UNCERTAINTY CALIBRATION VIA KNOWLEDGE FLOW UNDER LONG-TAILED DISTRIBUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

How to estimate the uncertainty of a given model is a crucial problem. Current calibration techniques treat different classes equally and thus implicitly assume that the distribution of training data is balanced, but ignore the fact that real-world data often follows a long-tailed distribution. In this paper, we explore the problem of calibrating the model trained from a long-tailed distribution. Due to the difference between the imbalanced training distribution and balanced test distribution, existing calibration methods such as temperature scaling can not generalize well to this problem. Specific calibration methods for domain adaptation are also not applicable because they rely on unlabeled target domain instances which are not available. Models trained from a long-tailed distribution tend to be more over-confident to head classes. To this end, we propose a novel knowledge flow based calibration method by estimating the importance weight for samples of tail classes to realize long-tailed calibration. Our method models the distribution of each class as a Gaussian distribution and view the source statistics of head classes as a prior to calibrate the target distributions of tail classes. We transfer knowledge from head classes to get the target probability density of tail classes. The importance weight is estimated by the ratio of the target probability density over the source probability density. Extensive experiments on CIFAR-10-LT, MNIST-LT, CIFAR-100-LT, and ImageNet-LT datasets demonstrate the effectiveness of our method.

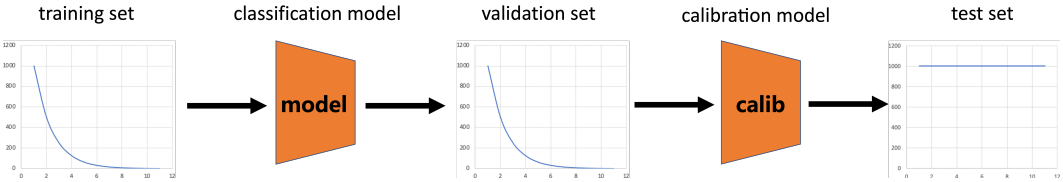
1 INTRODUCTION

With the development of deep neural networks, great progress has been made in image classification. In addition to performance, the uncertainty estimate of a given model is also receiving increasing attention, as the confidence of a model is expected to accurately reflect its performance. A model is called *perfect calibrated* if the predictive confidence of the model represents a good approximation of its actual probability of correctness (Guo et al., 2017). Model calibration is particularly important in safety-critical applications, such as autonomous driving, medical diagnosis, and robotics (Amodei et al., 2016). For example, if a prediction with low confidence is more likely to be wrong, we can take countermeasures to avoid unknown risks.

Most existing calibration techniques assume that the distribution of training data is balanced, i.e., each class has a similar number of training instances, so that each class is treated equally. As shown in Fig.1, the traditional calibration pipeline uses a balanced training set to train the classification model and a balanced validation set to obtain the calibration model, respectively. The target test set is in the same distribution as the training/validation set. However, data in the real-world often follows a long-tailed distribution, i.e., a few dominant classes occupy most of the instances, while much fewer examples are available for most other classes (Kang et al., 2020; Liu et al., 2019; Cui et al., 2019). When tested on balanced test data, classification models trained from the training set with a long-tailed distribution are naturally more over-confident to head classes. Only imbalanced validation set with the same long-tailed distribution is available for calibrating such models since the validation set is often randomly divided from the training set.

Due to the different distribution between the imbalanced training data and the balanced test data (Jamal et al., 2020), it is difficult for traditional calibration techniques to achieve balanced calibration among head classes and tail classes with different levels of confidence estimations. For instance, temperature scaling (Guo et al., 2017) with the temperature learned on a validation set obtains de-

(a) Calibration under balanced distribution.



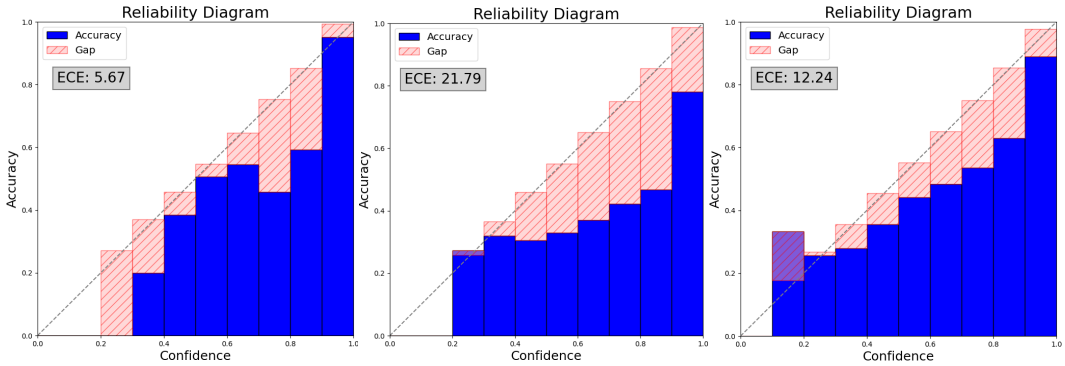
(b) Calibration under long-tailed distribution.

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

graded performance on the test set if the two sets are in different distribution (Tomani et al., 2021; Pampari & Ermon, 2020). As shown in Fig.2, a balanced test set suffers heavier overconfidence compared with a long-tailed validation set. Although temperature scaling can relieve such phenomenon, there still exists overconfidence after calibration. Domain adaptation calibration methods (Pampari & Ermon, 2020; Wang et al., 2020) aim to generalize calibration across domains under a covariate shift condition but they utilize unlabeled target domain instances. Similarly, domain generalization calibration method (Gong et al., 2021) uses support set to bridge the gap between the source domain and the target domain, which also rely on extra instances. These methods cannot apply to the long-tailed calibration task directly since instances from the balanced test domain are not available.

In this paper, we investigate the problem of *calibration under long-tailed distribution*. Since the distribution between the imbalanced validation set and the balanced target set is different, we utilize an importance weight strategy to alleviate the unreliable calibration for tail classes. The weight of each instance is the ratio between the probability density of the source domain and the target domain. We explicitly model the distribution of each class as a Gaussian distribution. Different from the source distribution, the target balanced distribution cannot be estimated directly. Since there exists common information between head classes and tail classes (Liu et al., 2020), we transfer knowledge from head classes to estimate the target probability density. For each instance in a tail class, we select the most similar head classes and acquire the corresponding distribution as a prior. Then we combine the prior distribution and self-distribution of the tail class to obtain the estimated density. Finally, we calibrate the model with the importance weights. Our contributions are summarized as:

- 1) We explore the problem of calibration under long-tailed distributions, which has important practical implications but is rarely studied. We apply the importance weight strategy to enhance the estimation of tail classes for more accurate calibration.
- 2) We propose an importance weight estimation method by viewing distributions of head classes as prior for distributions of tail classes. For each instance in a tail class, our method estimates its probability density from the distribution calibrated by head classes and calculates the importance weight to realize balanced calibration.
- 3) We conduct extensive experiments on the CIFAR-10-LT, CIFAR-100-LT (Cao et al., 2019), MNIST-LT (LeCun et al., 1998), ImageNet-LT (Liu et al., 2019) datasets and the results demonstrate the effectiveness of our method.



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

Figure 2: The reliability diagrams of (a) the validation set before calibration, (b) the test set before calibration, and (c) the test set after calibration with temperature scaling.

2 Related Work

Post-processing calibration. Current calibration techniques can be roughly divided into post-processing methods and regularization methods (Hebbalaguppe et al., 2022; Cheng & Vasconcelos, 2022). Post-processing methods focus on learning a re-calibration function on a given model. Platt scaling (Platt et al., 1999) transforms outputs of a classification model into a probability distribution over classes. It can solve the calibration of non-probabilistic methods like SVM (Cortes & Vapnik, 1995). Temperature scaling (Guo et al., 2017) extends Platt scaling and is applied to multi-class classification problems. It optimizes a parameter T to re-scale the output logits of a given model. Non-parametric isotonic regression (Zadrozny & Elkan, 2002) learns a piece-wise constant function that minimizes the residual between the calibrated prediction and the labels. In (Zhang et al., 2020), three properties, accuracy-preserving, data-efficient, and expressive of uncertainty calibration are proposed. Experiments show that a combination of the non-parametric method and the parametric method can achieve better results.

