Revisiting the Effective Number Theory for Imbalanced Learning

Ou Wu, Mengyang Li

Abstract—Imbalanced learning is a traditional yet hot research subarea in machine learning. There are a huge number of imbalanced learning methods proposed in previous literature. This study focuses on one of the most popular imbalanced learning strategies, namely, sample reweighting. The key issue is how to calculate the weights of samples in training. While most studies have relied on intuitive theoretical or heuristic inspirations, few studies have attempted to establish a comprehensive theoretical path for weight calculation. A recent study utilizes the effective number theory for random covering to construct a theoretical weighting framework. In this study, we conduct a deep analysis to theoretically reveal the defects in the existing effective number-based weighting theory. An enhanced effective number theory is established in which data scatter and covering offset among different categories are involved. Subsequently, a new weight calculation manner is proposed based on our new theory, yielding a new loss, namely, NENum loss. In this loss, weights are sample-wise instead of category-wise used in the existing effective number-based weighting. Furthermore, another novel loss that combines weighting and logit perturbation is designed inspired the limitations of the NENum loss. Meta learning is employed to optimize the concrete calculation based on sample-wise training dynamics. We conduct extensive experiments on benchmark imbalanced and standard data corpora. Results validate the reasonableness of our enhanced theory and the effectiveness of the proposed methodology.

Index Terms—Imbalanced learning, effective number, covering offset, weight calculation, meta learning.

1 Introduction

In many real data classification tasks, imbalance (especially class imbalance) exists inevitably due to the intrinsic nature of the involved data or the technical limitation of data collecting [1]. On these tasks, the learned models usually perform poorly on categories with small proportions of training samples. Addressing this challenge falls under the domain of imbalanced learning, a specialized subarea of machine learning that focuses on mitigating the impact of imbalanced data distributions across categories (or regression targets).

The dominant categories in a training set are referred to as majority categories, whereas those occupying little are called minority ones. Previous literature has proposed numerous imbalanced learning methods aiming to enhance model performance on both the minority categories and the entire data space. Generally, most existing methods can be roughly placed into six technical paths: data resampling, data reweighting, model adaptation, loss modification, data augmentation, and model ensemble. Many approaches combine multiple paths to form the final learning strategy. For instance, Zhou and Liu [2] employed ensemble learning to combine resampling and threshold-moving methods to address class imbalance. Their study provides valuable insights into the application of shallow classifiers for imbalanced learning.

Among the various imbalanced learning strategies, data reweighting is usually among the first choices as it is independent of the involved classification models and training loss and thus can work as a plug-in module. The primary issue for data reweighting lies in the calculation of the weights for training samples. Generally, the category proportion is considered and the larger the category proportion is, the smaller the weight is. Existing weighting strategies can also be roughly divided into category-wise and sample-wise ones, or into static and dynamic ones. Many strategies belong to category-wise, whereas methods such as Focal loss [3] belong to sample-wise. In current deep learning tasks, as the deep neural network (DNN) models are optimized over training epochs, the dynamic weighting strategy is demonstrated to be more effective than the static weighting strategy. Fernando and Tsokos [4] designed dynamically weighted loss based on both the class proportion and model prediction on each training sample. Ren et al. [5] inferred the sample weights at each epoch based on the training dynamics of samples in each epoch.

There are limited studies focusing on the theoretical aspect of the weight calculation for imbalanced learning. Most methods assume the category proportion is the categorical prior probability. Consequently, the inverse of the category proportion can be directly used as the categorical weight based on the Bayesian rule. Recently, Cui et al. [11] conducted a pioneering study to establish a theoretical framework for weight calculation based on the effective number theory which is explored in the random covering problem [12]. They assumed that categories with larger expected data volumes than others should have smaller weights and proposed the following calculation manner for the weight of the category y:

$$w_y = \frac{1-\beta}{1-\beta^{n_y}}, \tag{1}$$

- *Ou Wu and Mengyang Li are with the Center for Applied Mathematics, Tianjin University, Tianjin, China, 300072.*
E-mail: {wuou, limengyang}@tju.edu.cn

Manuscript received July 31, 2023.

where β is a hyper-parameter approaching one and n_y is the number of training samples for category y. This weight yields the class-balanced loss [11] and provides a rationale for the diminishing marginal benefits of increasing the training size in imbalanced learning. Although Cui et al.'s paper receives a large number of citations, no study in the papers that cite this original paper of effective number-based weighting has attempted to perform further discussion for the effective number-based theory (referred to as ENum weighting theory) for imbalanced learning. Moreover, even though ENum weighting has demonstrated effectiveness in various benchmark datasets, our analysis reveals that it has some non-trivial limitations in its theoretical basis, which is detailed in Sections 3.1 and 3.2.

In this study, we conduct an in-depth theoretical analysis of the ENum weighting theory in the context of imbalanced learning. We propose an enhanced ENum weighting theory, along with a novel methodology. First, we theoretically discuss the limitations of Cui et al.'s ENum weighting theory [11] and reveal that the determination of category-wise weights in imbalanced learning cannot solely rely on the data volume. Moreover, the ENum weighting theory overlooks an essential sample interaction in random covering. Second, we establish a new effective number-based weighting theory by considering two new factors, including data scatter of different categories and covering offset in random covering. Our theory is founded on more reasonable assumptions and establishes a new weighting mechanism. Third, we present two new loss functions. The first one is the NENum loss, which enables sample-wise reweighting. The second loss combines weighting with logit perturbation. Meta learning is leveraged to infer the hyper-parameters and important variables in the combining loss, which is called the Meta-ENum method. The experimental results as well as ablation study suggest that our Meta-ENum outperforms state-of-the-art (SOTA) weighting methods for imbalanced learning.

Our main contributions are summarized as follows:

- The defects of the existing effective number-based weighting theory used for imbalanced learning are revealed. Although the ENum weighting theory is a classical study for imbalanced learning, it is built upon an erroneous primary assumption and fails to consider another crucial factor, namely, data scatter.
- A new effective number-based weighting theory is established based on our theoretical exploration. The weighting mechanism in our theory can alleviate the defects of the existing theory and is more reasonable.
- Two new training losses are proposed. The hyper-parameters and important variables of the proposed combination loss are optimized via meta learning, which forms a new method, namely, Meta-ENum. It outperforms existing SOTA methods.

2 Related work

2.1 Imbalanced Learning

Recently, with the growing application of deep learning in various domains, imbalanced learning has received increasing attention in both the research and industrial communities. Most imbalanced learning studies concern the classification tasks, while some studies concern other tasks such as regression and clustering. At present, typical imbalanced learning methods can be placed into the following folds:

- Resampling/reweighting. This type of methods reorganizes the training set by down/over-sampling of the original training samples, or by exerting different weights on different samples. Xie et al. [41] proposed a new Gaussian distribution-based oversampling technique to handle the imbalanced classification.
- New model. This type of methods adapts classical learning models (e.g., DNNs) to new models special for imbalanced learning tasks. Zhou et al. [27] constructed a novel DNN architecture to learn effective feature representations for both the majority and the minority categories.
- New loss. This type of methods adapts classical training losses models (e.g., cross entropy loss) to new losses special for imbalanced learning tasks. Cao et al. [26] designed a novel category distribution-aware margin loss to deal with category imbalance.
- Data augmentation. This type of methods synthesizes new training samples for imbalanced learning tasks. Both explicit and implicit augmentation strategies are proposed in the literature. Classical explicit methods include SMOTE [9] and mixup [21]. Typical implicit augmentation methods include ISDA [22] and MetaSAug [23].
- Ensemble. This type of methods fuses different models to deal with imbalance. Liu et al. [14] proposed a new method to resample training data to generate multiple classifiers and formed a cascade ensemble model, in which the resampling strategy is optimized via meta learning. Yang et al. [7] combined ensemble learning and sample reweighting to construct a more effective and robust broad learning system for imbalanced learning.

Although numerous studies have been conducted in the previous literature, it is still inadequate to conclude that imbalanced learning has been fully addressed and numerous open problems remain unsolved. For instance, does the ENum weighting theory hold under arbitrary imbalanced conditions? The establishment of a theoretical analysis framework might aid the uncovering of a general conclusion. Additionally, some studies explore the issue of intra-class imbalance in which imbalance occurs in a category [39], [40].

2.2 Sample Reweighting

This study focuses on the technical path of sample reweighting, in which training samples are assigned distinct weights during training. Indeed, sample reweighting is a common technique in machine learning. Apart from imbalanced learning, sample reweighting finds widespread application in at least the following subdivisions:

- Noisy-label learning (NLL). Sample reweighting is also the primay solution for NLL. Noisy labels are judged and low or zero weights are assigned to these samples.

- Curriculum learning (CL). CL is motivated by human learning procedure that easy knowledge is learned first and hard is learned later [20]. It assigns low or zero weights for hard samples in the early training stage and then increases their weights along training epochs [25].
- Boosting. In shallow classifiers such as AdaBoost [19], hard training samples are assigned large weights in the next training iteration.

Clearly, there is a conflict in the inspirations between CL and Boosting. The former chooses the easy first scheme (i.e., easy samples are assigned with large weights), whereas the latter chooses the hard first one. Zhou and Wu [13] investigated the issue of which samples should be learned first in machine learning. However, limited studies aim to develop a theoretical framework for how to calculate the weights of training samples for imbalanced learning. To our knowledge, in addition to the conventional Bayesian rule, only Cui et al. [11] made an effort on this issue on the basis of the effective number theory.

2.3 Data Enhancement

Many recent imbalanced methods do not employ sample reweighting to address imbalance. Instead, they utilize data enhancement methodologies to optimize the training data. Data augmentation is a typical data enhancement methodology, which can simulate new training data for minor classes. Representative data augmentation-based imbalance learning methods include SMOTE [9] and mixup [21].

