALMS+BatchFormer [24]	2022	51.7	-
PaCo [11]	2021	51.9	56.0
PaCo+BatchFormer [24]	2022	52.4	-
BCL [72]	2022	52.0	56.6
MoE-based methods			
RIDE (3E) [60]	2021	48.3	-
ResLT (3E) [10]	2022	45.3	50.0
TLC (4E) [33]	2022	50.1	-
NCL (S) [34]	2022	53.3	56.8
NCL (3N) [34]	2022	54.2	58.2
Ours (3E)	-	**56.3**	**59.8**

Table 1. Comparison results on CIFAR100-LT with imbalance factor of 100 and 50.

Method	Year	ImageNet-LT		iNat
		R-50	RX-50	R-50
Single model				
OLTR [41]	2019	-	-	63.9
LDAM-DRW [5]	2019	-	-	68.0
cRT [27]	2020	47.3	49.6	65.2
τ-norm [27]	2020	46.7	49.4	65.6
BALMS [49]	2020	50.1	-	-
LA [44]	2021	-	-	66.4
CAM [66]	2021	-	-	70.9
GCL [36]	2022	54.9	-	72.0
Weight Balancing [2]	2022	-	53.9	70.2
Contrastive & Hybrid methods				
SSD [38]	2021	-	56.0	-
PaCo [11]	2021	57.0	58.2	73.2
BCL [72]	2022	56.0	-	71.8
RIDE+BF [24]	2022	55.7	-	74.1
BALMS+BF [24]	2022	51.1	-	-
MoE-based methods				
BBN [69]	2020	48.3	49.3	66.3
RIDE (3E) [60]	2021	55.4	56.8	72.6
ACE [4]	2021	54.7	56.6	-
NCL (S) [34]	2022	57.4	58.4	74.2
NCL (3N) [34]	2022	59.5	**60.5**	74.9
Ours (3E)	-	**59.7**	59.6	**75.4**

Table 2. Comparison results on ImageNet-LT and iNaturalist 2018 (iNat). R-50 and RX-50 are short for ResNet-50 and ResNeXt-50

Method	Year	Places-LT			
		Many	Med	Few	All
Single model					
Focal Loss [40]	2017	41.1	34.8	22.4	34.6
OLTR [41]	2019	**44.7**	37.0	25.3	35.9
NCM [27]	2020	40.4	37.1	27.3	36.4
cRT [27]	2020	42.0	37.6	24.9	36.7
τ-norm [27]	2020	37.8	40.7	31.8	37.9
LWS [27]	2020	40.6	39.1	28.6	37.6
BALMS [49]	2020	41.2	39.8	31.6	38.7
LADE [23]	2021	42.8	39.0	31.2	38.8
DisAlign [65]	2021	40.4	42.4	30.1	39.3
GCL [36]	2022	-	-	-	40.6
Contrastive & Hybrid methods					
LDAM+RSG [58]	2021	41.9	41.4	32.0	39.3
PaCo [11]	2021	37.5	**47.2**	33.9	41.2
MoE-based methods					
LFME [61]	2020	39.3	39.6	24.2	36.2
NCL (S) [34]	2022	-	-	-	41.5
NCL (3N) [34]	2022	-	-	-	41.8
Ours (3E)	-	43.6	39.2	**44.8**	**41.9**

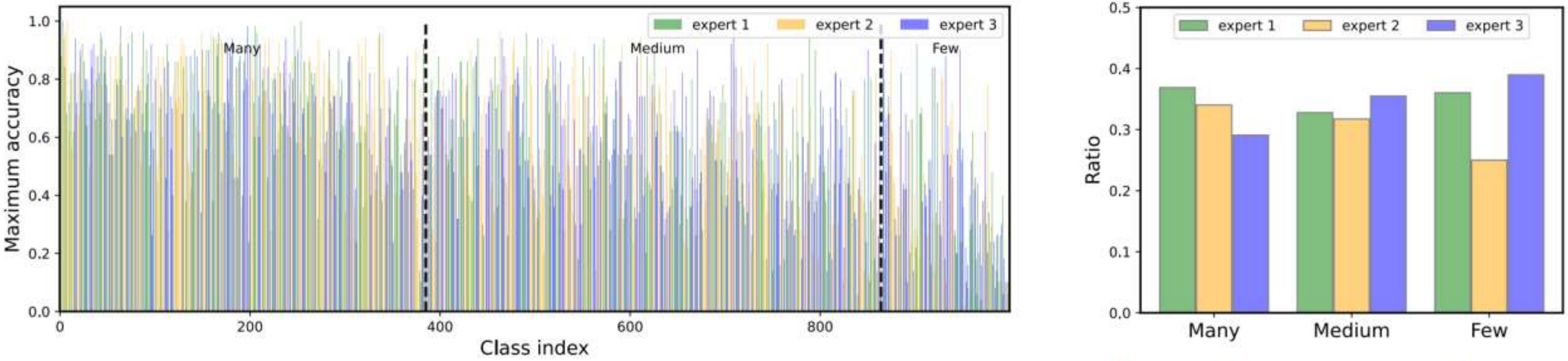
Table 3. Comparison results on Places-LT. The results are shown by different class divisions (Many, Medium, and Few) as well as the overall accuracy (All).

MoE	DKF	$\mathcal{L}_{mu}$	$\mathcal{L}_{nt}$	Acc
				50.04
✓				54.23
✓	✓			55.26
✓	✓	✓		55.59
✓	✓		✓	55.79
✓		✓	✓	54.82
✓	✓	✓	✓	**56.34**

Table 4. Ablation study on the effects of different components in the proposed SHIKE. The experiment is conducted on CIFAR100-LT with an imbalance factor of 100.