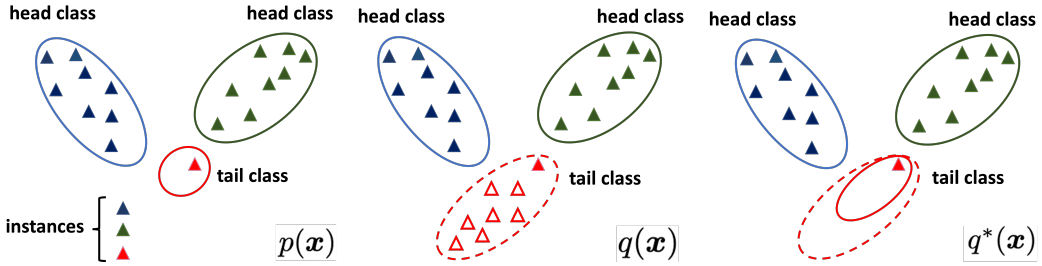
Domain shift calibration. The most common case is that the validation and test sets are in different domains (Wald et al., 2021). The re-calibration function learned by the validation domain cannot be generalized to the test domain. CPCS (Park et al., 2020) utilizes importance weight to correct for the shift from the training domain to the target domain and achieves good calibration for the domain adaptation model. TransCal (Wang et al., 2020) achieves more accurate calibration with lower bias and variance in a unified hyperparameter-free optimization framework. In (Gong et al., 2021), a support set is applied to bridge the gap between the source domain and a target domain, and three calibration strategies are proposed to achieve calibration for domain generalization. In (Tomani et al., 2021), current techniques have demonstrated that overconfidence is still existing under domain shift and a simple strategy where perturbations are applied to samples in the validation set before performing the post-hoc calibration step is proposed.

Although long-tailed distribution calibration also suffers the domain shift problem, it is different from domain shift calibration since unlabeled target domain instances or plenty of data to constitute a support set are not available for calibration. Therefore, we employ the importance weight and estimate the target probability density by utilizing the inherent property of long-tailed distribution.

3 Method

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



(a) Long-tailed distribution (b) Balanced distribution (c) Our estimated distribution

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 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.

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 $\mathcal{T}$. Without loss of generality, we assume that the classes are sorted by cardinality in decreasing order, i.e., $n_1 \geq n_2 \geq ... \geq n_C$. The data obeys the long-tailed distribution, i.e., most instances belong to only a few head classes, while each of the other tail classes only has a few instances. Moreover, we have 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 $\mathcal{T}$.

3.2 POST-PROCESSING CALIBRATION UNDER LONG-TAILED DISTRIBUTION

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

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

$$\hat{p}_i = \max s(\boldsymbol{z}_i) \quad \hat{y}_i = \underset{\{1,2,\cdots,C\}}{\arg\max}\ s(\boldsymbol{z}_i) \quad s(\boldsymbol{z}_i) = \frac{exp(\boldsymbol{z}_i)}{\sum_{j=1}^{C} exp(\boldsymbol{z}_j)} \tag{2}$$

This formulation means that for example 20% of all predictions with a confidence score of 80% should be false.

Temperature scaling. As shown in Eq. 3, temperature scaling (Guo et al., 2017) fits a single parameter T from the validation set and applies it to other test sets.

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

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.

Knowledge flow based temperature scaling. To tackle the generalization issue of the original temperature scaling in long-tailed distribution calibration, we propose our knowledge flow based temperature scaling method to achieve cross-distribution generalization. The calibration loss on the balanced target distribution $q(\boldsymbol{x})$ can be reformulated as 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 probability $w(\boldsymbol{x}) = q(\boldsymbol{x})/p(\boldsymbol{x})$ for each instance. Domain adaptation calibration like TransCal (Wang et al., 2020) utilizes LogReg (Qin, 1998; Bickel & Scheffer, 2006) to estimate the ratio of density. It estimates the density by training a logistic regression classifier that realizes binary classification of source and target domain. Such methods cannot be directly used for the long-tailed calibration problem since the balanced distribution of test data is unknown and thus binary classification cannot be applied.

We model the distribution $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ as mixtures of Gaussian distributions, respectively, by modeling each class as a Gaussian distribution. As shown in Fig.3, because head classes have plenty of instances in both $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$, the distributions of head classes in $q(\boldsymbol{x})$ can be viewed as the same as those in $p(\boldsymbol{x})$, which can be easily estimated from the training set. However, each tail class only has a few instances in $p(\boldsymbol{x})$ while sufficient instances are available in $q(\boldsymbol{x})$, the distributions of tail classes are different in $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$. Since it is difficult to acquire the balanced distribution $q(\boldsymbol{x})$, we constitute the estimated distribution $q^*(\boldsymbol{x})$ to approach the truth distribution $q(\boldsymbol{x})$, where the key is to approximate the probability density value of each instance in tail classes under $q^*(\boldsymbol{x})$ from the estimated Gaussian distributions of tail classes in $p(\boldsymbol{x})$.

For an instance $\boldsymbol{x}_i$ of the tail class y_i in $p(\boldsymbol{x})$, its output feature $\boldsymbol{f}_i \sim \mathcal{N}(\boldsymbol{\mu}_{y_i}, \boldsymbol{\sigma}^2_{y_i})$, where $\boldsymbol{\mu}_{y_i}$ and $\boldsymbol{\sigma}^2_{y_i}$ are calculated by the set of features belongs to class y_i in the training set. To estimate the probability density of $\boldsymbol{x}_i$ under $q^*(\boldsymbol{x})$, we transfer knowledge from the most similar head class y_j to $\boldsymbol{x}_i$ by the probability $\boldsymbol{z}_i$ after softmax, since there exists some common information between tail classes and head classes (Liu et al., 2020). We view the distribution $\mathcal{N}(\boldsymbol{\mu}_{y_j}, \boldsymbol{\sigma}^2_{y_j})$ as a prior to approximate the distribution of the tail class y_i under $q^*(\boldsymbol{x})$ as follows:

$$\boldsymbol{\mu}_{y_i^*} = \alpha\boldsymbol{\mu}_{y_i} + (1-\alpha)\boldsymbol{\mu}_{y_j} \qquad \boldsymbol{\sigma}_{y_i^*} = \alpha\boldsymbol{\sigma}_{y_i} + (1-\alpha)\boldsymbol{\sigma}_{y_j} \tag{5}$$

As shown in Eq.5, the synthetic distribution $\mathcal{N}(\boldsymbol{\mu}_{y_i^*}, \boldsymbol{\sigma}^2_{y_i^*})$ contains the information of two different distributions, where α is a hyper-parameter. Then, we can obtain $q^*(\boldsymbol{x}_i)$ by calculating the probability density value under $\mathcal{N}(\boldsymbol{\mu}_{y_i^*}, \boldsymbol{\sigma}^2_{y_i^*})$.

Based on the estimated $q^*(\boldsymbol{x}_i)$, the importance weight is defined in Eq.6.

$$w^*(\boldsymbol{x}_i) = \begin{cases} 1 & y_i \text{ belongs to head class} \\ min(max(q^*(\boldsymbol{x}_i)/p(\boldsymbol{x}_i), 0.3), 5.0) & y_i \text{ belongs to tail class} \end{cases} \tag{6}$$

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 to $q^*(\boldsymbol{x}_i)/p(\boldsymbol{x}_i)$. In practice, we restrict the value of the weight from 0.3 to 5.0 to avoid abnormal values.