Data perturbation is also an effective data enhancement methodology. It generates new features, logits, or labels on the basis of the raw training samples. Many recent classical imbalanced learning methods with different motivations can be attributed to the perturbation on logits such as logit adjustment (LA) [36], LDAM [26], and MetaSAug [23]. Let $\boldsymbol{\nu}(\boldsymbol{x}_i)$ be the logit output by a deep neural network for a sample $\boldsymbol{x}_i$, and let y_i be corresponding label. Let π_y be the proportion of training samples with y. Logit perturbation modifies the yth dimension of the logit into the following

$$\nu'_y(\boldsymbol{x}_i) = \nu_y(\boldsymbol{x}_i) + \delta_{y_i,y}, \tag{2}$$

where $\delta_{y_i,y}$ is yth quantity of the perturbation vector for the logits of the y_ith class. In LA, $\delta_{y_i,y} = log\pi_y$; in LDAM, $\delta_{y_i,y} = -log\pi_{y_i}$; in MetaSAug, $\delta_{y_i,y} = (\boldsymbol{\omega}_y - \boldsymbol{\omega}_{y_i})^T \Sigma_{y_i} (\boldsymbol{\omega}_y - \boldsymbol{\omega}_{y_i})$.

This study will combine sample reweighting and logit perturbation for imbalanced learning.

2.4 Meta Learning

Meta learning is also denoted "learning to learn" [28]. One main application of meta learning is the seeking of optimal hyper-parameters. Shu et al. [15] designed a sophisticated weighting network for NLL and imbalanced learning. The weighting network is trained on an unbiased meta set with meta learning. Li et al. [16] developed a logit perturbation strategy to address class imbalance and the perturbation hyper-parameters are optimized via meta learning. Meta learning requires an unbiased (e.g., clean and balanced) meta set. In NNL, a clean meta set may be difficult to construct. Nevertheless, a balanced meta set can be easily constructed from training data, so this study also utilizes meta learning to seek optimal hyper-parameters.

2.5 Cost-sensitive Learning

Cost-sensitive learning is an application-oriented machine learning division. In many real-world applications, such as medical diagnosis, the misclassified costs of different categories are different. Numerous classical machine learning methods have been adapted to cost-sensitive learning, such as cost-sensitive SVM [18] and cost-sensitive boosting [17].

Reweighting (also called rescaling in cost-sensitive learning) is almost the most popular approach to cost-sensitive learning [30]. Previous studies [10], [24] revealed that cost-sensitive learning methodologies are also good solutions to the class imbalance problem. Chen et al. [55] proposed a cost-sensitive online adaptive kernel learning algorithm to handle large-scale imbalanced multi-class classification problems. Siers and Islam [42] constructed a taxonomy which encompasses approaches to both class imbalance treatment and cost-sensitive classification. Some studies [6] investigated the influence of class imbalance on cost-sensitive learning and perspective findings were revealed.

Notation	Description
$\boldsymbol{x}$	a sample or its feature
y (or z)	a categorical label
C	the number of categories in a classification task
n_y	the number of training samples in category y
π_y	the proportion of training samples in y
δ_{y_1,y_2}	the y_2th perturbation quantity for the logits of y_1
$Net(\boldsymbol{x})$	the output of the feature encoding layer for x
$\boldsymbol{\nu}(\boldsymbol{x})$	the logit vector of x
$f(\boldsymbol{x})$	the output of a classifier f for a sample $\boldsymbol{x}$
$\boldsymbol{\omega}_y$	the weight vector for y in the last layer
$\boldsymbol{\Sigma}_y$	the covariance matrix for category y
$\mathcal{S}_y$	the set of all data in the feature space of y
V_y	the volume of category y
E_k	the effective number of k random samples
w_y	the weight of category y
$\boldsymbol{I}$	the unit matrix
$\mathcal{A}(f, y)$	the classification accuracy of f on y
$P(y)$	the prior probability of y
$\mathbb{R}^n$	the n-dimensional Euclid space
$\mathcal{N}(0, 1)$	the normal distribution
σ_y	the variance (factor) for y
$\mathcal{DS}$	the data scatter degree
$p(y\|\boldsymbol{x})$	the posterior probability of y on $\boldsymbol{x}$
l	the loss
$\boldsymbol{\Theta}$	the parameters for the backbone network
$\boldsymbol{\Omega}$	the parameters for the meta-learning modular
β	the hyper-parameter for weighting

TABLE 1: Summary of the Notations.

3 Theoretical Analysis

This section briefly reviews the existing ENum weighting theory [11]. The defects of this theory are then discussed. The main notations of this study are summarized in Table 1.

3.1 The Existing ENum Weighting Theory

The effective number (E_k) represents the expected volume of a set containing k random samples. The ENum weighting theory established by Cui et al. [11] is built on the basis of the following four assumptions:

Assumption 1. Each sample has the unit volume of 1.

Assumption 2. $\forall y$, $\mathcal{S}_y$ is bounded. Alternatively, V_y is not infinity. The values of V_ys of all categories are identical[1].

Assumption 3. There are only two ways that a newly sampled data can interact with existing sampled data: it either overlaps with one of the existing data with a probability p, or it remains entirely outside the existing data with a probability $1-p$.

Assumption 4. The weight assigned to category y is inversely proportional to the effective number (E_{n_y}) of the training samples in category y.

Based on the above assumptions and some straightforward inference, Cui et al. [11] calculated the effective number E_{n_y} of a category by using

$$E_{n_y} = \frac{1-\beta_y^{n_y}}{1-\beta_y}, \tag{3}$$

where $\beta_y = 1 - 1/V_y \in (0,1)$. According to Assumption 4, Cui et al. proposed the following weighting mechanism for category y:

$$w_y \propto \frac{1}{E_{n_y}} = \frac{1-\beta_y}{1-\beta_y^{n_y}}. \tag{4}$$

The four assumptions and the above weighting formulation constitute Cui et al.'s effective number-based theory. Evidently, the following two equations can be obtained with Taylor expansion for (4):

$$\lim_{n_y \to +\infty} w_y \propto \frac{1}{V_y} \quad \text{and} \quad \lim_{V_y \to +\infty} w_y \propto \frac{1}{n_y}. \tag{5}$$

The conclusion of (5) indicates that the category weight remains small when its volume (V_y) or training size (n_y) is large. Both of these findings are highly reasonable. In particular, (5) can well explain the marginal benefits of increasing the training size n_y. As n_y grows to a considerable value, the weight of category y approaches to a constant value ($\propto 1/V_y$). Moreover, the weights of all categories are equal as V_ys become identical according to Assumption 2.

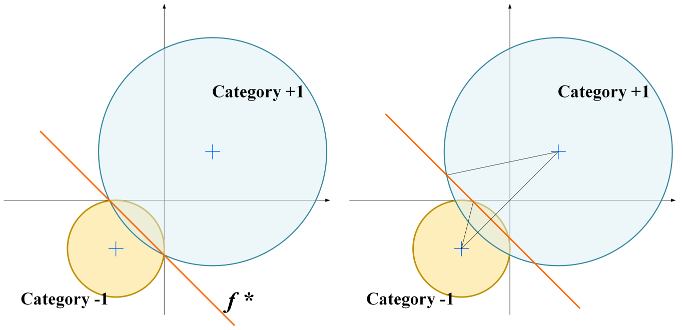
Extensive experiments [11] validate the effectiveness of Cui et al's theory and algorithm. Nevertheless, we argue that their theory has the following limitations:

Limitation 1. There is no theoretical justification for why only the effective number determines the weight in imbalanced learning, as described in Assumption 4.

Limitation 2. Assumption 3 ignores that two or more categories may intersect with each other. Therefore, a newly sampled data may offset another category's sample, thereby suggesting a third way of sample interaction.

Limitation 3. The ENum weights are static. Nevertheless, in deep learning, the data volumes of each category may vary as the feature space continuously optimizes during training. This inconsistency clearly contradicts the underlying assumption of using static weights.

It is noteworthy that, although the identical volume assumption in Assumption 2 is also incorrect which has been identified by Cui et al. [11], this study still inherits this assumption because it is nearly impossible to calculate the true values of V_ys. As Limitation 3 is obvious, the subsequent subsection presents a theoretical analysis of Limitations 1 and 2.



(a) $p_+ : p_- = K^2 : 1$ (b) $p_+ : p_- = 1:1$

Fig. 1: Two binary learning tasks under uniform distribution. The classes' volumes are different.

3.2 Analysis for the Limitations

3.2.1 Analysis for Limitation 1

Two typical learning cases are explored to expose the first limitation that the effective number is only determined by the class volume (i.e., effective number) assumed in the existing theory.

Case I: Learning under unequal classes' volumes

In this case, classification tasks with unequal class volumes are constructed to analyze the relationship between class weights and their volumes.

Considering the following binary classification task. The data from each category follow a uniform distribution $\mathcal{D}$ within two circles that are centered on $\boldsymbol{\theta}$ and $-\boldsymbol{\theta}$, respectively. A K-factor difference is found between two circles' radius: $r_+ : r_- = K : 1$ and $K > 1$. The data follow

$$\begin{aligned}
&P(y=+1) = p_+, P(y=-1) = p_-, \\
&\boldsymbol{\theta} = [\eta, \eta]^T \in \mathbb{R}^2, \eta > 0, \\
&\boldsymbol{x} \sim \begin{cases} \frac{p_+}{\pi K^2 \eta^2}, & \text{if } y=+1, ||\boldsymbol{x}-\boldsymbol{\theta}||_2 \le K\eta, \\ \frac{p_-}{\pi \eta^2}, & \text{if } y=-1, ||\boldsymbol{x}+\boldsymbol{\theta}||_2 \le \eta, \\ 0, & \text{otherwise.} \end{cases}
\end{aligned} \tag{6}$$

Fig. 1 illustrates two binary learning tasks with different prior probability ratios: (a) $p_+ : p_- = K^2 : 1$ and (b) $p_+ : p_- = 1 : 1$. We next reveal that in these two tasks, the data scatter rather than the volume determines the category weight when imbalanced learning is involved. Let the optimal linear classifier f^* be obtained by minimizing the average classification error, i.e.,

$$\begin{aligned}
f^* = \arg\min_f \{ &P(f(\boldsymbol{x}) \neq y \mid y=+1) \times p_+ \\
&+ P(f(\boldsymbol{x}) \neq y \mid y=-1) \times p_- \},
\end{aligned} \tag{7}$$

In the first task, the two categories have *equal* probability densities as $\frac{p_+}{\pi K^2 \eta^2} = \frac{p_-}{\pi \eta^2}$; in the second task, the two categories have unequal probability densities. The following two theorems with opposite conclusions can be obtained.

Theorem 1. For a data distribution $\mathcal{D}$ in (6) and $p_+ : p_- = K^2 : 1$, let f^* be the achieved optimal linear classifier.

[1] Although Cui et al. [11] had pointed out that V_ys may be different, V_ys are still assumed identical in their final method.