E	Depth arrangement		
1	A	B	C
	54.10	54.75	**55.84**
2	A B	B C	A C
	57.39	**58.79**	58.62
3		A B C	
		59.72	
4	A B C A	A B C B	A B C C
	59.83	**59.90**	59.77

Table 5. Ablation study on the effects of expert number in the proposed SHIKE. The experiment is conducted on ImageNet-LT.



(a) Maximum class accuracy among experts.

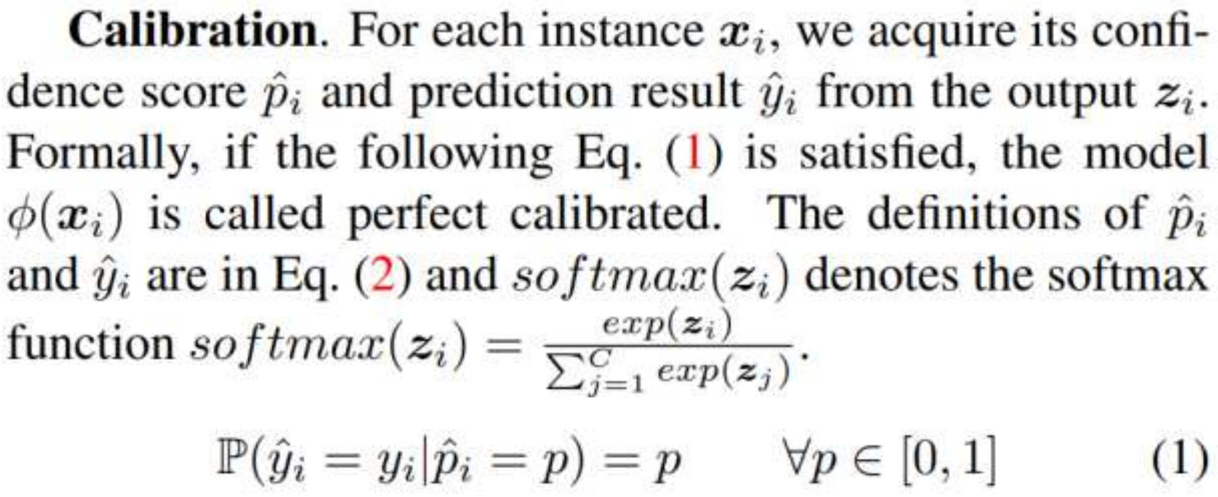
(b) Ratio of classes with the highest accuracy.

Figure 3. Preferences of different experts in SHIKE. (a) The highest accuracy among experts is shown for each class on the test set of ImageNet-LT. (b) We calculate the ratio of class numbers that each expert is most skilled at within three divisions. The experiment is conducted with ResNet-50 and the number of experts is set to 3.

Transfer Knowledge from Head to Tail: Uncertainty Calibration under Long-tailed Distribution

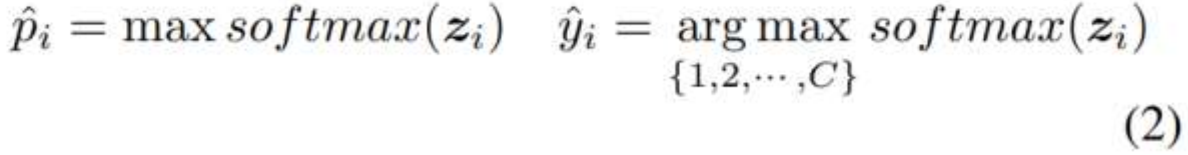
Jiahao Chen[1 2], Bing Su[1 2 †]
[1] Gaoling School of Artificial Intelligence, Renmin University of China
[2] Beijing Key Laboratory of Big Data Management and Analysis Methods
{nicelemon666, subingats}@gmail.com

CVPR 2023



Calibration. For each instance $\boldsymbol{x}_i$, we acquire its confidence score $\hat{p}_i$ and prediction result $\hat{y}_i$ from the output $\boldsymbol{z}_i$. Formally, if the following Eq. (1) is satisfied, the model $\phi(\boldsymbol{x}_i)$ is called perfect calibrated. The definitions of $\hat{p}_i$ and $\hat{y}_i$ are in Eq. (2) and $softmax(\boldsymbol{z}_i)$ denotes the softmax function $softmax(\boldsymbol{z}_i) = \frac{exp(\boldsymbol{z}_i)}{\sum_{j=1}^{C} exp(\boldsymbol{z}_j)}$.

$$\mathbb{P}(\hat{y}_i = y_i | \hat{p}_i = p) = p \qquad \forall p \in [0, 1] \tag{1}$$



$$\hat{p}_i = \max softmax(\boldsymbol{z}_i) \quad \hat{y}_i = \underset{\{1,2,\cdots,C\}}{\arg\max}\ softmax(\boldsymbol{z}_i) \tag{2}$$