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.7.

$$T^* = \arg\min_T \mathbb{E}_p[w^*(\boldsymbol{x}_i)\mathcal{L}(s(\boldsymbol{z}_i/T), y_i)] \tag{7}$$

Corollary 3.1 *We denote the distribution of the k^{th} class in the long-tailed distribution, the ground truth balanced distribution, and the estimated distribution by $p_k(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}_{p_k}, \boldsymbol{\sigma}^2_{p_k})$, $q_k(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}_{q_k}, \boldsymbol{\sigma}^2_{q_k})$, and $q_k^*(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}_{q_k^*}, \boldsymbol{\sigma}^2_{q_k^*})$, respectively. The absolute error $|\mathbb{E}_{p_k}[w_k(\boldsymbol{x})\mathcal{L}(s(\boldsymbol{z}/T), y)] - \mathbb{E}_{p_k}[w_k^*(\boldsymbol{x})\mathcal{L}(s(\boldsymbol{z}/T), y)]|$ is sensitive to $\epsilon = \mathbb{E}_{p_k}[(w_k(\boldsymbol{x}) - w_k^*(\boldsymbol{x}))^2]$ and the bound of ϵ is shown as follows, where the formula $d_2(\cdot||\cdot)$ presents the exponential in base 2 of the Renyi-divergence (Rényi et al., 1961).*

$$\epsilon \in [(\sqrt{d_2(q_k||p_k)} - \sqrt{d_2(q_k^*||p_k)})^2, d_2(q_k||p_k) + d_2(q_k^*||p_k)] \tag{8}$$

The proof is provided in appendix.A.1. Corollary.3.1 presents the error bound of our method, which is closely related to Renyi-divergence. It is obvious that when $q_k = q_k^*$, the lower bound reaches the minimum and equals 0. Therefore, our estimation method aims to keep q_k^* approaching q_k to reduce the calibration error on the test data.

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

IF	Dataset	Method						
		Base	TS	ETS	TS-IR	IR	IROvA	Ours
IF=100	CIFAR-10	21.79	12.24	12.16	11.64	12.36	13.36	**10.16**
	CIFAR-10.1	28.97	16.75	16.7	16.65	17.13	17.93	**14.24**
	CIFAR-10.1-C	58.22	43.01	43	43.05	43.34	43.83	**40.02**
	CIFAR-F	29.22	15.27	15.24	15.52	15.75	16.23	**12.55**
IF=50	CIFAR-10	17.36	7.65	8.04	8.22	9.75	9.45	**4.93**
	CIFAR-10.1	22.79	10.36	10.99	11.72	13.35	12.7	**6.81**
	CIFAR-10.1-C	55.52	38.66	39.9	40.16	41.58	40.76	**34.29**
	CIFAR-F	25.37	11.3	12.21	12.67	14.39	13.37	**7.59**
IF=10	CIFAR-10	8.39	2.23	1.64	2.03	2.29	2.42	**1.47**
	CIFAR-10.1	13.8	4.87	4.25	4.54	5.38	5.23	**3.79**
	CIFAR-10.1-C	48.31	32.77	31.07	32.11	32.29	31.94	**20.73**
	CIFAR-F	19.73	8.15	6.8	8.42	8.97	8.13	**6.54**

Analysis. We explore how the importance weight influences the calibration compared with temperature scaling. For simplicity, we constitute one-dimensional Gaussian distribution $p(x) \sim \mathcal{N}(\mu_a, \sigma_a^2)$ and $q(x) \sim \mathcal{N}(\mu_b, \sigma_b^2)$, where $\mu_a \neq \mu_b$ and $\sigma_a^2 < \sigma_b^2$. We draw the conclusion that the value of $w(x) > 1$ if $\tau_1 > x$ or $x > \tau_2$, and $w(x) < 1$ if $\tau_1 < x < \tau_2$, where τ_1 and τ_2 are calculated as:

$$\begin{aligned} \tau_1 &= \frac{\mu_a\sigma_b^2 - \mu_b\sigma_a^2 - \sigma_a\sigma_b\sqrt{(\mu_a-\mu_b)^2 + (\sigma_b^2-\sigma_a^2)(\ln\sigma_b^2 - \ln\sigma_a^2)}}{\sigma_b^2 - \sigma_a^2} \\ \tau_2 &= \frac{\mu_a\sigma_b^2 - \mu_b\sigma_a^2 + \sigma_a\sigma_b\sqrt{(\mu_a-\mu_b)^2 + (\sigma_b^2-\sigma_a^2)(\ln\sigma_b^2 - \ln\sigma_a^2)}}{\sigma_b^2 - \sigma_a^2} \end{aligned} \tag{9}$$

The proof is presented in the appendix.A.2. Normally, instances clustered around the mean are more likely to be classified correctly. Therefore, the instances whose importance weight $w(x) < 1$ are likely to be classified right while $w(x) > 1$ on the contrary. In practice, a model trained with imbalanced data generalizes well for head classes but easily overfits tail classes, and hence obtains degraded performances on balanced test data. Our importance weight estimation method actually assigns larger weights to instances of tail classes that are classified incorrectly.

4 Experiment

4.1 Datasets

CIFAR-10-LT. CIFAR-10-LT (Cao et al., 2019) is simulated from balanced CIFAR-10 (Krizhevsky et al., 2009). We conduct experiments with different imbalance factor (IF), which is defined as N_{max}/N_{min} . N_{max} and N_{min} denote the volumes of the most frequent class and the least frequent class, respectively. We 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 (Recht et al., 2018), (3) CIFAR10.1-C (Hendrycks & Dietterich, 2019): 95 synthetics datasets generated on CIFAR-10.1 with different transformations, (4) CIFAR-F (Sun et al., 2021): 20 real-word test sets collected from Flickr. **MNIST-LT**. MNIST-LT is simulated from MNIST (LeCun et al., 1998). 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 (Netzer et al., 2011), (3) USPS (Hull, 1994), (4) Digital-S (Sun et al., 2021): 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 (Cao et al., 2019) 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 (Liu et al., 2019) is simulated from ImageNet (Deng et al., 2009). 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 extensive experiments on a balanced test set.

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

IF	Dataset	Method						
		Base	TS	ETS	TS-IR	IR	IROvA	Ours
IF=100	MNIST	2.52	1.27	1.84	2.82	2.84	1.84	**1.09**
	SVHN	16.06	7.2	11.62	21.25	22.18	14.93	**6.11**
	USPS	15	9.52	12.25	13.25	13.62	10.58	**8.42**
	Digital-S	32.1	22.13	27.35	30.13	31.01	27.48	**20.31**
IF=50	MNIST	1.12	0.85	1.14	1.53	1.54	1.02	**0.8**
	SVHN	22.79	10.36	10.99	11.72	13.35	12.7	**6.81**
	USPS	**2.32**	3.95	3.33	11.42	12.15	2.63	4.18
	Digital-S	11.21	8.14	12.81	11.89	11.91	10.54	**8.1**
IF=10	MNIST	0.56	0.23	**0.21**	0.5	0.52	0.23	0.37
	SVHN	5.75	6.76	6.94	8.1	**4.51**	5.31	7.41
	USPS	8.29	4.81	4.6	6.59	6.98	4.76	**4.46**
	Digital-S	13.55	8.21	8.09	15.37	13.34	8.31	**7.36**

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

Model	Dataset	Method						
		Base	TS	ETS	TS-IR	IR	IROvA	Ours
ResNet-32	CIFAR-100	20.38	2.5	2.1	6.07	9.35	5.92	**1.58**
DenseNet-40	CIFAR-100	16	3.43	2.51	5.57	8.42	5.76	**1.66**
VGG-19	CIFAR-100	27.86	3.81	2.36	6.35	10.35	6.66	**2.18**