The classification accuracy ($\mathcal{A}$) of class +1 is larger than that of class -1, i.e., $\mathcal{A}\left(f^*, +1\right) > \mathcal{A}\left(f^*, -1\right)$.

Theorem 2. For a data distribution $\mathcal{D}$ in (6) and $p_+ : p_- = 1 : 1$, let f^* be the achieved optimal linear classifier. The classification accuracy of class +1 is smaller than that of class -1, i.e., $\mathcal{A}\left(f^*, +1\right) < \mathcal{A}\left(f^*, -1\right)$.

The proof is presented in the online supplementary materials. Theorem 1 indicates that if the probability densities at each sample are identical, then the category with the larger volume will have better classification performance. Consequently, a small weight should be assigned to this category (category '+1' in the setting) if imbalance (classification fairness) is concerned. That is, Assumption 4 holds in the learning setting of Theorem 1. However, Theorem 2 contradicts the validity of Assumption 4, as it suggests that category '-1' should receive a smaller weight, despite having a smaller volume than category '+1', when classification fairness is a primary concern.

Considering that sampled data in regions with small probability densities are usually more scattered than those in regions with large probability densities. The contradictory conclusions from Theorems 1 and 2 suggest that another important data property, namely, scatter, also impacts the category-wise performance and thus the category weight. In Theorem 2, although category '-1' has a small volume, its scatter is smaller than that of category '+1'. Accordingly, category '-1' is superior to category '+1' and it should be assigned a small weight in imbalanced learning.

Case II: Learning under equal classes' volumes

In this case, classification tasks with equal class volumes are constructed. Xu et al. [29] examined the performance gap between two categories within a binary classification task, adhering to the following distribution:

$$
\begin{aligned}
& y \overset{u.a.r}{\sim} \{-1, +1\}, \quad \boldsymbol{\theta} = [\eta, \ldots, \eta]^T \in \mathbb{R}^d, \eta > 0, \\
& \boldsymbol{x} \sim \begin{cases} \mathcal{N}\left(\boldsymbol{\theta}, \sigma_+^2 \boldsymbol{I}\right), & \text{if } y = +1 \\ \mathcal{N}\left(-\boldsymbol{\theta}, \sigma_-^2 \boldsymbol{I}\right), & \text{if } y = -1 \end{cases} ,
\end{aligned} \tag{8}
$$

where a K-factor difference is set between two classes' variances: $\sigma_+ : \sigma_- = K : 1$ and $K > 1$. Fig. 2 illuminates the two categories. There are no bounds for both categories and category '-1' is more compact than category '+1'. Under the optimal linear classifier f^* defined in Eq. (7), the performance gap is

$$
\begin{aligned}
\nabla_{\mathcal{A}} &= \mathcal{A}\left(f^*, -1\right) - \mathcal{A}\left(f^*, +1\right) \\
&= P\left\{-K \cdot B + R \leq \mathcal{N}(0,1) \leq B - K \cdot R\right\} > 0\text{'}
\end{aligned} \tag{9}
$$

where $B = \frac{2}{K^2-1} \frac{\sqrt{d}\eta}{\sigma_-}$, $R = \sqrt{B^2 + q(K)}$, and $q(K) = \frac{2 \log K}{K^2-1}$. That is, category '+1' is more harder and has smaller classification accuracy than category '-1'.

The volumes of the two categories in the above task are infinite. To meet the conditions of the effective number theorem that categories have bounded volumes, we construct a new learning task based on the above learning task. For each category in (8), there must be a bound ρ such that the probability of samples whose distances to the categorical center are larger than the bound is less than $\nabla_{\mathcal{A}}/100$:

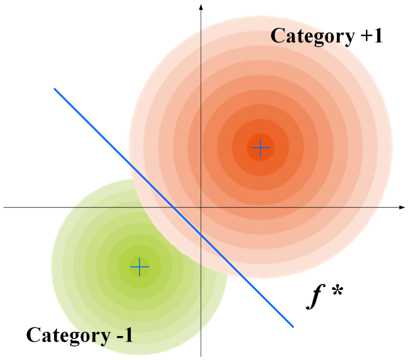


Fig. 2: Two categories with Gaussian distributions. Based on them, a task with equal classes' volumes can be constructed.

$$
\begin{aligned}
P(\|\boldsymbol{x} - \boldsymbol{\theta}\|_2 > \rho | y = +1) \leq \frac{\nabla_{\mathcal{A}}}{100} \\
P(\|\boldsymbol{x} + \boldsymbol{\theta}\|_2 > \rho | y = -1) \leq \frac{\nabla_{\mathcal{A}}}{100}
\end{aligned} . \tag{10}
$$

The data samples distributed in the areas described in (10) can be relocated to other areas within the same category. We can certainly devise a relocation strategy such that the optimal linear classifier f^* remains unchanged and aligns with the classifier f^* for the distribution in (8)[2].

As f^* remains unchanged, for the new performance gap between categories '+1' and '-1', we obtain

$$
\nabla'_{\mathcal{A}} = \mathcal{A}\left(f^*, -1\right) - \mathcal{A}\left(f^*, +1\right) \geq \frac{98}{100} \nabla_{\mathcal{A}} > 0. \tag{11}
$$

The proof is straightforward. This inference denotes that even when the two categories have identical volumes, category '-1' still exhibits higher accuracy than category '+1'.

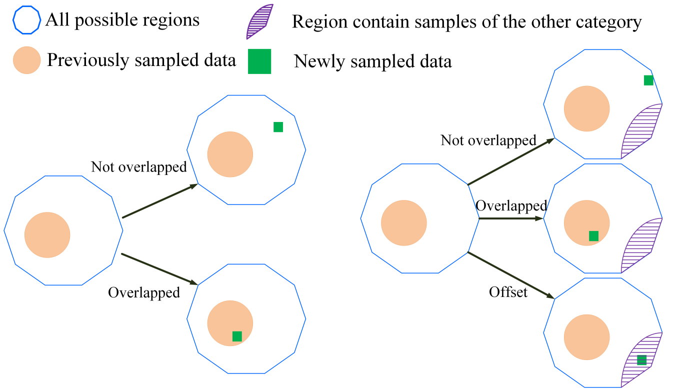
Furthermore, if the bound ρ for category '+1' is adjusted to 2ρ while the bound ρ for category '-1' is adjusted to 1.5ρ or 3ρ, the conclusion stated in (11) remains valid. Alternatively, the relative volume of two categories does not affect their relative accuracy for the two categories. That is, if imbalanced learning is involved, then the weights are not solely dependent on the class volume.

Summary

Based on the aforementioned analysis on two cases, the following conclusions are obtained:

- We present a theoretical condition ensuring the validity of Assumption 4: when dealing with two categories having bounded volumes and both following a uniform distribution, if their probability densities are equal, the category with the larger volume receives a lower weight. Nevertheless, when their probability densities are not equal, Assumption 4 may not hold.
- We reveal that another crucial factor, known as data scatter, also heavily determines the category weight. Note that a consistent relative relationship exists among all the above designed cases: the variance factor σ_-^2 for category '-1' is consistently smaller

[2] The online supplementary file provides an illustrative example for how to construct such a relocation strategy.



(a) Two interaction ways in existing study.

(b) Three interaction ways considered in our study.

Fig. 3: The interaction between a newly sampled data and existing sampled data. If the newly sampled data falls into the region containing samples of the other category, the covering of this new sample will be offset as shown in (b).

than the factor σ_+^2 for category '+1'. The variance factor determines the relative probability density. The above designed distribution cases suggest that the data scatter still seems more deterministic than the volume factor on performance and subsequently the weights when class imbalance is concerned.

3.2.2 Analysis for Limitation 2

Limitation 2 concerns Assumption 2 in which only two interaction ways are assumed between a newly selected sample and all existing samples, as depicted in Fig. 3(a). However, this assumption overlooks the presence of multiple categories, which could lead to interactions between samples from different categories. While overlaps between samples from the same category merely maintain the current expected volume (i.e., E_{n-1}), overlaps between samples from different categories can also preserve the current expected volume (i.e., E_{n-1}). As a result, the third interaction way, offset, should not be disregarded.

We consider a straightforward random covering scenario, illustrated in Fig. 3(b). When a new sample is taken, it can overlap with existing data from the same category with the probability p_1, or with existing data from another category with the probability p_2, or falls entirely outside with the probability $1 - p_1 - p_2$. Then we have

$$\begin{aligned} &E_1 = 1 - p_2 \\ &E_n = p_1 E_{n-1} + (1-p_1)[p_2 E_{n-1} + (1-p_2)(E_{n-1}+1)] \end{aligned}. \tag{12}$$

According to inference with Eq. (12), we have

$$E_n = (1-p_2)\frac{1-{\beta'_c}^{n_c}}{1-\beta'_c}, \tag{13}$$

where $\beta'_c = \beta_c + (1-\beta_c)p_2$. The inference is presented in the supplementary file. When $p_2 \equiv 0$, E_n is reduced to the original calculation defined in Eq. (3); when $p_2 \equiv 1$, E_n is reduced to 0. These two extreme cases are reasonable. In a concrete learning task, let $p(y|\boldsymbol{x})$ be the posterior probability of category y on a sample $\boldsymbol{x}$. The value of p_2 for the sample $\boldsymbol{x}$ with label y can be calculated as follows:

$$p_2(\boldsymbol{x}) = 1 - p(y|\boldsymbol{x}). \tag{14}$$

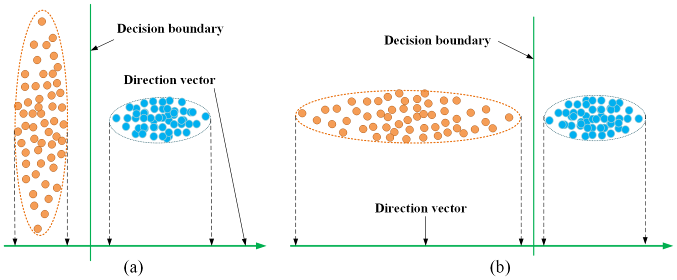


Fig. 4: Two binary classification tasks in $\mathbb{R}^2$. The raw distribution characteristics do not exactly reflect the scatter degrees of each class. As only the horizontal dimension takes effects in classification for both tasks, the variances of the samples projected to the horizontal dimension can reflect the scatter of each class. Therefore, the scatter degree of the orange class is smaller than that of the blue class in (a), and the scatter degree of the orange class is larger than that of the blue class in (b).