Reliability Diagrams

可靠性直方图是用来描绘模型可靠性的一类直方图。 完美标定的(perfect calibrated) 模型应当是一条对角线。

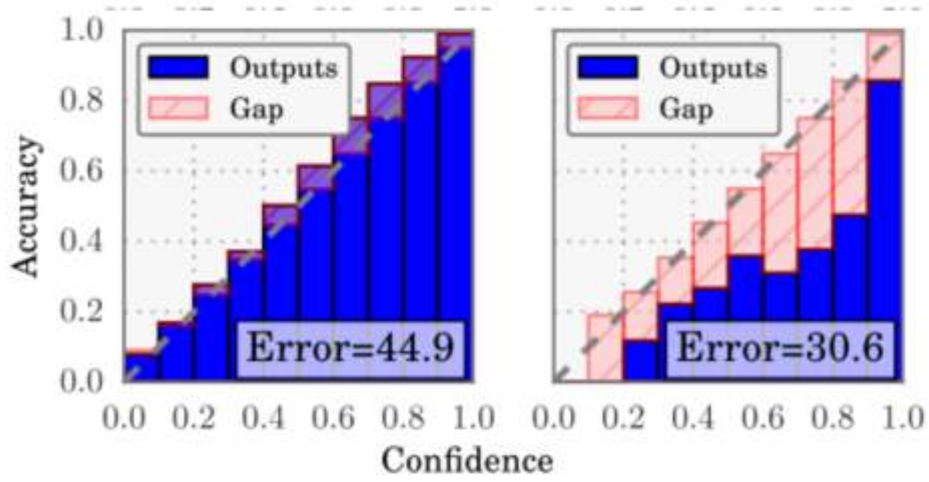
在有限的样本个数下，我们根据模型输出的prediction将样本分为M组，分别计算其Accuracy与confidence。如果我们用B_m指代第m个样本集合。那么有：

$$\mathrm{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}\left(\hat{y}_i = y_i\right)$$