4.2 Experiments set up

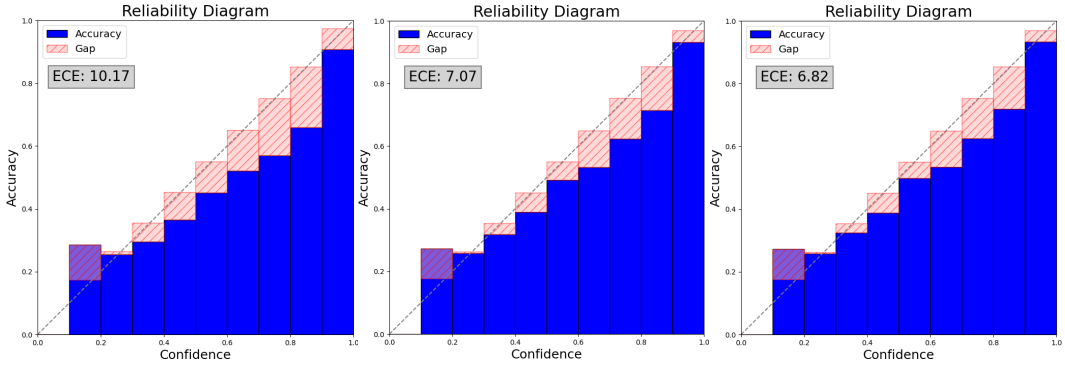
Classification model. We use our method to calibrate different classification models. For CIFAR-10-LT and MNIST-LT, we use ResNet-32 (He et al., 2016) and LeNet-5 (LeCun et al., 1998) as the classification model, respectively. To verify our method can be applied to different models, we apply ResNet-32, DenseNet-40 (Huang et al., 2017), VGG-19 (Simonyan & Zisserman, 2014) as classification models and test on CIFAR-100-LT, respectively. We do experiments on the large-scale dataset, ImageNet-LT, with ResNet-50. Details about training strategies are presented in the appendix.A.3. **Metrics/Baselines**. The most popular evaluation metric for calibration is ECE (Naeini et al., 2015). Besides, we also use SCE (Nixon et al., 2019) and ACE (Neumann et al., 2018) as evaluation metrics. We compare our method with temperature scaling (TS) (Guo et al., 2017), ETS (Zhang et al., 2020), TS-IR (Zhang et al., 2020), IROvA (Zadrozny & Elkan, 2002), and IRM (Zhang et al., 2020). As for our method, all the experiments are conducted with hyper-parameter $\alpha = 0.999$ if not specified.

4.3 Results

CIFAR-10-LT. As shown in Tab.1, our method achieves the best performance on the CIFAR-10-LT dataset. Usually, the model trained with a heavier imbalanced dataset suffers heavier miscalibration. Our method can realize competitive results in different situations. Since CIFAR-10.1 and CIFAR-F are collected from the real-world, the excellent results indicate that our method can generalize to different domains. The results of CIFAR-10.1-C also verify that our method can handle datasets containing different transformations. **MNIST-LT**. Tab.2 demonstrates the effectiveness of our method on MNIST-LT. Except for (IF=50: USPS) and (IF=10: MNIST, SVHN), our method achieves the best performance. Although our method obtains negative optimization on (IF=50: USPS) and (IF=10: SVHN), and lower performance compared with ETS on (IF=10: MNIST), the performance of our method is still acceptable. Overall, our method outperforms other methods significantly in most cases. **CIFAR-100-LT**. For the CIFAR-100-LT benchmark, our method achieves the best results on calibrating different models as shown in Tab.3. We do experiments on three different architectures including ResNet, DenseNet, and VGG. Compared with DenseNet, the performance gains of ResNet and VGG are even higher, while our method achieves the smallest ECE on DenseNet. **ImageNet-LT**. As shown in Tab.4, our method achieves the best performance on the ImageNet-LT benchmark. This indicates that our method can be scaled to large-scale datasets. For detail, our method reduces the ECE value from 10.18% to 6.21% while the second best method,

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

Model	**Dataset**	**Method**						
		Base	TS	ETS	TS-IR	IR	IROvA	Ours
ResNet-50	ImageNet	10.18	6.72	6.87	10.23	11.15	7.63	**6.21**



(a) $\alpha = 0.999$ (b) $\alpha = 0.997$ (c) $\alpha = 0.995$

Figure 4: The reliability diagram of our method with (a) $\alpha = 0.999$, (b) $\alpha = 0.997$, and (c) $\alpha = 0.995$.

TS, can only reduce it to 6.72%. For all datasets, the results of SCE and ACE are presented in the appendix.A.4 and similar conclusions can be observed.

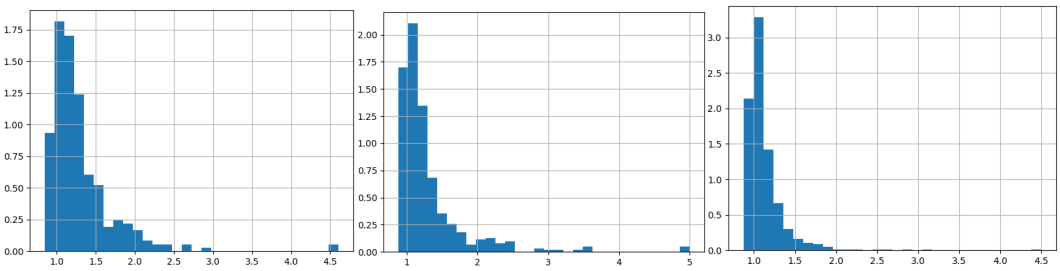
4.4 Ablation Study

Reliability diagram. We also visualize the reliability diagram of our method with different α on the CIFAR-10 test set. As shown in Fig.4, our method can give more reliable results and alleviate the overconfidence problem. The reliability diagrams of the baseline and TS are shown in Fig.2, and our method achieves competitive results compared with them. Specifically, a higher value of α achieves better results in the reliability diagram on this benchmark. Compared with TS, our method well calibrates the predictions for instances with high confidence. We also observe that our method leads to slight underconfidence in instances with very small confidence values, this may be because these samples themselves are more difficult to classify.

The distribution of $w^*(\boldsymbol{x})$. Our method is heavily influenced by the value of $w^*(\boldsymbol{x})$, so we explore the distribution of $w^*(\boldsymbol{x})$ on different datasets. We do experiments on CIFAR-10-LT with IF=100, IF=50, and IF=10, respectively. As shown in Fig.5, the overall distribution of w is clustered around the value $w = 1$. The more imbalanced the dataset, the more instances with larger w values. More instances have larger values of w at IF=100 than at IF=10. Since the dataset with IF=100 suffers a heavier imbalance, it faces a more serious domain shift, and more instances equipped with a large value of w are rational.

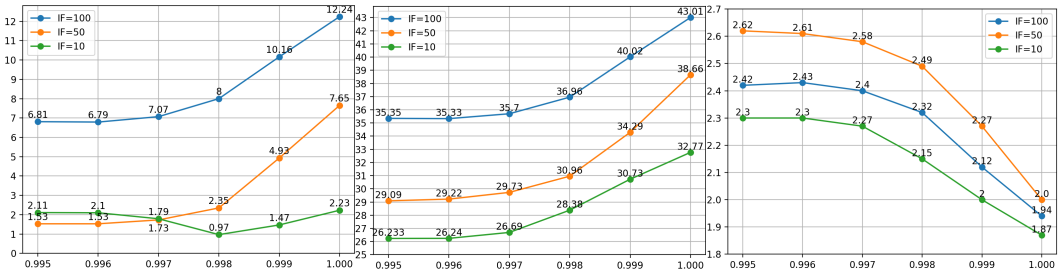
Ablation study on hyper-parameter α. The most important hyper-parameter of our method is α, which controls how much the information of the head class is transferred. Normally, a smaller value of α means we utilize more information from head classes. As shown in Fig.6, with the growth of value α, the value of our temperature exhibits a downtrend. Note that $\alpha = 1.0$ represent traditional temperature scaling since $w(\boldsymbol{x}) = 1$ for all instances. A larger temperature can relieve the overconfidence phenomenon effectively. In addition, a heavily imbalanced dataset (IF=100) needs a larger temperature value compared with a slightly imbalanced dataset (IF=10).

As shown in Fig.6, for the performance on the CIFAR-10 test set, different α achieve different performances. Actually, a heavily imbalanced dataset (IF=100) achieves the best performance on $\alpha = 0.995$ while a slightly imbalanced dataset (IF=10) achieves best on $\alpha = 0.998$. This indicates that we need to utilize more information from head classes if fewer instances of tail classes are available. However, the CIFAR-10.1-C dataset presents different results and we achieve the best performance with $\alpha = 0.995$ for all imbalanced situations. Since CIFAR-10.1-C are synthetic



(a) IF=100 (b) IF=50 (c) IF=10

Figure 5: The distribution histograms of $w^*(\boldsymbol{x})$ with $\alpha = 0.999$. The horizontal axis represents the value of $w^*(\boldsymbol{x})$ and the vertical axis represents the probability density. The model is trained on (a) CIFAR-10-LT with IF=100, (b) CIFAR-10-LT with IF=50, and (c) CIFAR-10-LT with IF=10, respectively.