If $p(y|\boldsymbol{x}) \equiv 1$, then $p_2(\boldsymbol{x}) \equiv 0$, meaning that there is no overlap between different categories. Accordingly, the third interaction way does not exist in this task and E_n in Eq. (13) is reduced to the original effective number.

3.3 The Data Scatter Measurement

As data scatter is revealed to be more deterministic than data volume in several typical cases, this subsection proposes the measurement of the data scatter. In scenarios involving the bounded uniform or the bounded Gaussian distributions, the category with a low probability density tends to exhibit inferior performance compared to the other category. The low probability density leads to sampled data with either low density or high scatter. However, utilizing probability density directly is not appropriate for two main reasons. First, the probability density is unknown in training and the estimation is also a challenging problem. Second, in Gaussian distributions, the probability densities of different samples vary, making it difficult to derive category-wise weights based on the varied probability densities.

As previously stated, a low probability density will result in a large scatter degree. Covariance matrix can reflect the scatter degree of sampled data. However, covariance matrix is not a single value, and thus, it cannot be directly utilized. Further, not all the raw distribution information is useful in scatter measurement. The two learning tasks in Fig. 4 illustrate that only the variance of projected samples to a specific direction is useful for scatter measurement..

In LDA [35] for binary classification tasks, scatter is quantified by the variance of the features mapped from the original feature space to the classification boundary parameterized by $\boldsymbol{w}$. Let $\boldsymbol{\Sigma}_y$ be the covariance matrix for category y. Let $\boldsymbol{w}$ be the coefficient vector of a given linear boundary. In LDA, the data scatter for category y concerning $\boldsymbol{w}$ in a binary task can be evaluated using $\boldsymbol{w}^T\boldsymbol{\Sigma}_y\boldsymbol{w}$ with the constraint that $\boldsymbol{w}^T\boldsymbol{w} = 1$. This constraint ensures the scatter degree remains unchanged when the direction of $\boldsymbol{w}$ is fixed. As the constraint cannot be guaranteed in deep learning, the following modified measure is used:

$$\mathcal{DS}_y = \boldsymbol{w'}^T\boldsymbol{\Sigma}_y\boldsymbol{w'} = \frac{\boldsymbol{w}^T}{||\boldsymbol{w}^T\boldsymbol{w}||_2}\boldsymbol{\Sigma}y\frac{\boldsymbol{w}}{||\boldsymbol{w}^T\boldsymbol{w}||_2} = \frac{\boldsymbol{w}^T\boldsymbol{\Sigma}_y\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{w}}. \tag{15}$$

Naturally, $\boldsymbol{w'}^T\boldsymbol{w'} = 1$. Under this measure, the scatter degree remains unchanged when the direction of $\boldsymbol{w}$ is fixed. During the deep learning training process, a direction vector can be obtained between each pair of categories at each epoch, so this vector is used to measure the scatter degree of each category using Eq. (15). By applying this metric, the scatter degrees of the categories in the learning cases listed in Section 3.2.1 can be calculated and the results are found to be reasonable. Take the learning case described by Eq. (8) as an example. Let $\boldsymbol{w} = [1, \cdots, 1]^d$. $\mathcal{DS}_{+1} = \sigma_+^2$ and $\mathcal{DS}_{-1} = \sigma_-^2$. Then $\mathcal{DS}_{+1} > \mathcal{DS}_{-1}$. For the task in Fig. 4, the relative scatter orders between the two classes in terms of either the average probability density or the norm of covariance matrix are identical. Nevertheless, the relative scatter orders are different yet reasonable according to our measure. Therefore, our measure is better than them.

The above metric is only suitable for binary classification tasks. In the case of a multi-class task (containing C categories), we can first measure the scatter of a category in terms of each of the remaining categories. Alternatively, the scatter for category y consists of $C-1$ values:

$$\mathcal{DS}_{y,z} = \frac{(\boldsymbol{\omega}_y - \boldsymbol{\omega}_z)^T \boldsymbol{\Sigma}_y (\boldsymbol{\omega}_y - \boldsymbol{\omega}_z)}{(\boldsymbol{\omega}_y - \boldsymbol{\omega}_z)^T (\boldsymbol{\omega}_y - \boldsymbol{\omega}_z)}, \qquad z = 1, \cdots, C, \tag{16}$$

where $\boldsymbol{\omega}_y$ and $\boldsymbol{\omega}_z$ are the weight coefficients in the softmax layer for the yth and the zth categories, respectively. $\boldsymbol{\omega}_y - \boldsymbol{\omega}_z$ is the direction vector[3] of the linear boundary between the two categories. Consequently, each category corresponds to $C-1$ scatter degrees. The average of the $C-1$ scatter degrees can be leveraged to measure the overall scatter degree of a category giving the direction vector.

4 New Theory and Methods

This section firstly describes our novel theory of effective number-based sample reweighting for imbalanced learning. The new methodology is then introduced.

4.1 Our New Theory

The existing ENum weighting theory is founded on the Assumptions 1-4 in Section 3.1. According to our analysis on their limitations, new theory is established.

In our new theory, we retain Assumptions 1 and 2 particularly including the identical volume assumption as it is nearly impossible to calculate the volumes accurately. Building upon the preceding discussion, we present the following assumptions to supplant Assumptions 3 and 4.

Assumption 5. There are three ways[4] that a newly sampled data can interact with existing sampled data: overlapped with one of the existing data in the same category with the probability p_1, or, offset with one of the existing data from a different category with the probability $(1-p_1)p_2$, or, outside with the probability $(1-p_1)(1-p_2)$.

Assumption 6. The weight of category y depends on two factors: the effective number (E_{n_y}) of the training samples in category y and the data scatter degree of category y. When the effective numbers are equal, the weights vary normally with the data scatter for each category; when the data scatter degrees are equal, the weights vary inversely with the effective number for each category.

Assumptions 1, 2, 5, and 6 consist of the basis of our new theory. Nevertheless, in real tasks, the probability p_2 in Eq. (13) is not a constant value. It varies at different positions/samples of the category. As a result, obtaining a concise formula for E_n becomes challenging, given that p_2 relies on $\boldsymbol{x}$. However, we can prove the following lemma:

Lemma 1. If the volume of each newly sampled data $\boldsymbol{x}$ in category y is enlarged by $\frac{1}{p(y|\boldsymbol{x})}$, then E_{n_y} for category y is still $\frac{1-\beta_y^{n_y}}{1-\beta_y}$.

The proof is available at the online supplementary material. According to Lemma 1, effective number-based weighting can be modified to $\frac{1-\beta}{1-\beta^{n_{y_i}}} \times \frac{1}{p(y_i|\boldsymbol{x}_i)}$. Thereafter, a theoretical weight can be calculated for each training sample as follows

$$w(\boldsymbol{x}_i) \propto \overline{\mathcal{DS}}_{y_i} \times \frac{1-\beta}{1-\beta^{n_{y_i}}} \times \frac{1}{p(y_i|\boldsymbol{x}_i)}, \tag{17}$$

where $\overline{\mathcal{DS}}_{y_i} = \frac{1}{C-1}\Sigma_z \mathcal{DS}_{y_i,z}$ is the average data scatter for category y_i. The term $\frac{1}{p(y_i|\boldsymbol{x}_i)}$ is used according to Lemma 1. β_{y_i} becomes β as Assumption 2 is inherited and V_{y_i}s are identical for all categories. In real training process, $p(y_i|\boldsymbol{x}_i)$ can be approximated by the softmax output of the current trained model that may be imperfect, and samples may be noisy, so we use a function $g(\boldsymbol{x}_i, y_i)$ to replace $\frac{1}{p(y_i|\boldsymbol{x}_i)}$. The following theoretical weight is then used

$$w(\boldsymbol{x}_i) \propto \overline{\mathcal{DS}}_{y_i} \times \frac{1-\beta}{1-\beta^{n_{y_i}}} \times g(\boldsymbol{x}_i, y_i). \tag{18}$$

When $g(\boldsymbol{x}_i, y_i) \propto [1 - p(y_i|\boldsymbol{x}_i)]^r$, the whole weight integrates the Focal loss. However, Focal loss is sensitive to noisy samples [13], so a slight modification will be introduced. The weights in Formula (18) are dynamic.

Assumptions 1, 2, 5, and 6, and the weighting mechanism described by Formula (18) consist of our new ENum weighting theory for imbalanced learning. In the succeeding part, the concrete form of $g(\cdot)$ is presented.

4.2 The Proposed Method

4.2.1 Two New Losses

Noisy-label training samples commonly exist in real applications and thus their predictions $p(y|\boldsymbol{x})$ may be quite low. Consequently, their weights may become excessively large if Formula (17) is used, which is harmful for the training process. Inspired by noisy-label learning, we first define

$$g(\boldsymbol{x}_i, y_i) = \left\{ \begin{array}{ll} [1 - p(y_i|\boldsymbol{x}_i)]^r, & \text{if } p(y_i|\boldsymbol{x}_i) \geq \tau \\ {[1 - \bar{p}(y_i)]^r}, & \text{else} \end{array} \right. , \tag{19}$$

where r and τ[5] are two hyper-parameters; $\bar{p}(y_i)$ is the average prediction probability for category y. Eq. (19) indicates that when the prediction (i.e., $p(y_i|\boldsymbol{x}_i)$) for a particular

[3] Note that that the samples in the linear boundary between the two categories y_1 and y_2 should satisfy $\boldsymbol{\omega}_{y_1}^T\boldsymbol{x} = \boldsymbol{\omega}_{y_2}^T\boldsymbol{x}$. Therefore, the linear boundary is $(\boldsymbol{\omega}_{y_1} - \boldsymbol{\omega}_{y_2})^T\boldsymbol{x} = 0$, and thus the direction is $\boldsymbol{\omega}_{y_1} - \boldsymbol{\omega}_{y_2}$.

[4] We do not consider the case that a category's volume may be reduced by a sample sampled from a different category, as this consideration results in the entire analysis considerably complex.

[5] In our experiments, to avoid grid-searching on too many hyper-parameters, r is directly set as 2 and τ is set as 0.2.

sample is rather low, it can be considered as noisy or quite difficult and thus its weight is reduced by replacing $p(y_i|\boldsymbol{x}_i)$ with $\bar{p}(y_i)$ to alleviate the negative effect of this sample. The Cross-entropy loss with the weights obtained by (18) and (19) forms a new loss called NENum loss in this study.