其中$\hat{y}$为预测输出，y 为真实样本标签。

$$\mathrm{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i$$

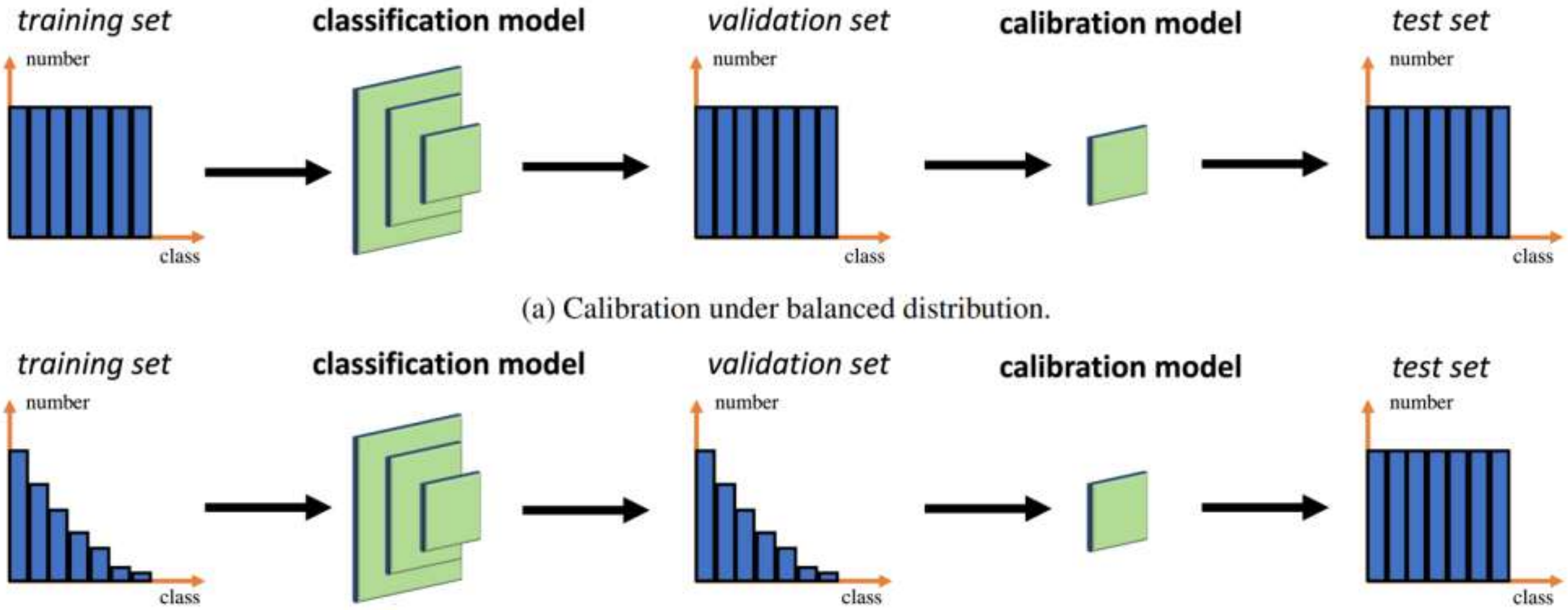
其中$\hat{p}_i$为第i个样本模型预测的信心confidence。



Reliability Diagrams

中间为gap，即confidence与精度之间的差距。完美的标定模型其可靠性图应当是一条对角线。confidence能够精确反应预测精度。

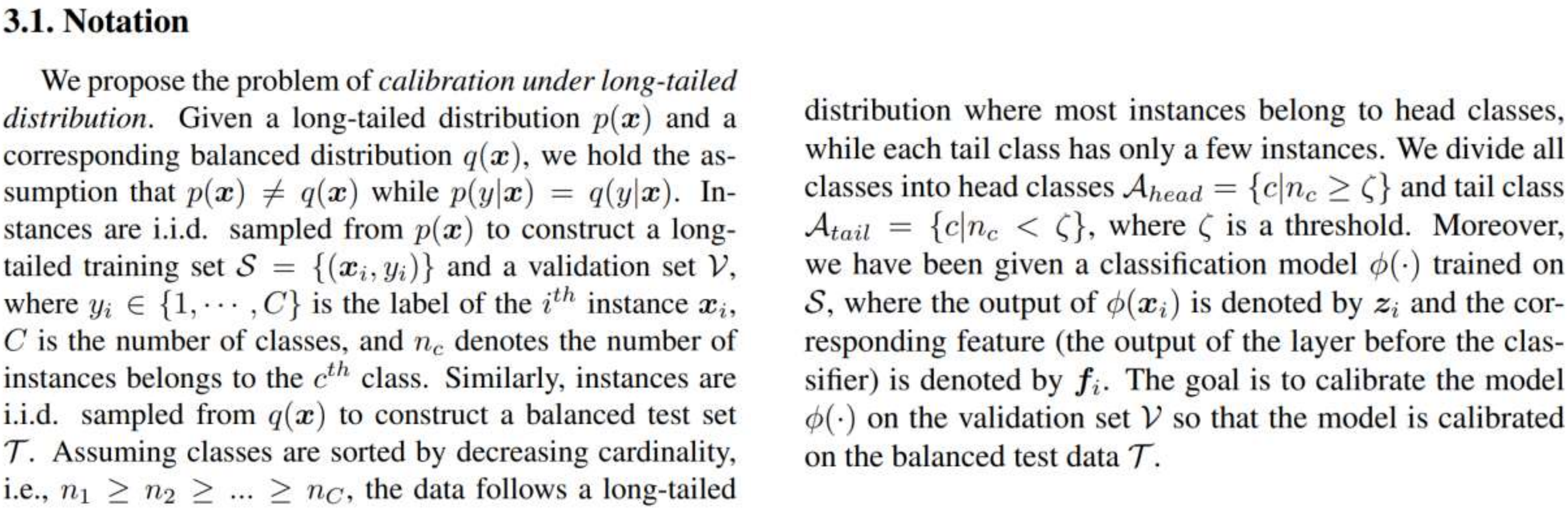
Problem: calibration under long-tailed distribution



(a) Calibration under balanced distribution.

(b) Calibration under long-tailed distribution.

Figure 1. The difference between calibration under balanced distribution and calibration under long-tailed distribution. (a) The classification model and the calibration model are trained on the balanced training and validation sets, respectively, and the test set is balanced. (b) The classification model and the calibration model are trained on the long-tailed training and validation sets, while the test set is balanced.



3.1. Notation

We propose the problem of *calibration under long-tailed distribution*. Given a long-tailed distribution $p(\boldsymbol{x})$ and a corresponding balanced distribution $q(\boldsymbol{x})$, we hold the assumption that $p(\boldsymbol{x}) \neq q(\boldsymbol{x})$ while $p(y|\boldsymbol{x}) = q(y|\boldsymbol{x})$. Instances are i.i.d. sampled from $p(\boldsymbol{x})$ to construct a long-tailed training set $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}$ and a validation set $\mathcal{V}$, where $y_i \in \{1, \cdots, C\}$ is the label of the i^{th} instance $\boldsymbol{x}_i$, C is the number of classes, and n_c denotes the number of instances belongs to the c^{th} class. Similarly, instances are i.i.d. sampled from $q(\boldsymbol{x})$ to construct a balanced test set $\mathcal{T}$. Assuming classes are sorted by decreasing cardinality, i.e., $n_1 \geq n_2 \geq ... \geq n_C$, the data follows a long-tailed distribution where most instances belong to head classes, while each tail class has only a few instances. We divide all classes into head classes $\mathcal{A}_{head} = \{c|n_c \geq \zeta\}$ and tail class $\mathcal{A}_{tail} = \{c|n_c < \zeta\}$, where ζ is a threshold. Moreover, we have been given a classification model $\phi(\cdot)$ trained on $\mathcal{S}$, where the output of $\phi(\boldsymbol{x}_i)$ is denoted by $\boldsymbol{z}_i$ and the corresponding feature (the output of the layer before the classifier) is denoted by $\boldsymbol{f}_i$. The goal is to calibrate the model $\phi(\cdot)$ on the validation set $\mathcal{V}$ so that the model is calibrated on the balanced test data $\mathcal{T}$.

Temperature scaling. As shown in Eq. (3), temperature scaling [9] fits a single parameter T from the validation set and applies it to other test sets.

$$T^* = \arg\min_T \mathbb{E}_p[\mathcal{L}(s(\boldsymbol{z}_i/T), y_i)] \quad (3)$$

Similar to the training classification task, the loss function $\mathcal{L}(\cdot)$ for calibrating the temperature is the Cross Entropy loss. Since the validation set also follows long-tailed distribution while the test set does not, the learned parameter T is difficult to generalize well to the test set.

Expected Calibration Error(ECE) 期望标定误差

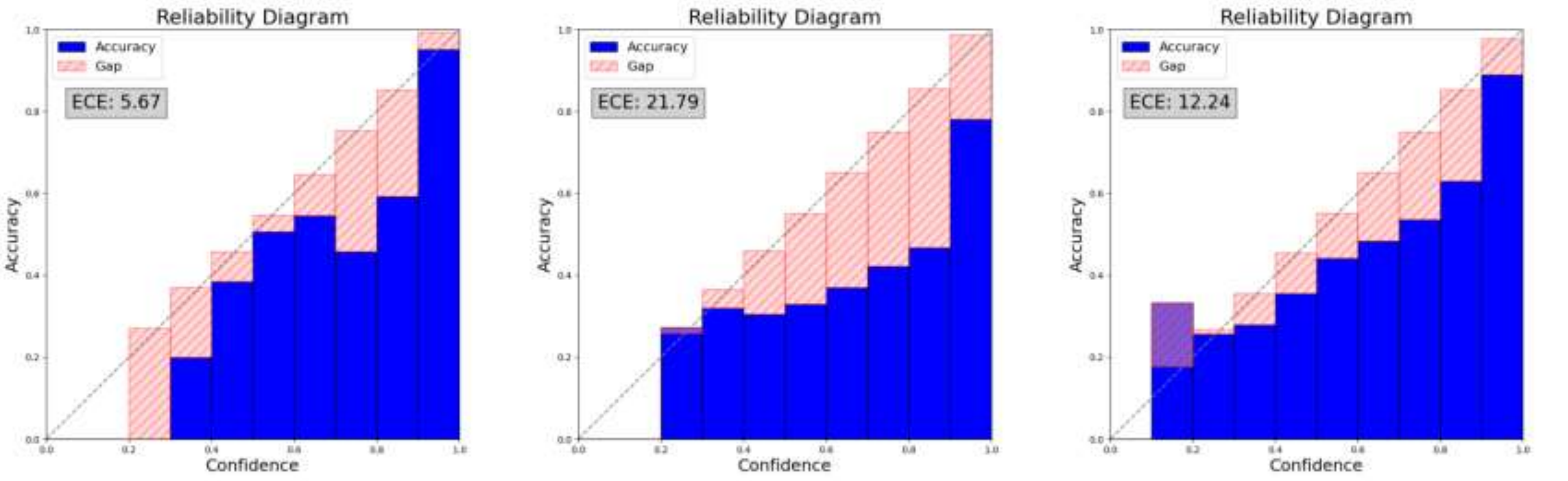
$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)|$$