(a) CIFAR-10 test set (b) CIFAR-10.1-C test set (c) The value of temperature

Figure 6: The blue line, orange line, and green line denote results on CIFAR-10-LT with IF=100, IF=50, and IF=10, respectively. The horizontal axis represents the hyper-parameter α. For (a) and (b) the vertical axis represents the ECE value, and for (c) the vertical axis represents the temperature value. (a) The performance tested on the original CIFAR-10 test set. (b) The performance tested on the CIFAR-10.1-C dataset. (c) The temperature value tested on the original CIFAR-10 test set.

datasets generated from CIFAR-10.1, it suffers a heavier domain shift compared with the CIFAR-10 test set and needs a larger value of temperature to relieve overconfidence. It is interesting that there is a downtrend and uptrend in the green curve in Fig.6 (a). The reason is that the model is underconfident when $\alpha < 0.998$ because of the larger value of temperature. Therefore, it is not a good choice to calibrate every model with a small value of α and it is proper to apply a smaller value of α on a heavily imbalanced dataset and a larger value of α on a slightly imbalanced dataset.

5 Conclusion

In this paper, we propose a novel importance weight-based strategy to realize post-processing calibration under long-tailed distribution. Different from traditional calibration tasks, the tackled problem faces the challenge that the validation set follows a long-tailed distribution while the distribution of the test data is balanced. To this end, we utilize the importance weight strategy to re-weight instances of tail classes. Since it is difficult to acquire the target probability density, we model the distribution of each class as a Gaussian distribution and enhance the estimation of tail class distributions by transferring knowledge from head classes. Extensive experiments on four benchmarks show the effectiveness of our method. In our future work, we intend to explore regularization terms to compensate for the imbalanced influences of head and tail classes for training calibrated models under the long-tailed distribution.

REFERENCES

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Steffen Bickel and Tobias Scheffer. Dirichlet-enhanced spam filtering based on biased samples. *Advances in neural information processing systems*, 19, 2006.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Jiacheng Cheng and Nuno Vasconcelos. Calibrating deep neural networks by pairwise constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13709–13718, 2022.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. *Advances in neural information processing systems*, 23, 2010.

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*, pp. 9268–9277, 2019.

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*, pp. 248–255. Ieee, 2009.

Yunye Gong, Xiao Lin, Yi Yao, Thomas G Dietterich, Ajay Divakaran, and Melinda Gervasio. Confidence calibration for domain generalization under covariate shift. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8958–8967, 2021.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

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*, pp. 770–778, 2016.

Ramya Hebbalaguppe, Jatin Prakash, Neelabh Madan, and Chetan Arora. A stitch in time saves nine: A train-time regularizing loss for improved neural network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16081–16090, 2022.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7610–7619, 2020.

Bingyi Kang, Yu Li, Sa Xie, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *International Conference on Learning Representations*, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2970–2979, 2020.

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*, pp. 2537–2546, 2019.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Relaxed softmax: Efficient confidence auto-calibration for safe pedestrian detection. 2018.

Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *CVPR Workshops*, volume 2, 2019.

Anusri Pampari and Stefano Ermon. Unsupervised calibration under covariate shift. *arXiv preprint arXiv:2006.16405*, 2020.

Sangdon Park, Osbert Bastani, James Weimer, and Insup Lee. Calibrated prediction with covariate shift via unsupervised domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, pp. 3219–3229. PMLR, 2020.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Jing Qin. Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika*, 85(3):619–630, 1998.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

Alfréd Rényi et al. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1. Berkeley, California, USA, 1961.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Xiaoxiao Sun, Yunzhong Hou, Hongdong Li, and Liang Zheng. Label-free model evaluation with semi-structured dataset representations. *arXiv preprint arXiv:2112.00694*, 2021.

Christian Tomani, Sebastian Gruber, Muhammed Ebrar Erdem, Daniel Cremers, and Florian Buettner. Post-hoc uncertainty calibration for domain drift scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10124–10132, 2021.

Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. On calibration and out-of-domain generalization. *Advances in neural information processing systems*, 34:2215–2227, 2021.

Ximei Wang, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable calibration with lower bias and variance in domain adaptation. *Advances in Neural Information Processing Systems*, 33:19212–19223, 2020.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 694–699, 2002.

Jize Zhang, Bhavya Kailkhura, and T Yong-Jin Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International conference on machine learning*, pp. 11117–11128. PMLR, 2020.

A APPENDIX

A.1 ERROR BOUND

Following previous works (Gong et al., 2021; Wang et al., 2020; Pampari & Ermon, 2020; Cortes et al., 2010), we analyze the error bound of importance weight strategy. We denote $p_k(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}_{p_k}, \boldsymbol{\sigma}^2_{p_k})$ as the long-tailed distribution and $q_k(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}_{q_k}, \boldsymbol{\sigma}^2_{q_k})$ as the ground truth balanced distribution, and $q^*_k(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}_{q^*_k}, \boldsymbol{\sigma}^2_{q^*_k})$ as estimated distribution, where index k denotes the k^{th} class. For simplicity, we analyze the error bound of a given class k, and the error bound of other tail classes can also be analyzed following the same procedure. We denote importance weight $w_k(\boldsymbol{x}) = q_k(\boldsymbol{x})/p_k(\boldsymbol{x})$ and $w^*_k(\boldsymbol{x}) = q^*_k(\boldsymbol{x})/p_k(\boldsymbol{x})$. The unbiased error is:

$$\begin{aligned} error &= |\mathbb{E}_{p_k}[w_k(\boldsymbol{x})\mathcal{L}(s(\boldsymbol{z}/T), y)] - \mathbb{E}_{p_k}[w^*_k(\boldsymbol{x})\mathcal{L}(s(\boldsymbol{z}/T), y)]| \\ &= |\mathbb{E}_{p_k}[w(\boldsymbol{x})\mathcal{L}(s(\boldsymbol{z}/T), y) - w^*_k(\boldsymbol{x})\mathcal{L}(s(\boldsymbol{z}/T), y)]| \\ &\leq \sqrt{\mathbb{E}_{p_k}[(w_k(\boldsymbol{x}) - w^*_k(\boldsymbol{x}))^2]\mathbb{E}_{p_k}[(\mathcal{L}(s(\boldsymbol{z}/T), y))^2]} \qquad \text{(Cachy-Schwarz Ineqaulity)} \\ &\leq \frac{1}{2}(\mathbb{E}_{p_k}[(w_k(\boldsymbol{x}) - w^*_k(\boldsymbol{x}))^2] + \mathbb{E}_{p_k}[(\mathcal{L}(s(\boldsymbol{z}/T), y))^2]) \quad \text{(AM/GM Inequality)} \end{aligned} \tag{10}$$

As shown in Eq.10, the unbiased error is sensitive to the term $\mathbb{E}_{p_k}[(w_k(\boldsymbol{x}) - w^*_k(\boldsymbol{x}))^2]$ since $\mathbb{E}_{p_k}[(\mathcal{L}(s(\boldsymbol{z}/T), y))^2])$ is determined. To better understand our method, we analyze the upper bound and lower bound for the first term and denote $\epsilon = \mathbb{E}_{p_k}[(w_k(\boldsymbol{x}) - w^*_k(\boldsymbol{x}))^2]$. The Eq.11 shows that the upper bound of error, that is $d_2(q_k||p_k) + d_2(q^*_k||p_k)$. The formula $d_2(q||p)$ presents the exponential in base 2 of the Renyi-divergence (Rényi et al., 1961) and is defined in Eq.13 and Eq.14.