The NENum loss still exhibits two limitations. First, the data scatter measure is initially designed for binary tasks, as shown in Eq. (16). The average $(\overline{\mathcal{DS}}_{y_i})$ alone cannot fully capture the data scatter relationships among different categories. Second, the function $g(\cdot)$ accounts for the overall offset effect from the rest categories, overlooking the individual contributions from each of the remaining categories. To address these two limitations, the idea of logit adjustment [36] is incorporated and the final training loss combining category-wise weighting and sample-wise logit perturbation becomes

$$\ell(f(\boldsymbol{x}_i), y_i) = -\frac{1-\beta}{1-\beta^{n_{y_i}}}\times \\ \log\frac{\exp\nu_{y_i}(\boldsymbol{x}_i)}{\sum_{z\in[C]}\exp[\nu_z(\boldsymbol{x}_i)+\tau_1 h(\boldsymbol{x}_i,z)+\tau_2\mathcal{DS}_{y_i,z}]}, \tag{20}$$

where τ_1 and τ_2 are two hyper-parameters; $\nu(\boldsymbol{x}_i)$ is the logit output of the backbone network; $h(\boldsymbol{x}_i,z)$ reflects the offset effects between $\boldsymbol{x}_i$ and z. Likewise, $h(\boldsymbol{x}_i,z)$ should be non-increasing over $p(z|\boldsymbol{x}_i)$, and if $p(z|\boldsymbol{x}_i)$ is larger than a certain threshold, $h(\boldsymbol{x}_i,z)$ should be low as the sample may be noisy. The concrete form of $h(\boldsymbol{x}_i,z)$ is approximated by using meta learning described in the succeeding part.

We emphasize that the Eq. (20) loss adheres to Assumptions 1, 2, 5, and 6. First, Assumption 1 is trivial and is certainly adopted. Second, β is equal for all categories indicating that the volumes of each category are still assumed to be identical as claimed in Assumption 2. Third, three interaction ways are considered as claimed in Assumption 5. Lastly, when $\tau_1\equiv 0$ and the larger the data scatter $\mathcal{DS}_{y_i,z}$, the larger the loss of the sample. As a result, the sample will play more important role in training. Alternatively, a large data scatter will result in a large weight, which is in accordance with Assumption 6.

We further elaborate the rationale behind utilizing the Eq. (20) loss, employing the first-order Taylor expansion for the Cross-entropy (CE) loss. We have

$$\ell(\boldsymbol{\nu}+\Delta\boldsymbol{\nu})\approx\ell(\boldsymbol{\nu})+(\frac{\partial\ell}{\partial\boldsymbol{\nu}})^\top\Delta\boldsymbol{\nu}=\ell(\boldsymbol{\nu})+(\boldsymbol{q}-\boldsymbol{y})^\top\Delta\boldsymbol{\nu}, \tag{21}$$

where $\boldsymbol{q}=\text{softmax}(\boldsymbol{\nu})$ and $\boldsymbol{y}$ is the one-hot label for category y. Then the regularizer $(\boldsymbol{q}-\boldsymbol{y})^\top\Delta\boldsymbol{\nu}$ becomes

$$\mathcal{R}(f(\boldsymbol{x}),y)=\sum_{z\neq y}q_z[\tau_1 h(\boldsymbol{x},z)+\tau_2\mathcal{DS}_{y,z}]. \tag{22}$$

A large value of $h(\boldsymbol{x},z)$ indicates that $\boldsymbol{x}$ could potentially be mis-classified into category z by the current trained model. Note that $q_z > 0$. Therefore, the regularizer in Eq. (22) further attempts to reduce the value of $p(z|\boldsymbol{x})$ $(z\neq y)$ and the data scatter of category y concerning category z if $h(\boldsymbol{x},z)$ is large. However, when $p(z|\boldsymbol{x})$ is considerably high, $\boldsymbol{x}$ may be noisy and its true label may be z. Theoretically, $h(\boldsymbol{x},z)$ should be small, and thus less regularization will be placed on $p(z|\boldsymbol{x})$.

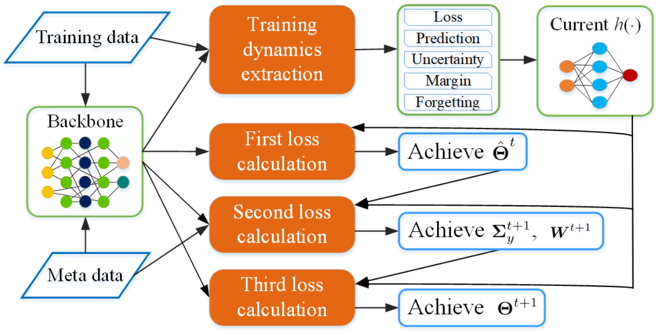

Training data
Backbone
Meta data
Training dynamics extraction
Loss
Prediction
Uncertainty
Margin
Forgetting
Current $h(\cdot)$
First loss calculation
Achieve $\hat{\boldsymbol{\Theta}}^t$
Second loss calculation
Achieve $\boldsymbol{\Sigma}_y^{t+1}$, $\boldsymbol{W}^{t+1}$
Third loss calculation
Achieve $\boldsymbol{\Theta}^{t+1}$

Fig. 5: The main pipeline of the meta learning-based training in a training epoch.

4.2.2 Meta Learning-based Training

To avoid explicitly defining the form of $h(\cdot)$ and to alleviate the inaccurate estimation of co-variance matrices (Σ_y) for data scatter, meta learning is employed.

Our meta learning-based training procedure follows the steps used in Meta-Weight-Net [15]. Let $\boldsymbol{\Theta}^t$ and $\boldsymbol{\Omega}^t$ (the parameters $\boldsymbol{W}$ for $h(\cdot)$ and the covariance matrices $\boldsymbol{\Sigma}_{\{y\}}$ for all categories) be the parameters of the backbone network and the meta-learning modular, respectively, in the tth epoch. Fig. 5 illustrates the pipeline of our training process in a training epoch which contains three main steps. In the first step, a batch of n_b training data is input into the backbone network. Their training dynamics are extracted to calculate $h(\cdot)$ and the data scatter for each class is calculated based on $\boldsymbol{\Omega}^t$. A temporal value of $\boldsymbol{\Theta}$ is then obtained by conventional gradient-based optimizing technique such as SGD as follows:

$$\hat{\boldsymbol{\Theta}}^t=\boldsymbol{\Theta}^t-\frac{\lambda_1}{n_b}\sum_{i=1}^{n_b}\nabla_{\boldsymbol{\Theta}}l(f_{\boldsymbol{\Theta}^t}(\boldsymbol{x}_i),y_i;\boldsymbol{\Sigma}_{\{y\}}^t,\boldsymbol{W}^t), \tag{23}$$

where λ_1 is the learning rate. In the second step, a batch of n_m meta data is input into the backbone network. Then, $\boldsymbol{\Omega}$ is updated based on $\hat{\boldsymbol{\Theta}}^t$ and the training loss on the meta data as follows:

$$\boldsymbol{\Sigma}_y^{t+1}=\boldsymbol{\Sigma}_y^t-\frac{\lambda_2}{n_m}\sum_{j=1}^{n_m}\nabla_{\boldsymbol{\Sigma}_y}l(f_{\hat{\boldsymbol{\Theta}}^t}(\boldsymbol{x}_j),y_j;\hat{\boldsymbol{\Theta}}^t,\boldsymbol{W}^t) \tag{24}$$

and

$$\boldsymbol{W}^{t+1}=\boldsymbol{W}^t-\frac{\lambda_3}{n_m}\sum_{j=1}^{n_m}\nabla_{\boldsymbol{W}}l(f_{\hat{\boldsymbol{\Theta}}^t}(\boldsymbol{x}_j),y_j;\hat{\boldsymbol{\Theta}}^t,\boldsymbol{\Sigma}_{\{y\}}^{t+1}), \tag{25}$$

where λ_2 and λ_3 are the learning rates. In the third step, both $h(\cdot)$ and the data scatter are updated according to $\boldsymbol{\Omega}^{t+1}$. The backbone parameters $\boldsymbol{\Theta}$ are then updated based on SGD and the new training loss as follows:

$$\boldsymbol{\Theta}^{t+1}=\boldsymbol{\Theta}^t-\frac{\lambda_1}{n_b}\sum_{i=1}^{n_b}\nabla_{\boldsymbol{\Theta}}l(f_{\boldsymbol{\Theta}^t}(\boldsymbol{x}_i),y_i;\boldsymbol{\Sigma}_{\{y\}}^{t+1},\boldsymbol{W}^{t+1}). \tag{26}$$

We use a three-layer MLP network to approximate $h(\boldsymbol{x}_i,z)$ depending on the following training dynamics, rather than the sole value of $p(z|\boldsymbol{x}_i)$, which may lead to inaccuracy:

- Training loss $(\boldsymbol{x}_{i,1}^m)$: The CE loss (-log$p(y_i|\boldsymbol{x}_i)$) is usually used to directly deduce the weight of a sample or to judge whether a sample is noisy or not [37].