可用于度量标定。在图中表现为gap的均值。

Maximum Calibration Error(MCE) 最大标定误差

$$\mathrm{MCE} = \max_{m \in \{1,\ldots,M\}} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)|$$

MCE为最大标定误差，即图中最大的一个gap。



(a) Validation set (b) Test set (c) Temperature scaling

We propose the problem of *calibration under long-tailed distribution*. Given a long-tailed distribution $p(\boldsymbol{x})$ and a corresponding balanced distribution $q(\boldsymbol{x})$, we hold the assumption that $p(\boldsymbol{x}) \neq q(\boldsymbol{x})$ while $p(y|\boldsymbol{x}) = q(y|\boldsymbol{x})$. In-

3.3. Calibration under long-tailed distribution

To tackle the generalization issue of the original temperature scaling in calibration under long-tailed distribution, we propose our knowledge-transferring-based temperature scaling method to achieve cross-distribution generalization. The calibration loss on the balanced target distribution $q(\boldsymbol{x})$ can be reformulated as the weighted calibration error of the source distribution $p(\boldsymbol{x})$:

$$\begin{aligned}
\mathbb{E}_q[\mathcal{L}(s(\boldsymbol{z}_i/T), y_i)] &= \int_q q(\boldsymbol{x}_i)\mathcal{L}(s(\boldsymbol{z}_i/T), y_i)dx \\
&= \int_p \frac{q(\boldsymbol{x}_i)}{p(\boldsymbol{x}_i)} p(\boldsymbol{x}_i)\mathcal{L}(s(\boldsymbol{z}_i/T), y_i)dx \\
&= \mathbb{E}_p[w(\boldsymbol{x}_i)\mathcal{L}(s(\boldsymbol{z}_i/T), y_i)]
\end{aligned} \tag{4}$$

As shown in Eq. (4), we can acquire the target distribution error $\mathbb{E}_q[\mathcal{L}(s(\boldsymbol{z}_i/T), y_i)]$ by estimating the ratio of probabilities $w(\boldsymbol{x}) = q(\boldsymbol{x})/p(\boldsymbol{x})$ for each instance. Domain adaptation calibration like TransCal [40] utilizes LogReg [2, 32] to estimate the ratio of density. It estimates the density by training a logistic regression classifier that realizes binary classification of source and target domains. Such methods cannot be used for the long-tailed calibration problem since the balanced distribution of test data is unknown and thus binary classification cannot be applied.

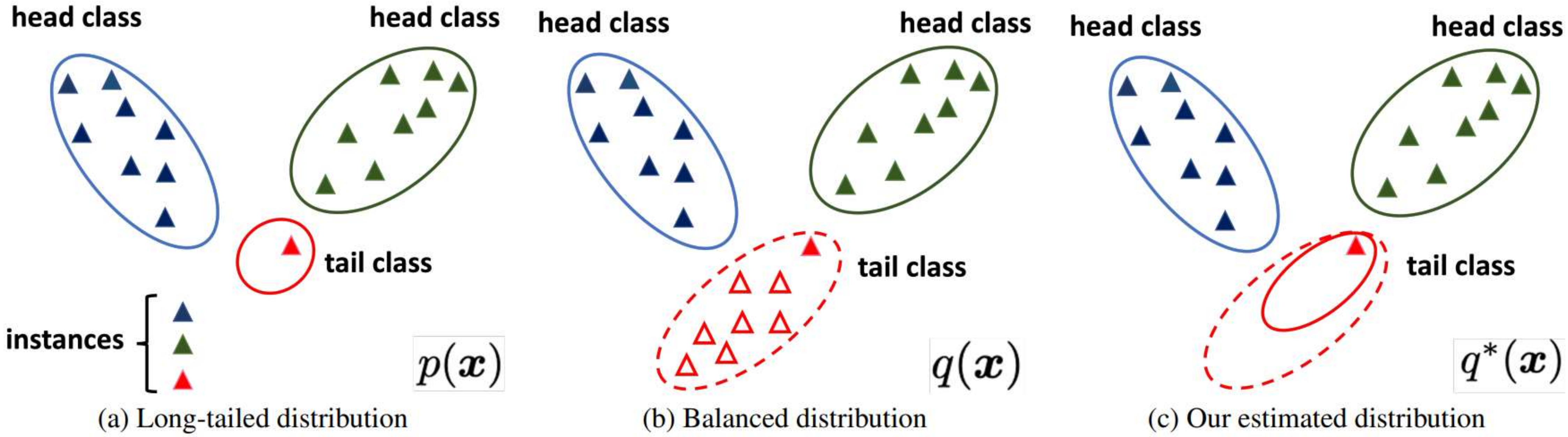


Figure 3. (a) The long-tailed distribution $p(\boldsymbol{x})$. (b) The balanced distribution $q(\boldsymbol{x})$. Compared with (a), the distributions of head classes are the same, while the distributions of tail classes are not. (c) Our estimated distribution $q^*(\boldsymbol{x})$. With the help of head classes, we can estimate the distribution of tail classes and acquire their density.