$$\begin{aligned} \epsilon &= \mathbb{E}_{p_k}[(w(\boldsymbol{x}_i) - w^*(\boldsymbol{x}_i))^2] \\ &= \mathbb{V}_{p_k}[w_k(\boldsymbol{x}) - w^*_k(\boldsymbol{x})] + (\mathbb{E}_{p_k}[w_k(\boldsymbol{x}) - w^*_k(\boldsymbol{x})])^2 \\ &= \mathbb{V}_{p_k}[w_k(\boldsymbol{x}) - w^*_k(\boldsymbol{x})] \qquad (\mathbb{E}_{p_k}[w_k(\boldsymbol{x})] = \mathbb{E}_{p_k}[w^*_k(\boldsymbol{x})] = 1) \\ &= \mathbb{V}_{p_k}[w_k(\boldsymbol{x})] + \mathbb{V}_{p_k}[w^*_k(\boldsymbol{x})] - 2Cov(w_k(\boldsymbol{x}), w^*_k(\boldsymbol{x})) \quad (Cov \text{ denotes covariance function.}) \\ &= d_2(q_k||p_k) + d_2(q^*_k||p_k) - 2Cov(w_k(\boldsymbol{x}), w^*_k(\boldsymbol{x})) - 2 \\ &= d_2(q_k||p_k) + d_2(q^*_k||p_k) - 2\mathbb{E}_{p_k}[w_k(\boldsymbol{x})w^*_k(\boldsymbol{x})] \\ &\leq d_2(q_k||p_k) + d_2(q^*_k||p_k) \end{aligned} \tag{11}$$

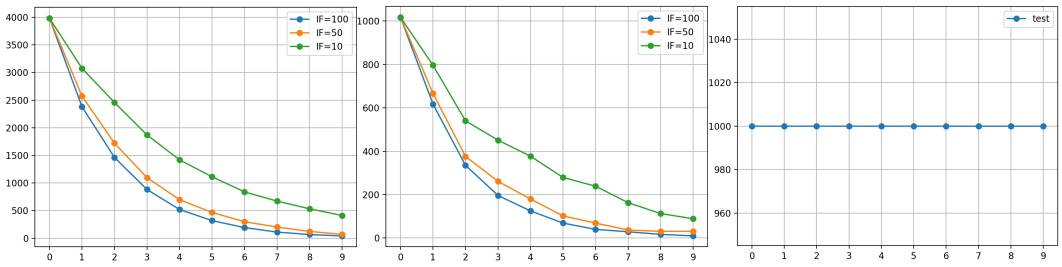
We also analyze the lower bound of our strategy, which is shown in Eq.12. This indicates that the error of our method will larger than $(\sqrt{d_2(q_k||p_k)} - \sqrt{d_2(q^*_k||p_k)})^2$.

$$\begin{aligned} \epsilon &= d_2(q_k||p_k) + d_2(q^*_k||p_k) - 2\mathbb{E}_{p_k}[w_k(\boldsymbol{x})w^*_k(\boldsymbol{x})] \\ &\geq d_2(q_k||p_k) + d_2(q^*_k||p_k) - 2\sqrt{\mathbb{E}_{p_k}[(w_k(\boldsymbol{x}))^2]\mathbb{E}_{p_k}[(w^*_k(\boldsymbol{x}))^2]} \\ &= d_2(q_k||p_k) + d_2(q^*_k||p_k) - 2\sqrt{(\mathbb{V}_{p_k}[w_k(\boldsymbol{x}))] + 1)(\mathbb{V}_{p_k}[w^*_k(\boldsymbol{x}))] + 1)} \\ &= d_2(q_k||p_k) + d_2(q^*_k||p_k) - 2\sqrt{d_2(q_k||p_k)(d_2(q^*_k||p_k)} \\ &= (\sqrt{d_2(q_k||p_k)} - \sqrt{d_2(q^*_k||p_k)})^2 \end{aligned} \tag{12}$$

$$D_\alpha(q||p) = \frac{\alpha}{2}(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^T[\alpha\boldsymbol{\sigma}^2_p + (1-\alpha)\boldsymbol{\sigma}^2_q]^{-1}(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p) - \frac{1}{2(\alpha-1)}\ln\frac{|\alpha\boldsymbol{\sigma}^2_p + (1-\alpha)\boldsymbol{\sigma}^2_q|}{|\boldsymbol{\sigma}^2_q|^{1-\alpha}|\boldsymbol{\sigma}^2_p|^\alpha} \tag{13}$$

$$d_\alpha(q||p) = 2^{D_\alpha(q||p)} \tag{14}$$

Therefore, we draw the conclusion that $\epsilon \in [(\sqrt{d_2(q_k||p_k)} - \sqrt{d_2(q^*_k||p_k)})^2, d_2(q_k||p_k) + d_2(q^*_k||p_k)]$.



(a) Training set (b) Validation set (c) Test set

Figure 7: The distribution of long-tailed cifar-10-LT. The horizontal axis represents the class index and the vertical axis represents the number of instances.

A.2 THE SOLUTION OF ANALYSIS

For simplicity, we constitute one-dimensional Gaussian distribution $p(x) \sim \mathcal{N}(\mu_a, \sigma_a^2)$ and $q(x) \sim \mathcal{N}(\mu_b, \sigma_b^2)$, where $\mu_a \neq \mu_b$ and $\sigma_a^2 < \sigma_b^2$. When $q(x) > p(x)$ we have $w(x) > 1$. As shown in Eq.15, the formula equals to solve the inequality $h(x) = 2\sigma_b^2(x-\mu_a)^2 - 2\sigma_a^2(x-\mu_b)^2 + 4\sigma_a^2\sigma_b^2(\ln\sigma_a - \ln\sigma_b) > 0$. $h(x)$ is a quadratic function.

$$\begin{aligned} q(x) &> p(x) \\ \ln q(x) &> \ln p(x) \\ \ln\sigma_a - \frac{(x-\mu_b)^2}{2\sigma_b^2} &> \ln\sigma_b - \frac{(x-\mu_a)^2}{2\sigma_a^2} \\ 4\sigma_a^2\sigma_b^2\ln\sigma_a - 2\sigma_a^2(x-\mu_b)^2 &> 4\sigma_a^2\sigma_b^2\ln\sigma_b - 2\sigma_b^2(x-\mu_a)^2 \end{aligned} \tag{15}$$

The equation of $h(x) = 0$ have two solutions and denote as τ_1 and τ_2, $\tau_1 < \tau_2$, respectively. The condition for the inequality to hold is $x > \tau_2$ or $x < \tau_1$. Normally, the solution of quadratic function is $\frac{-B \pm \sqrt{B^2-4AC}}{2A}$. The coefficient of our function is shown in Eq.16. Therefore, we get the value of τ_1 and τ_2 in Eq.17.

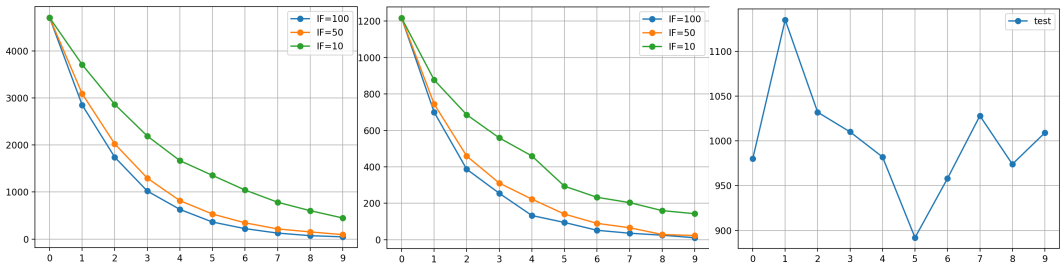
$$\begin{aligned} A &= 2\sigma_b^2 - 2\sigma_a^2 \\ B &= 4\mu_b\sigma_a^2 - 4\mu_a\sigma_b^2 \\ C &= 2\mu_a^2\sigma_b^2 - 2\mu_b^2\sigma_a^2 - 4\sigma_a^2\sigma_b^2(\ln\sigma_b - \ln\sigma_a) \end{aligned} \tag{16}$$

$$\begin{aligned} \tau_1 &= \frac{\mu_a\sigma_b^2 - \mu_b\sigma_a^2 - \sigma_a\sigma_b\sqrt{(\mu_a-\mu_b)^2 + (\sigma_b^2-\sigma_a^2)(\ln\sigma_b^2 - \ln\sigma_a^2)}}{\sigma_b^2 - \sigma_a^2} \\ \tau_2 &= \frac{\mu_a\sigma_b^2 - \mu_b\sigma_a^2 + \sigma_a\sigma_b\sqrt{(\mu_a-\mu_b)^2 + (\sigma_b^2-\sigma_a^2)(\ln\sigma_b^2 - \ln\sigma_a^2)}}{\sigma_b^2 - \sigma_a^2} \end{aligned} \tag{17}$$