Algorithm 1: Meta-ENum

```
Input: $D^{\text{train}}$, $D^{\text{meta}}$, $\beta$, $\tau_1$, $\tau_2$, batch size $n_b$, meta batch
  size $n_m$, $T_1$ and $T_2$, learning rates $\lambda_1$, $\lambda_2$, and $\lambda_3$.
Output: Trained backbone network $f_{\boldsymbol{\Theta}}$.

Initialize $h_{\boldsymbol{W}}$ and backbone $f_{\boldsymbol{\Theta}}$;
for $t = 1$ to $T_1$ do
    Sample a batch of samples from $D^{\text{train}}$;
    Calculate the standard CE loss on these samples;
    Update $\boldsymbol{\Theta}$ using SGD;
end for
for $t = T_1 + 1$ to $T_2$ do
    Sample $n_b$ (denoted as $\mathcal{B}_1 = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_b}$) and
      $n_m$ (denoted as $\mathcal{B}_2 = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^{n_m}$) samples from $D^{\text{train}}$
      and $D^{\text{meta}}$, respectively;
    Calculate covariance matrices $\boldsymbol{\Sigma}_c$ for each class;
    Extract $\{\boldsymbol{x}_{i,1}^{m}, \cdots, \boldsymbol{x}_{i,5}^{m}\}$ for each training data;
    Calculate $\mathcal{DS}_{c,z}$ by Eq. (16) for all categories;
    Calculate loss by Eq. (20) for all samples in $B_1$; and
      update temporary $\hat{\boldsymbol{\Theta}}$ by Eq. (23);
    Fixed $\hat{\boldsymbol{\Theta}}$ and $\boldsymbol{W}$, calculate loss by Eq. (20) for all data in
      $B_2$, and update $\boldsymbol{\Sigma}_{\{c\}}$ for all categories by Eq. (24);
    Fixed $\hat{\boldsymbol{\Theta}}$ and $\boldsymbol{\Sigma}_{\{c\}}^{t+1}$, calculate loss by Eq. (20) for all
      samples in $B_2$, and update $\boldsymbol{W}$ by Eq. (25);
    Fixed $\boldsymbol{W}^{t+1}$ and $\boldsymbol{\Sigma}_{\{c\}}^{t+1}$, calculate loss by Eq. (20) for all
      samples in $B_1$, and update $\boldsymbol{\Theta}$ by Eq. (26) .
end for
```

- Prediction ($\boldsymbol{x}_{i,2}^{m}$): This quantity ($p(z|\boldsymbol{x}_i)$) is also used for sample weighting in previous literature [3].
- Uncertainty ($\boldsymbol{x}_{i,3}^{m}$): The information entropy of the softmax output quantifies uncertainty [38]. Samples with high uncertainties are usually close to the classification boundary [43].
- Margin ($\boldsymbol{x}_{i,4}^{m}$): Margin measures the distance from the sample to the classification boundary between two categories. The margin is calculated by $(\omega_{y_i} - \omega_z)^T f(\boldsymbol{x}_i)$.
- Forgetting ($\boldsymbol{x}_{i,5}^{m}$): Sample forgetting is revealed to be an effective quantity to characterize training samples [44]. Forgetting measures the proportions of times that $(\boldsymbol{\omega}_{y_i} - \boldsymbol{\omega}_z)^T f(\boldsymbol{x}_i) < 0$ occurs in previous epochs.

The whole method is called Meta-ENum and the algorithmic steps are shown in Algorithm 1.

5 Experiments

In addition to the imbalance datasets, standard datasets are also leveraged to assess the performance of the proposed methodology. The reason for leveraging standard datasets lies in that the imbalance in terms of data scatter is also considered in our study.

5.1 Results on Imbalance Datasets

Four benchmark datasets are involved including the imbalance versions of CIFAR10 (i.e., CIFAR10-LT) and CIFAR100 (i.e., CIFAR100-LT) [51] and two large datasets iNaturalist 2017 (iNat2017) [49] and iNaturalist 2018 (iNat2018) [50]. The training/validation/testing configurations in [23], [34], [36] are followed and the details are as follows:

- **CIFAR10-LT/CIFAR100-LT** In the standard CIFAR10/CIFAR100 data sets, both have 50,000 training images and 10,000 testing images. Menon et al. [36] compiled imbalanced CIFAR corpora under different imbalance ratios (the ratio between the numbers of the head and the tail categories). In this study, two imbalance ratios are considered, namely, 10:1 and 100:1, as the compiled corpora were released by Menon et al. The numbers of training samples in the four sets are: 20391 (CIFAR10 (10:1)), 12380 (CIFAR10 (100:1)), 19541 (CIFAR100 (10:1)), and 19829 (CIFAR100 (100:1)). The original test sets are used. Following [23], ten samples per class from the original training set are randomly selected to construct the validation set (also the meta dataset)[6].
- **iNaturalist 2017/2018** iNat2017 has 579,184 training images and 5,089 categories with an imbalance ratio of 3919:9. iNat2018 has 435,713 images and 8,142 categories with an imbalance ratio of 500:1. Following [23], five and two random images per class from the training sets of iNat2017 and iNat2018 are selected to construct the meta data, respectively. The meta sets are also fixed according to the fixed random seeds provided by Li et al. [23].

The following classical and SOTA methods are compared by adopting the same settings for hyper-parameters used in previous studies [22], [36][7]:

- **Class-balanced CE loss** [11] This method assigns a weight to each category with $w_y = \frac{1-\beta}{1-\beta^{n_y}}$. The value of β is searched in {0.9, 0.99, 0.999, 0.9999}.
- **Class-balanced fine-tuning** [45] This method consists of two stages. In the first stage, a neural network is trained on the whole imbalanced training set. In the second stage, the network is fine-tuned on a balanced subset of the training set.
- **Meta-weight net** [15] This method assigns each sample a weight which is inferred by meta learning. Hyper-parameters for the meta-learning module are described in our supplementary file.
- **Focal loss** [3] This method determines the weights of each sample according to the formula $w_i = (1-p_i)^{\gamma}$, where p_i is the Softmax output on the true label and γ is searched in {0.5, 1, 2}.
- **Class-balanced focal loss** [11] This method combines class-balanced loss and focal loss using the following weighting scheme: $w_i = \frac{1-\beta}{1-\beta^{n_{y_i}}}(1-p_i)^{\gamma}$, where β and γ are set as 0.999 and 0.1, respectively.
- **LDAM** [26] As described in Section 2.3, LDM perturbs the logit vector of each sample in training. The perturbation on the yth quantity of the logit vector is $\delta_{y_i,y} = -log(\pi_{y_i}^{1/4}/\lambda)$ for the training sample x_i, where λ is set as 0.5 on CIFAR corpora and 0.3 for iNaturalist corpora.
- **ISDA + Dropout** [22] This method also perturbs the logit vector of each sample in training. The

[6] The released codes of [23] provide fixed random seeds which ensure that the same validation set is used for each competing method.

[7] Some recent classical methods are not in the same family as our proposed method, so they are not involved in the comparison.

perturbation on the yth quantity of the logit vector is $\delta_{y_i,y} = \lambda(\omega_y - \omega_{y_i})^T \mathbf{\Sigma}_{y_i}(\omega_y - \omega_{y_i})$, where λ is searched in {0.1, 0.25, 0.5, 0.75} for CIFAR corpora and {1, 2.5, 5.0, 7.5} for iNaturalist corpora.

- **LA** [36] This method perturbs the yth quantity of the logit vector of a training sample x_i using $\delta_{y_i,y} = \tau log \frac{\pi_{y_i}}{\pi_y}$, where τ is set as 1.
- **MetaSAug** [23] This method is an improvement of ISDA, as ISDA performs poor in imbalanced learning. It learns $\mathbf{\Sigma}_{\{y\}}$s for each class using meta learning. The algorithmic parameter λ is searched in {0.25, 0.5, 0.75, 1.0}.
- **LPL** [16] This method infers the perturbation vector for the logit of a training sample via an optimization process. The critical components include category set split threshold τ and perturbation bound ϵ. τ determines the categories that should be positively or negatively augmented, while ϵ determines the augmentation extent. The value of τ is searched in {0.4C, 0.5C, 0.6C}. The ϵ is related to three concrete hyper-parameters, namely, ϵ_c, $\Delta\varepsilon$, and α. The value of ϵ_c is set to 0, $\Delta\varepsilon$ is searched in {1.5, 2.5, 5}, and α is searched in {0.1, 0.2, 0.3}.
- **KPS** [46] This method modifies the logits using both perturbation and scaling. First, the logit quantity on the label y is perturbed via $m_y = m'_y \frac{m_{max}}{maxm'}$, where m'_y is dependent of n_y and a hyper-parameter n_{max}; second, all the logit quantities are multiplied a factor s. There are three hyper-parameters in KPS, namely, n_{max}, m_{max}, and s. They are set as 50, 15, and 0.5, respectively, on the CIFAR corpora and 50, 15, and 0.3, respectively, on the iNaturalist corpora.
- **NENum loss** This is the first proposed loss defined in Formulas (18) and (19). Its hyper-parameters are set according to Cui et al. [11], with r set to 2 and β set to 0.9999. ν in Eq. (19) is set to 0.2.
- **Meta-ENum** This is the second proposed loss defined in Eq. (20) based on meta learning. Its hyper-parameters are set according to Shu et al. [15]. The number of the middle nodes in the MLP is 100. SGD is used over a total of 240 epochs on the CIFAR corpora. The initial learning rate (λ_1) is set to 0.1 and decayed at the 160th and 200th epochs with a factor of 0.1. The momentum and the weight decay is set to 0.9 and 5e-4, respectively. In Meta-ENum, $h(\cdot)$ and $\mathbf{\Sigma}_{\{y\}}$ are optimized at the same time, with batch size set to 100, τ_1 set to 1.0 and τ_2 set to 1.0. For $\mathbf{\Sigma}_{\{y\}}$, since the gradient for $\mathbf{\Sigma}_{\{y\}}$ in Algorithm 1 is quite small during training, λ_2 is set to 1e2 for CIFAR-10-LT and 1e3 for CIFAR-100-LT, respectively. For $\boldsymbol{W}$ in $h(\cdot)$, optimization is performed using the Adam optimizer alone, with λ_3 set to 0.001 following Shu et al. [15]. T_1 is set as 160, and the other settings follow the study of MetaSAug. On the iNaturalist Corpora, all hyper-parameters and settings in the CIFAR experiments are mainly retained, except for λ_2, which is set to 1e3 for both iNat2017 and iNat2018.

Both the Class-balanced CE and the Class-balanced focal losses are directly derived from the ENum weighting theory [11]. The rest hyper-parameters for each listed method (including the base neural networks) are presented in the supplementary file.

TABLE 2: Test Top-1 errors on CIFAR100-LT (ResNet-32).

Ratio	100:1	10:1
Class-balanced CE loss	61.23%	42.43%
Class-balanced fine-tuning	58.50%	42.43%
Meta-weight net	58.39%	41.09%
Focal loss	61.59%	44.22%
Class-balanced focal loss	60.40%	42.01%
LDAM	59.40%	42.71%
LDAM-DRW	57.11%	41.22%
ISDA + Dropout	62.60%	44.49%
LA	56.11%	41.66%
MetaSAug	53.13%	38.27%
LPL	55.75%	39.03%
KPS	54.97%	40.16%
NENum loss	54.46%	40.39%
Meta-ENum	**52.08**%	**37.33**%

TABLE 3: Test Top-1 errors on CIFAR10-LT (ResNet-32).