For each class $c \in \{1, 2, \cdots, C\}$, the corresponding distribution is $p_c(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$, where the mean $\boldsymbol{\mu}_c$ and the variance $\boldsymbol{\Sigma}_c$ are calculated by the set of output features belonging to class c in the training set. When $c \in \mathcal{A}_{tail}$, its distribution $p_c(\boldsymbol{x})$ is not reliable due to the limited instances. It is crucial to estimate the probability density under balanced distribution. Since there exists common information between head classes and tail classes [21–23], it is rational to transfer knowledge from all head classes to the c^{th} tail class and recover its balanced distribution to the utmost. Normally, the more similar two classes are, the more information they share. We measure the similarity of the

two classes by the Wasserstein distance to consider the first-order and second-order statistics. We construct a distance vector $\boldsymbol{d}_c \in \mathbb{R}^{|\mathcal{A}_{head}|}$, where $\boldsymbol{d}_c^k$ is the k^{th} element of $\boldsymbol{d}_c$ and denotes the similarity between class c and head class k.

$$\boldsymbol{d}_c^k = Wasserstein(p_c(\boldsymbol{x}), p_k(\boldsymbol{x})) \tag{5}$$

Following the attention mechanism [38], we calculate the attention score $\boldsymbol{s}_c$ from the similarities as follows,

$$\boldsymbol{s}_c = softmax(-\frac{\boldsymbol{d}_c}{\sqrt{dim(\boldsymbol{f})}}) \tag{6}$$

where the $dim()$ function acquires the feature dimension. A head class will be assigned a large attention score if its distribution is similar to the distribution of tail class c.

We calibrate the distribution of each tail class by transferring knowledge from all head classes based on $\boldsymbol{s}_c$. Specifically, for the c^{th} tail class, the statistics of its calibrated distribution are estimated as follows.

$$\begin{aligned} \boldsymbol{\mu}_{c^*} &= \alpha\boldsymbol{\mu}_c + (1-\alpha)\sum_{k\in\mathcal{A}_{head}} \boldsymbol{s}_c^k \boldsymbol{\mu}_k \\ \sqrt{\boldsymbol{\Sigma}}_{c^*} &= \alpha\sqrt{\boldsymbol{\Sigma}}_c + (1-\alpha)\sum_{k\in\mathcal{A}_{head}} \boldsymbol{s}_c^k \sqrt{\boldsymbol{\Sigma}}_k \end{aligned} \tag{7}$$

The estimated distribution $\mathcal{N}(\boldsymbol{\mu}_{c^*}, \boldsymbol{\Sigma}_{c^*})$ contains the information of all head classes according to their similarity scores in $\boldsymbol{d}_c$, where α is a hyper-parameter.

For each instance $\boldsymbol{x}_i$ in tail classes, we can then acquire its probability under the estimated distribution $q^*_{y_i}(\boldsymbol{x}_i) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_{y_i^*}, \boldsymbol{\Sigma}_{y_i^*})$. Based on the estimated $q^*_{y_i}(\boldsymbol{x}_i)$, the importance weight is defined in Eq. (8).

$$w^*(\boldsymbol{x}_i) = \begin{cases} 1 & y_i \in \mathcal{A}_{head} \\ min(max(\frac{q^*_{y_i}(\boldsymbol{x}_i)}{p_{y_i}(\boldsymbol{x}_i)}, \eta_1), \eta_2) & y_i \in \mathcal{A}_{tail} \end{cases} \tag{8}$$

For each instance in head classes, the importance weight equals 1 since head classes in the two distributions are the same. For each instance in tail classes, the importance weight equals $q^*_{y_i}(\boldsymbol{x}_i)/p_{y_i}(\boldsymbol{x}_i)$. In practice, we restrict the value of the weight from η_1 to η_2 to avoid abnormal values. Empirically, we set $\eta_1 = 0.3$ and $\eta_2 = 5.0$.

By using the importance weight to bridge the training long-tailed distribution and the test balanced distribution, we learn the temperature T in the final softmax layer on the validation set to calibrate the classification confidence. The final optimization function is shown in Eq. (9).

$$T^* = \arg\min_T \mathbb{E}_p[w^*(\boldsymbol{x}_i)\mathcal{L}(s(\boldsymbol{z}_i/T), y_i)] \tag{9}$$

3. Training procedures

Our calibration method consists of four procedures. 1. Estimate the feature distribution of each class on the validation set: the effect is to obtain the statistics for both head and tail classes for similarity calculation and knowledge transfer. 2. Calculate attentions between head and tail classes based on the Wasserstein distance between their distributions: the effect is to determine how much knowledge is transferred from each head class to a tail class in a principled manner. 3. Estimate importance weights with the calibrated distributions: the effect is to compensate for tail classes by reweighting their samples. 4. Learn the temperature T with the importance weights: the effect is to scale prediction confidence scores for calibration under long-tailed distribution.