A.3 DATASET AND TRAINING STRATEGY

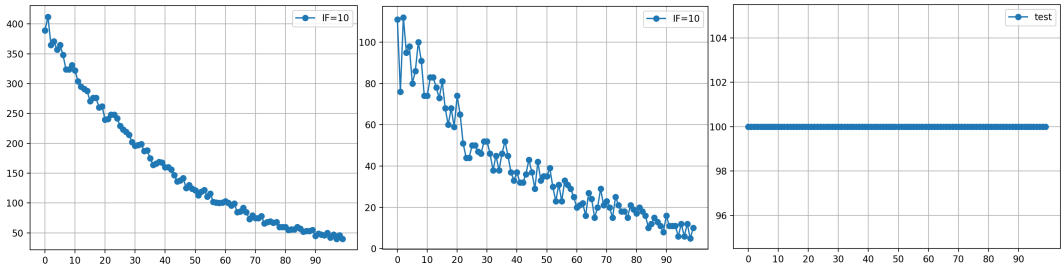
Dataset. The distribution of CIFAR-10-LT, MNIST-LT, CIFAR-100-LT, and ImageNet-LT, are shown in Fig.7, Fig.8, Fig.9, and Fig.10, respectively. The training set and validation set follows the long-tailed distribution while the test set is not. For CIFAR-100-LT and ImageNet-LT, the local distribution of the validation set and the training set exists a little difference. Since we split data randomly and such a phenomenon is rational.

Training strategy. For CIFAR-10-LT and CIFAR-100-LT, we use ResNet-32 as our backbone following (Cui et al., 2019). We use the SGD optimizer and set the initial learning rate to 0.1. The model has trained a total of 200 epochs. The first five epochs are trained with the linear warm-up (Goyal et al., 2017) learning rate schedule. The learning rate drops by 0.1 at epoch 160 and epoch 180, respectively. We follow the most popular setting to set the batch size, the momentum, and the weight decay to 128, 0.9, and 5×10^{-4}, respectively. For MNIST-LT, we use LeNet-5 as



(a) Training set (b) Validation set (c) Test set

Figure 8: The distribution of long-tailed MNIST-LT. The horizontal axis represents the class index and the vertical axis represents the number of instances.



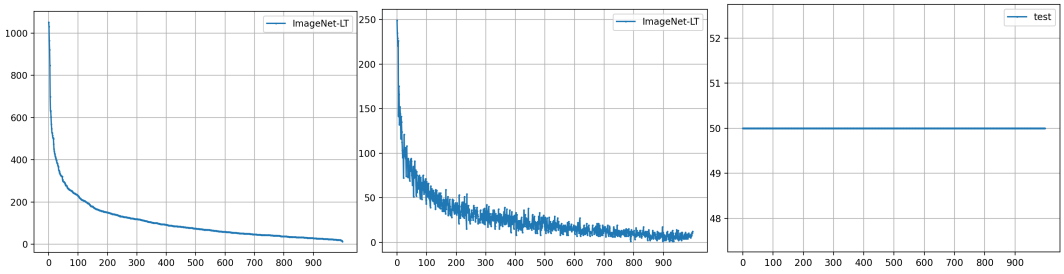
(a) Training set (b) Validation set (c) Test set

Figure 9: The distribution of long-tailed CIFAR-100-LT. The horizontal axis represents the class index and the vertical axis represents the number of instances.

the backbone. We use an SGD optimizer and set the initial learning rate to 0.1. The model has trained a total of 100 epochs. The learning rate drops by 0.1 at epoch 60. We follow the most popular setting to set the batch size, the momentum, and the weight decay to 256, 0.9, and 5×10^{-4}, respectively. For ImageNet-LT, we use ResNet-50 as backbones and adopt the cosine learning rate schedule (Loshchilov & Hutter, 2016) that gradually decays from 0.1 to 0 in the first stage. The model has trained a total of 180 epochs. We follow the most popular setting to set the batch size, the momentum, and the weight decay to 256, 0.9, and 5×10^{-4}, respectively.

A.4 RESULTS

We evaluate our method with different evaluation metrics. The results are shown from Tab.5 to Tab.16. Ours1 and ours2 denote our method with $\alpha = 0.999$ and $\alpha = 0.995$, respectively. For the SCE metric and ACE metric, our method achieves competitive results. The accuracy table shows that our method will preserve the model's accuracy.



(a) Training set (b) Validation set (c) Test set

Figure 10: The distribution of long-tailed ImageNet-LT. The indices are sorted by the number of instances per class. The horizontal axis represents the class index and the vertical axis represents the number of instances.

Table 5: The SCE (%) on CIFAR-10-LT.

IF	**Dataset**	**Method**							
		Base	TS	ETS	TS-IR	IR	IROvA	Ours1	Ours2
IF=100	CIFAR-10	5.11	4.21	4.2	4.03	4.09	4.22	4.11	**3.97**
	CIFAR-10.1	6.49	4.97	4.96	4.82	5.01	5.02	4.77	**4.48**
	CIFAR-10.1-C	12.63	10.21	10.2	10.34	10.31	10.21	9.72	**8.97**
	CIFAR-F	6.61	4.95	4.94	5.01	4.99	4.96	4.74	**4.46**
IF=50	CIFAR-10	4.08	3.36	3.36	**3.19**	3.26	3.35	3.29	3.31
	CIFAR-10.1	5.29	4.03	4.06	3.81	4.02	4.04	3.86	**3.78**
	CIFAR-10.1-C	12.18	9.56	9.69	9.64	9.74	9.5	8.86	**8.04**
	CIFAR-F	5.87	4.32	4.37	4.38	4.51	4.32	4.11	**4.04**
IF=10	CIFAR-10	1.93	1.31	**1.29**	1.32	1.32	1.3	**1.29**	1.36
	CIFAR-10.1	3.17	1.97	1.88	2.12	2.17	1.91	**1.86**	1.87
	CIFAR-10.1-C	10.91	8.51	8.25	8.76	8.58	8.35	8.2	**7.55**
	CIFAR-F	4.56	3.22	3.14	3.32	3.35	3.23	3.13	**3.09**

Table 6: The ACE (%) on CIFAR-10-LT.

IF	**Dataset**	**Method**							
		Base	TS	ETS	TS-IR	IR	IROvA	Ours1	Ours2
IF=100	CIFAR-10	4.99	4.16	4.15	4.02	4.07	4.15	4.06	**3.98**
	CIFAR-10.1	6.22	4.89	4.89	4.8	4.85	4.96	4.66	**4.44**
	CIFAR-10.1-C	12.44	10.16	10.15	10.27	10.24	10.14	9.69	**8.98**
	CIFAR-F	6.37	4.87	4.87	4.87	4.84	4.86	4.68	**4.51**
IF=50	CIFAR-10	3.96	3.33	3.35	**3.2**	3.24	3.33	3.3	3.39
	CIFAR-10.1	4.99	4.02	4.06	**3.84**	3.93	3.96	**3.84**	3.85
	CIFAR-10.1-C	11.97	9.5	9.63	9.59	9.66	9.44	8.85	**8.08**
	CIFAR-F	5.62	4.28	4.33	4.31	4.43	4.28	4.15	**4.14**
IF=10	CIFAR-10	1.76	1.31	**1.28**	1.29	**1.28**	1.3	**1.28**	1.37
	CIFAR-10.1	2.82	1.81	1.76	1.91	1.89	1.78	**1.74**	1.8
	CIFAR-10.1-C	10.71	8.47	8.23	8.68	8.52	8.35	8.18	**7.56**
	CIFAR-F	4.29	3.12	3.07	3.14	3.12	3.08	3.06	**3.01**

Table 7: The Accuracy (%) on CIFAR-10-LT.