Ratio	100:1	10:1
Class-balanced CE loss	27.32%	13.10%
Class-balanced fine-tuning	28.66%	16.83%
Meta-weight net	26.43%	12.45%
Focal loss	29.62%	13.34%
Class-balanced focal loss	25.43%	12.52%
LDAM	26.45%	12.68%
LDAM-DRW	25.88%	11.63%
ISDA + Dropout	26.45%	12.98%
LA	22.33%	11.07%
MetaSAug	19.46%	10.56%
LPL	22.05%	10.59%
KPS	18.77%	10.95%
NENum loss	20.75%	10.91%
Meta-ENum	**17.92**%	**9.83**%

The experimental results reported by Li et al. [23] for some of the above competing methods are directly adopted. The training settings are fixed. Similar to their experimental settings, ResNet-32 [47] is used as the backbone neural network for the two CIFAR datasets. The average top-1 error of five repeated runs is presented. Let n_{test} be the test size. The top-1 error is defined as follows

$$\text{Top-1 error} = \frac{\sum_{i=1}^{n_{test}} I(y_{predict} \neq y_i)}{n_{test}}. \tag{27}$$

In a single comparison, a smaller top-1 error indicates a better performance.

Tables 2 and 3 show the top-1 errors of all the involved methods on CIFAR10-LT and CIFAR100-LT, respectively. Our method, Meta-ENum, outperforms all other competing methods, including another meta-learning based approach, MetaSAug. Our proposed direct weighting method NENum loss also achieves good results. It is inferior or comparable to MetaSAug, KPS, and LPL on the four datasets. Nevertheless, it is better than the rest competing methods. It outperforms Class-balanced focal loss, indicating that the data scatter considered in our loss is useful.

In iNat2017 and iNat2018, the results of some competing methods reported in [16] are directly borrowed. ResNet-50 [47] is used as the backbone neural network following the setting of [48]. Table 4 presents the top-1 errors of all involved methods on the iNat2017 and iNat2018 datasets. Similar conclusions are obtained. Our proposed method Meta-ENum still achieves the lowest top-1 errors on both datasets. NENum loss still achieves comparable results and outperforms most existing methods including Class-

TABLE 4: Test Top-1 errors on two real-world datasets (ResNet-50).

Method	iNat2017	iNat2018
Class-balanced CE loss	42.02%	33.57%
Class-balanced fine-tuning	41.77%	34.16%
Meta-weight net	37.48%	32.50%
Focal loss	38.98%	32.69%
Class-balanced focal loss	41.92%	38.88%
LDAM	39.15%	34.13%
LDAM-DRW	37.84%	32.12%
ISDA + Dropout	43.37%	39.92%
LA	36.75%	31.56%
MetaSAug	38.47%	32.06%
LPL	35.86%	30.59%
KPS	35.56%	29.65%
NENum loss	36.88%	31.67%
Meta-ENum	**35.02**%	**29.31**%

balanced focal loss and another meta learning-based method MetaSAug.

5.2 Results on Standard Datasets

The standard versions of CIFAR10-LT and CIFAR100-LT are involved in this part including CIFAR10 and CIFAR100. In both corpora, there are 50,000 images for training and 10,000 images for testing. Following [23], ten random samples per class from training set to construct the validation set for both data sets. Li et al. [23] provided the random seeds in their released codes, so the two validation sets are fixed. The validation data are also taken as meta data.

The following classical and SOTA sample weighting loss functions and logit perturbation-based losses are compared by adopting the same settings for hyper-parameters used in previous studies:

- **Large Margin** [52] This method replaces the logit quantity $(\boldsymbol{\omega}_y^T f_y(x))$ of the true label y with $||\boldsymbol{\omega}_y||||f_y(x)||\psi(\theta_y)$, where ψ is a function of θ_y with a hyper-parameter m, which is set as 0.2.
- **Disturb label** [53] This method actually disturbs the true label of each training sample with a probability α. For each disturbed sample, its label is randomly drawn from a uniform distribution over all the possible labels. α is set as 0.05 in Wide-ResNet-28-10 on both CIFAR-10 and CIFAR-100 datasets and ResNet-110 on CIFAR 10, while it is set as 0.1 for ResNet-110 on CIFAR-100.
- **Center loss** [54] This method adds a regularizer to the CE loss. The regularizer for a training sample is defined as $Reg(x) = \lambda||Net(x) - \bar{c}_y||_2^2$, λ is a hyper-parameter and $\bar{c}_y$ is the mean feature of the class y. λ is searched in {0.0001, 0.001, 0.01, 0.1, 1.0}.
- **Lq loss** [56] This method defines a new loss using $l(x) = \frac{(1-f_y(x)^q)}{q}$, where $q \in (0,1]$ is a hyper-parameter. Following [56], q is set as 0.4.
- **LPL** The value of τ is set as the average of the average predictions on the true label for samples in each category. ϵ_c is set as the average prediction on the true label for each category, $\Delta\varepsilon$ is searched in {0.1, 0.2}, and α is searched in {0.01, 0.02, 0.03}.
- **ISDA/ISDA+Dropout/MetaSAug/NENum/Meta-ENum** These methods have been briefly described in Section 5.1. The same settings for the hyper-parameters are adopted.

TABLE 5: Mean values and standard deviations of the test Top-1 errors for all the involved methods on CIFAR10.

Method	WRN-28-10	ResNet-110
Basic	3.82 ± 0.15%	6.76 ± 0.34%
Large Margin	3.69 ± 0.10%	6.46 ± 0.20%
Disturb Label	3.91 ± 0.10%	6.61 ± 0.04%
Focal loss	3.62 ± 0.07%	6.68 ± 0.22%
Center loss	3.76 ± 0.05%	6.38 ± 0.20%
Lq loss	3.78 ± 0.08%	6.69 ± 0.07%
ISDA	3.60 ± 0.23%	6.33 ± 0.19%
ISDA + Dropout	3.58 ± 0.15%	5.98 ± 0.20%
MetaSAug	3.85 ± 0.33%	7.22 ± 0.34%
LPL	3.37 ± 0.04%	5.72 ± 0.05%
NENum loss	3.64 ± 0.09%	6.28 ± 0.12%
Meta-ENum	**2.80 ± 0.06%**	**5.14 ± 0.04%**

Some methods such as KPS compared in Section 5.1 are not involved as they are particularly designed for imbalanced learning. The rest hyper-parameters for each listed method are presented in the supplementary file. Wide-ResNet-28-10 (WRN-28-10) [58] and ResNet-110 [47] are used as the base neural networks. The top-1 errors reported in the LPL paper [16] for the above competing methods are presented directly as the training/testing configuration is identical for both sets. Some results are directly from the original papers of the competing algorithms. The experimental settings for the base neural networks follow the descriptions given in the ISDA paper [22] and the released codes. The top-1 error is leveraged as the evaluation metric.

Tables 5 and 6 show the top-1 errors of the involved methods on the two standard datasets CIFAR10 and CIFAR100, respectively. Meta-ENum achieves the lowest top-1 errors on both datasets under two different backbone DNN architectures. Note that although Meta-ENum is based on a meta set, the meta set is compiled from the original training set without requiring any additional human labeling. Therefore, the comparison is fair for all the competing methods. The results indicate that our methodology considers the data scatter imbalance which is effective for benchmark datasets that are not considered as imbalance in terms of sample proportion. Our proposed NENum loss does not achieve competitive performance when ResNet-110 is used on CIFAR10 and WRN-28-10 is used on CIFAR100. The reason lies in that our NENum loss is particularly designed for the imbalance scenario.

TABLE 6: Mean values and standard deviations of the test Top-1 errors for all the involved methods on CIFAR100.

Method	WRN-28-10	ResNet-110
Basic	18.53 ± 0.07%	28.67 ± 0.44%
Large Margin	18.48 ± 0.05%	28.00 ± 0.09%
Disturb Label	18.56 ± 0.22%	28.46 ± 0.32%
Focal loss	18.22 ± 0.08%	28.28 ± 0.32%
Center loss	18.50 ± 0.25%	27.85 ± 0.10%
Lq loss	18.43 ± 0.37%	28.78 ± 0.35%
ISDA	18.12 ± 0.20%	27.57 ± 0.46%
ISDA + Dropout	17.98 ± 0.15%	26.35 ± 0.30%
MetaSAug	18.61 ± 0.29%	28.75 ± 0.22%
LPL	17.61 ± 0.30%	25.42 ± 0.07%
NENum loss	18.27 ± 0.28%	26.19 ± 0.40%
Meta-ENum	**16.31 ± 0.15%**	**24.86 ± 0.13%**

5.3 Ablation Study

5.3.1 *Ablation Study for NENum*

In all the experiments, our NENum loss achieves superior results compared with the Class-balanced CE loss, indicating that the considered data scatter holds significant meaning. Our NENum loss modifies the Focal loss part as described in Eq. (20). This modification aims to enhance the robustness of our NENum loss, especially when dealing with noisy labels in the training data, as noisy labels are nearly unavoidable in real learning tasks. To this end, we compare our NENum loss with its simplified version when the original Focal-loss part is used as follows:

$$\boldsymbol{w}(\boldsymbol{x}_i) \propto \overline{\mathcal{DS}}_{y_i} \times \frac{1-\beta}{1-\beta^{n_{y_i}}} \times [1 - p(y_i|\boldsymbol{x}_i)]^r. \tag{28}$$

Further, a more simplified version is also considered as follows:

$$\boldsymbol{w}(\boldsymbol{x}_i) \propto \overline{\mathcal{DS}}_{y_i} \times \frac{1-\beta}{1-\beta^{n_{y_i}}}. \tag{29}$$

Tables 7 and 8 present the comparison among the NENum loss, its simplified version (referred to as NENum-), and the more simplified version (referred to as NENum–), based on the four imbalanced datasets. Our modified parts including both the introduced data scatter and $g(\boldsymbol{x}, z)$ are useful. Note that $g(\boldsymbol{x}, z)$ is introduced with the illumination of Assumption 5 and the data scatter factor is inspired by Assumption 6. The comparison results support the two assumptions.

TABLE 7: The comparison results among NENum, NENum-, and NENum- on CIFAR10-LT and CIFAR100-LT.