CIFAR-10-LT. CIFAR-10-LT [3] is simulated from balanced CIFAR-10 [17]. We conduct experiments with different imbalance factors (IF) and generate three imbalanced datasets with IF=100, IF=50, and IF=10, respectively. For each dataset, we randomly split 80% instances as the training set and 20% as the validation set. For comparison, we use four test sets: (1) original CIFAR-10 test set, (2) CIFAR-10.1 [33], (3) CIFAR10.1-C [12]: 95 synthetics datasets generated on CIFAR-10.1 with different transformations, (4) CIFAR-F [36]: 20 real-word test sets collected from Flickr. **MNIST-LT**. MNIST-LT is simulated from MNIST [20]. Similar to CIFAR-10-LT, we generate three imbalanced datasets with IF=100, IF=50, and IF=10, respectively. For comparison, we use four test sets: (1) original MNIST test set, (2) SVHN [26], (3) USPS [14], (4) Digital-S [36]: 5 test sets that are searched from Shutterstock based on different options of color. Note that the original MNIST test set is slightly imbalanced, which is closer to reality. **CIFAR-100-LT**. CIFAR-100-LT [3] is generated from the CIFAR-100 dataset. We generate imbalanced datasets with IF=10 and conduct experiments on the original CIFAR-100 test set. **ImageNet-LT**. ImageNet-LT [24] is simulated from ImageNet [7]. We merge the long-tailed training set and balanced validation set from the original ImageNet-LT. Following the principle of CIFAR-10-LT, we generate a long-tailed training set and a long-tailed validation set. We conduct experiments on a balanced test set. More details are presented in Appendix 4.

IF	**Dataset**	**Method**								
		Base	TS	ETS	TS-IR	IR	IROvA	SBC	GPC	Ours
IF=100	CIFAR-10	21.79	12.24	12.16	11.64	12.36	13.36	12.13	11.65	**9.84**
	CIFAR-10.1	28.97	16.75	16.70	16.65	17.13	17.93	16.78	15.71	**13.86**
	CIFAR-10.1-C	58.22	43.01	43.00	43.05	43.34	43.83	42.53	41.98	**39.58**
	CIFAR-F	29.22	15.27	15.24	15.52	15.75	16.23	15.45	14.18	**12.15**
IF=50	CIFAR-10	17.36	7.65	8.04	8.22	9.75	9.45	7.55	7.78	**3.99**
	CIFAR-10.1	22.79	10.36	10.99	11.72	13.35	12.70	10.32	10.82	**5.74**
	CIFAR-10.1-C	55.52	38.66	39.9	40.16	41.58	40.76	38.94	39.39	**33.09**
	CIFAR-F	25.37	11.30	12.21	12.67	14.39	13.37	11.4	11.76	**6.64**
IF=10	CIFAR-10	8.39	2.23	1.64	2.03	2.29	2.42	2.49	2.01	**1.00**
	CIFAR-10.1	13.80	4.87	4.25	4.54	5.38	5.23	5.63	4.66	**3.95**
	CIFAR-10.1-C	48.31	32.77	31.07	32.11	32.29	31.94	33.16	31.37	**29.98**
	CIFAR-F	19.73	8.15	6.80	8.42	8.97	8.13	8.54	7.10	**5.97**

Table 1. The ECE (%) on CIFAR-10-LT.

IF	**Dataset**	**Method**								
		Base	TS	ETS	TS-IR	IR	IROvA	SBC	GPC	Ours
IF=100	MNIST	2.52	1.27	1.84	2.82	2.84	1.84	1.92	1.76	**1.08**
	SVHN	16.06	7.20	11.62	21.25	22.18	14.93	9.59	13.67	**6.09**
	USPS	15.00	9.52	12.25	13.25	13.62	10.58	10.10	11.44	**8.40**
	Digital-S	32.10	22.13	27.35	30.13	31.01	27.48	23.34	27.60	**20.28**
IF=50	MNIST	1.12	0.85	1.14	1.53	1.54	1.02	1.01	1.12	**0.79**
	SVHN	**2.32**	3.95	3.33	11.42	12.15	2.63	9.43	**2.32**	4.53
	USPS	11.21	8.14	12.81	11.89	11.91	10.54	8.57	11.21	**8.02**
	Digital-S	15.22	10.81	17.81	20.96	21.81	13.64	16.74	15.18	**10.34**
IF=10	MNIST	0.56	0.23	**0.21**	0.50	0.52	0.23	0.25	0.41	0.36
	SVHN	5.75	6.76	6.94	8.10	**4.51**	5.31	7.00	5.31	7.43
	USPS	8.29	4.81	4.60	6.59	6.98	4.76	5.12	5.88	**4.55**
	Digital-S	13.55	8.21	8.09	15.37	13.34	8.31	7.67	8.24	**7.37**

Table 2. The ECE (%) on MNIST-LT.

Model	Dataset	Method								
		Base	TS	ETS	TS-IR	IR	IROvA	SBC	GPC	Ours
ResNet-32	CIFAR-100	20.38	2.50	2.10	6.07	9.35	5.92	6.74	3.27	**1.50**
DenseNet-40	CIFAR-100	16.00	3.43	2.51	5.57	8.42	5.76	5.96	2.73	**2.37**
VGG-19	CIFAR-100	27.86	3.81	2.36	6.35	10.35	6.66	8.03	3.82	**1.99**

Table 3. The ECE (%) on CIFAR-100-LT.

Model	Dataset	Method								
		Base	TS	ETS	TS-IR	IR	IROvA	SBC	GPC	Ours
ResNet-50	ImageNet	10.18	6.72	6.06	10.23	11.15	7.63	9.12	5.46	**3.45**

Table 4. The ECE (%) on ImageNet-LT.