IF	**Dataset**	**Method**							
		Base	TS	ETS	TS-IR	IR	IROvA	Ours1	Ours2
IF=100	CIFAR-10	69.38	69.38	69.38	70.55	70.27	68.86	69.38	69.38
	CIFAR-10.1	59.8	59.8	59.8	60.9	60.5	59.3	59.8	59.8
	CIFAR-10.1-C	27.55	27.55	27.55	27.74	27.7	27.4	27.55	27.55
	CIFAR-F	57.79	57.79	57.79	58.45	58.56	57.29	57.79	57.79
IF=50	CIFAR-10	74.68	74.68	74.68	74.92	74.65	74.22	74.68	74.68
	CIFAR-10.1	66.1	66.1	66.1	65.8	66	65.4	66.1	66.1
	CIFAR-10.1-C	29.06	29.06	29.06	28.75	28.71	28.9	29.06	29.06
	CIFAR-F	61.51	61.51	61.51	61.3	61.21	61.06	61.51	61.51
IF=10	CIFAR-10	86.1	86.1	86.1	86.36	86.09	85.86	86.1	86.1
	CIFAR-10.1	77.75	77.75	77.75	77.9	77.75	77.45	77.75	77.75
	CIFAR-10.1-C	33.45	33.45	33.45	33.79	33.66	33.34	33.45	33.45
	CIFAR-F	67.84	67.84	67.84	68.24	68.06	67.65	67.84	67.84

Table 8: The SCE (%) on MNIST-LT.

IF	**Dataset**	**Method**							
		Base	TS	ETS	TS-IR	IR	IROvA	Ours1	Ours2
IF=100	MNIST	**0.82**	0.85	**0.82**	0.97	0.97	0.85	0.87	0.93
	SVHN	4.77	3.68	4.06	7.25	7.35	4.59	3.61	**3.42**
	USPS	3.94	3.34	3.66	4.23	4.24	3.32	3.25	**3.22**
	Digital-S	7.99	6.24	7.16	8.07	8.11	6.91	6	**5.33**
IF=50	MNIST	**0.43**	0.44	**0.43**	0.49	0.49	**0.43**	0.44	0.45
	SVHN	3.08	3.22	3.07	6.33	6.71	**2.87**	3.24	3.33
	USPS	3.39	3.24	3.55	4.23	4.27	3.31	3.23	**3.19**
	Digital-S	5.11	4.59	4.48	8.05	8.2	4.55	4.53	**4.34**
IF=10	MNIST	**0.2**	0.22	0.22	0.21	0.22	0.21	0.23	0.23
	SVHN	**3.52**	3.73	3.76	6.21	5.51	3.56	3.79	3.84
	USPS	**2.91**	3.07	3.09	3.27	3.29	3.15	3.19	3.29
	Digital-S	4.86	4.27	4.25	6.02	5.69	4.35	**4.24**	**4.24**

Table 9: The ACE (%) on MNIST-LT.

IF	**Dataset**	**Method**							
		Base	TS	ETS	TS-IR	IR	IROvA	Ours1	Ours2
IF=100	MNIST	**0.78**	0.85	0.82	0.92	0.93	0.83	0.86	0.92
	SVHN	4.84	3.81	4.24	7.31	7.37	4.54	3.74	**3.59**
	USPS	3.82	3.28	3.56	4.11	4.02	3.38	3.22	**3.09**
	Digital-S	7.84	6.29	7.09	7.99	8	6.85	6.06	**5.51**
IF=50	MNIST	**0.4**	0.43	0.41	0.45	0.45	0.41	0.43	0.44
	SVHN	3.21	3.51	3.22	6.19	6.6	**3.12**	3.53	3.69
	USPS	3.26	3.15	3.37	4.01	4.01	3.27	3.14	**3.13**
	Digital-S	5.05	4.58	5.38	7.69	7.83	4.66	4.56	**4.39**
IF=10	MNIST	**0.17**	0.19	0.19	0.19	0.19	0.18	0.19	0.2
	SVHN	**3.79**	4.02	4.03	6.26	5.58	3.88	4.08	4.15
	USPS	**2.82**	3.03	3.05	3.05	3.1	3.06	3.1	3.2
	Digital-S	4.84	4.43	4.42	5.86	5.55	4.4	**4.37**	**4.37**

Table 10: The Accuracy (%) on MNIST-LT.

IF	Dataset	Method							
		Base	TS	ETS	TS-IR	IR	IROvA	Ours1	Ours2
IF=100	MNIST	95.12	95.12	95.12	94.69	94.57	94.92	95.12	95.12
	SVHN	26.23	26.23	26.23	22.85	23.39	25.74	26.23	26.23
	USPS	72.99	72.99	72.99	71.79	71.79	72.24	72.99	72.99
	Digital-S	30.88	30.88	30.88	28.84	28.84	30.13	30.88	30.88
IF=50	MNIST	97.44	97.44	97.44	97.49	97.5	97.34	97.44	97.44
	SVHN	36.26	36.26	36.26	32.16	31.79	35.8	36.26	36.26
	USPS	76.08	76.08	76.08	75.73	75.63	75.83	76.08	76.08
	Digital-S	41.58	41.58	41.58	39.26	38.8	41.13	41.58	41.58
IF=10	MNIST	98.57	98.57	98.57	98.38	98.35	98.56	98.57	98.57
	SVHN	35.71	35.71	35.71	29.69	30.6	33.49	35.71	35.71
	USPS	79.67	79.67	79.67	78.62	78.52	79.47	79.67	79.67
	Digital-S	43.38	43.38	43.38	40.26	41.03	41.81	43.38	43.38

Table 11: The SCE (%) on CIFAR-100-LT.

Model	Dataset	Method							
		Base	TS	ETS	TS-IR	IR	IROvA	Ours1	Ours2
ResNet-32	CIFAR-100	0.52	0.33	**0.32**	0.39	0.4	0.34	0.33	**0.32**
DenseNet-40	CIFAR-100	0.44	**0.33**	**0.33**	0.37	0.37	**0.33**	**0.33**	**0.33**
VGG-19	CIFAR-100	0.64	0.28	0.28	0.28	0.29	0.29	**0.27**	0.28

Table 12: The ACE (%) of CIFAR-100-LT.

Model	Dataset	Method							
		Base	TS	ETS	TS-IR	IR	IROvA	Ours1	Ours2
ResNet-32	CIFAR-100	0.37	**0.2**8	0.29	0.35	0.35	0.29	**0.28**	**0.28**
DenseNet-40	CIFAR-100	0.33	**0.28**	**0.28**	0.31	0.31	**0.28**	**0.28**	**0.28**
VGG-19	CIFAR-100	0.39	0.26	0.26	0.35	0.34	0.27	**0.26**	0.27

Table 13: The Accuracy (%) on CIFAR-100-LT.

Model	Dataset	Method							
		Base	TS	ETS	TS-IR	IR	IROvA	Ours1	Ours2
ResNet-32	CIFAR-100	56.13	56.13	56.13	54.67	54.89	55.98	56.13	56.13
DenseNet-40	CIFAR-100	60.39	60.39	60.39	59.74	59.5	60.25	60.39	60.39
VGG-19	CIFAR-100	56.06	56.06	56.06	54.4	54.8	56.01	56.06	56.06

Table 14: The SCE (%) of ImageNet-LT.

Model	Dataset	Method							
		Base	TS	ETS	TS-IR	IR	IROvA	Ours1	Ours2
ResNet-50	ImageNet	0.051	**0.049**	**0.049**	0.053	0.053	0.05	**0.049**	**0.049**

Table 15: The ACE (%) on ImageNet-LT.

Model	Dataset	Method							
		Base	TS	ETS	TS-IR	IR	IROvA	Ours1	Ours2
ResNet-50	ImageNet	0.042	**0.041**	0.042	0.043	0.043	**0.041**	**0.041**	**0.041**

Table 16: The Accuracy (%) on ImageNet-LT.

Model	**Dataset**	**Method**							
		Base	TS	ETS	TS-IR	IR	IROvA	Ours1	Ours2
ResNet-50	ImageNet	48.68	48.68	48.68	48.11	48.11	48.61	48.68	48.68