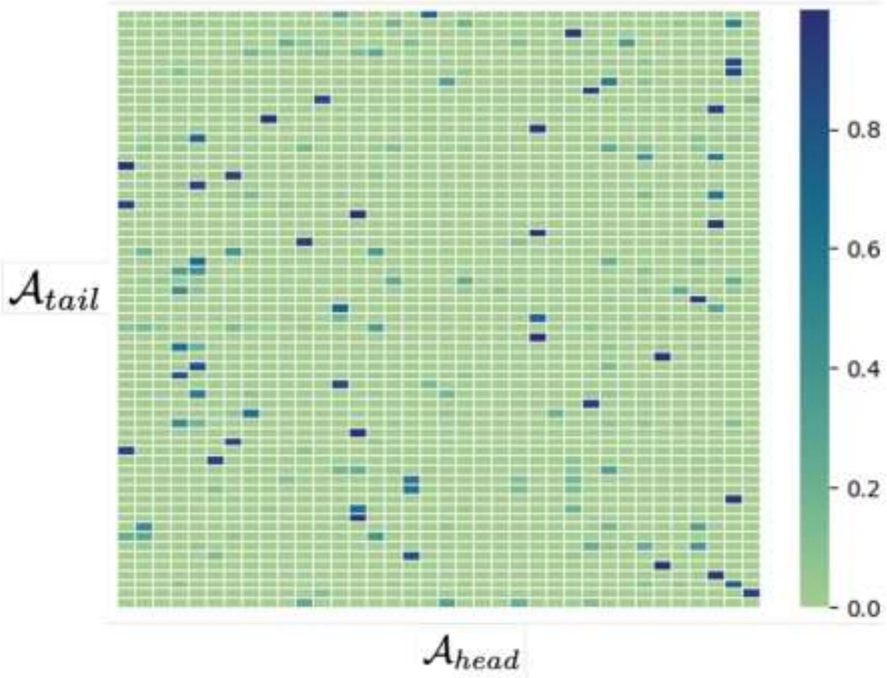
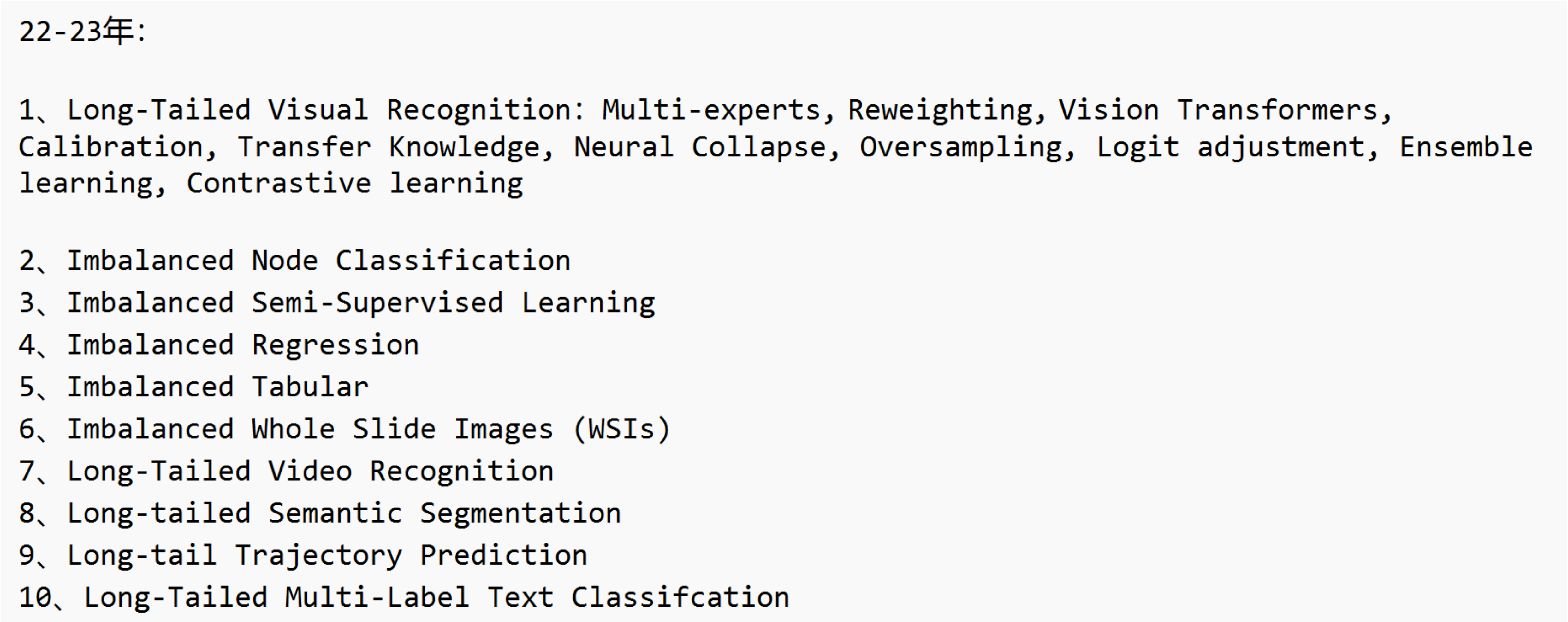


Figure 5. Experiment on CIFAR-100-LT. The shape of the matrix is $|\mathcal{A}_{tail}| \times |\mathcal{A}_{head}|$. Each row denotes the attention vector $\boldsymbol{s}$.



22-23年:

1、Long-Tailed Visual Recognition: Multi-experts, Reweighting, Vision Transformers, Calibration, Transfer Knowledge, Neural Collapse, Oversampling, Logit adjustment, Ensemble learning, Contrastive learning

2、Imbalanced Node Classification
3、Imbalanced Semi-Supervised Learning
4、Imbalanced Regression
5、Imbalanced Tabular
6、Imbalanced Whole Slide Images (WSIs)
7、Long-Tailed Video Recognition
8、Long-tailed Semantic Segmentation
9、Long-tail Trajectory Prediction
10、Long-Tailed Multi-Label Text Classifcation

THANKS