	CIFAR10-LT		CIFAR100-LT	
Ratio	100:1	10:1	100:1	10:1
NENum	**20.75%**	**10.91%**	**54.46%**	**40.39%**
NENum-	22.89%	11.37%	56.10%	42.06%
NENum–	24.63%	12.92%	58.89%	42.28%

TABLE 8: The comparison results among NENum, NENum-, and NENum- on iNat2017 and iNat2018.

	iNat2017	iNat2018
NENum	**36.88%**	**31.67%**
NENum-	38.07%	33.18%
NENum–	40.64%	33.32%

5.3.2 Ablation Study for Meta-ENum

In our Meta-ENum loss, as defined in Eq. (21), there are two perturbation terms including $h(\boldsymbol{x}, z)$ and $\mathcal{DS}_{y,z}$. Therefore, we conduct experiments to exam the significance of these two terms. Tables 9 and 10 show the results of Meta-ENum when one of the two terms is removed on the four imbalanced datasets. In addition, the results for the variation without both $h(x, z)$ and $\mathcal{DS}_{y,z}$ are also presented. Results validate the importance of both terms.

TABLE 9: The comparison results among Meta-ENum and its variations on CIFAR10-LT and CIFAR100-LT.

	CIFAR10-LT		CIFAR100-LT	
Ratio	100:1	10:1	100:1	10:1
Meta-ENum	**17.92%**	**9.83%**	**52.08%**	**37.33%**
-$h(\boldsymbol{x}, z)$	18.65%	11.28%	54.31%	38.75%
-$\mathcal{DS}_{y,z}$	19.23%	10.95%	55.65%	39.27%
-$h(x, z)$-$\mathcal{DS}_{y,z}$	27.32%	13.10%	61.23%	42.43%

We also examine whether the meta learning is truly useful. Therefore, we replace the two terms ($h(x, z)$ and $\mathcal{DS}_{y,z}$) with their non-meta-learning versions. Specifically, $h(x, z)$ is replaced by $g(x, z)$ and Σ_y in $\mathcal{DS}_{y,z}$ is calculated by the contained training samples in the class. Results are shown in Tables 11 and 12. The results on these four datasets indicate that meta learning does take effect in our method.

TABLE 10: The comparison results among Meta-ENum and its variations on iNat2017 and iNat2018.

	iNat2017	iNat2018
Meta-ENum	**35.02%**	**29.31%**
-$h(\boldsymbol{x}, z)$	35.99%	30.31%
-$\mathcal{DS}_{y,z}$	36.74%	31.14%
-$h(x, z)$-$\mathcal{DS}_{y,z}$	42.02%	33.57%

TABLE 11: The comparison between Meta-ENum and its non-meta-learning version on CIFAR10-LT and CIFAR100-LT.

	CIFAR10-LT		CIFAR100-LT	
Ratio	100:1	10:1	100:1	10:1
Meta-ENum	**17.92%**	**9.83%**	**52.08%**	**37.33%**
non-meta-learning	18.76%	10.73%	52.60%	38.28%

5.4 Discussion

5.4.1 Statistical test

To obtain a reliable comparison conclusion for the involved competing methods, the significance test is utilized with the Friedman test [57], which can be used to verify whether the performance differences among at least three methods on multiple datasets are significant. On the imbalanced datasets, there are fourteen competing methods. The test value T_F calculated by the Friedman test is 37.52, which is larger than the critical value (10.64) when $\alpha = 0.05$ (α is the significance level). On the standard datasets, there are twelve competing methods. The T_F value calculated by the Friedman test is 29.07, which is larger than the critical value (7.81) when $\alpha = 0.05$. Both results reject the hypothesis that all competing methods have equal performance. In other words, the performance differences among the involved competing methods are significant.

5.4.2 Sensitivity Analysis on τ_1 and τ_2

The loss in Meta-ENum contains two hyper-parameters, namely, τ_1 and τ_2. In the aforementioned experiments, their values are directly set to 1 to avoid excessive burden on grid search. This part analyzes the sensitivity of the learning performance concerning these two hyper-parameters. Fig. 6(a) shows the performance variations under different values of τ_1 on CIFAR10 (Backbone uses WRN-28-10) and CIFAR100-LT (100:1), while τ_2 is fixed as 1. Fig. 6(b) shows the performance variation with different τ_2 values for the same dataset scenario while τ_1 is fixed to 1. The results reveal that fine-tuning the hyper-parameters leads to further improvements in model performance. Nonetheless, even with both τ_1 and τ_2 fixed at 1, the model still achieves competitive results compared to the more detailed tuning approach.

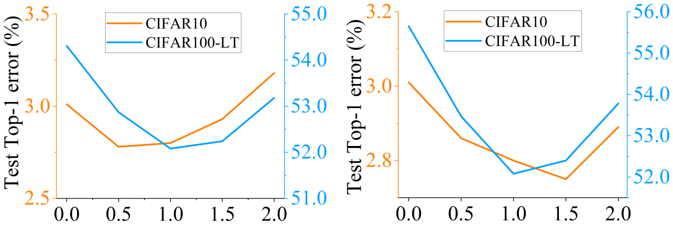
5.4.3 Analysis on the Sample-wise Weights

We make a statistic on the weights generated by our proposed NENum loss on CIFAR10-LT (10:1) and CIFRAR100-LT (10:1). For each set, we select one head category and

TABLE 12: The comparison between Meta-ENum and its non-meta-learning version on iNat2017 and iNat2018.

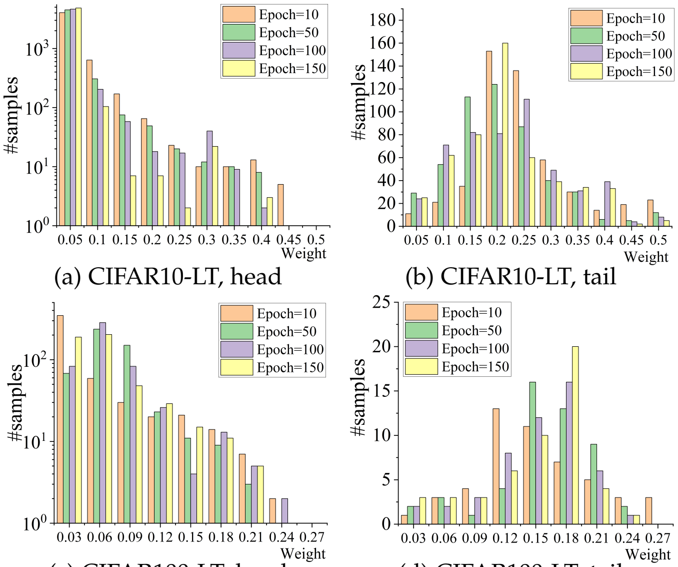
	iNat2017	iNat2018
Meta-ENum	**35.02%**	**29.31%**
non-meta-learning	36.34%	31.03%

one tail category. The weight distributions of the training samples in CIFAR10- LT and CIFAR100-LT, corresponding to the head and tail categories, are depicted in Fig. 7 at different epochs. Specifically, Fig. 7(a) and Fig. 7(b) show the distributions of the head and the tail categories in CIFAR10-LT, respectively, while Fig. 7(c) and Fig. 7(d) display the weight distributions for CIFAR100-LT. The analysis reveals that the overall weights of the head category tend to be relatively small, although there are still instances where large weights occur. On the other hand, the weights of the tail categories are relatively larger, whereas there are still samples with small weights in the tail categories.



(a) Fixed τ_1, search τ_2 (b) Fixed τ_2, search τ_1

Fig. 6: Sensitivity analysis of τ_1 and τ_2.



(a) CIFAR10-LT, head (b) CIFAR10-LT, tail

(c) CIFAR100-LT, head (d) CIFAR100-LT, tail

Fig. 7: Sample weight distribution of training data at different epochs.

6 Conclusion

This study has re-investigated the effective number theory for sample reweighting in imbalanced learning. The limitations of this existing theory are summarized. Another important factor, namely, data scatter, is defined to better capture the relationship between two categories, and another sample interaction way, namely, offset, is introduced to better model the random covering issue for imbalanced learning. New effective number-based imbalanced learning theory is then constructed, and a meta learning-based imbalanced learning method is proposed. Extensive experiments indicate the effectiveness of the proposed method. Effective number should not be the sole possible theory for imbalanced learning. Our future work will combine the effective number-based theory with existing promising theories to construct a more solid theory for imbalanced learning. In addition, as intrinsic relations exist between cost-sensitive learning and imbalanced learning, we also aim to explore more effective learning methods for them together.

References

[1] H. He and E. A. Garcia, "Learning from Imbalanced Data," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263-1284, 2009.

[2] Zhi-Hua Zhou and Xu-Ying Liu, "Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 63-77, 2006.

[3] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *CVPR*, pp. 2999–3007, 2017.

[4] K. R. M. Fernando and C. P. Tsokos, "Dynamically Weighted Balanced Loss: Class imbalanced learning and Confidence Calibration of Deep Neural Networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 99, pp. 2162–2388, 2021.

[5] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to Reweight Examples for Robust Deep Learning," in *ICML*, pp. 6900–6909, 2018.

[6] X. -y. Liu and Z. -h. Zhou, "The Influence of Class Imbalance on Cost-Sensitive Learning: An Empirical Study," in *IIEEE ICDM*, pp. 970-974, 2006.

[7] K. Yang, Z. Yu, C. L. P. Chen, W. Cao, J. You and H. -S. Wong, "Incremental Weighted Ensemble Broad Learning System for Imbalanced Data," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5809-5824, 2022.

[8] Y. Sun, L. Cai, B. Liao, W. Zhu and J. Xu, "A Robust Oversampling Approach for Class Imbalance Problem With Small Disjuncts," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 6, pp. 5550-5562, 2023.

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique", in *J. Artif. Intell. Res.*, vol. 16, pp. 321-357, 2002.

[10] G.M. Weiss, "Mining with rarity - problems and solutions: a unifying framework," in *SIGKDD Explorations*, vol.6, no.1, pp.7–19, 2004

[11] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, Serge Belongie, "Class-Balanced Loss Based on Effective Number of Samples", in *CVPR*, pp. 9260–9269, 2019.

[12] S. Janson., "Random Coverings in Several Dimensions", in *Acta Mathematica*, 1986.

[13] Xiaolin Zhou and Ou Wu, "Which Samples Should be Learned First, Easy or Hard?", in *IEEE Transactions on Neural Network and Learning System*, 2023.

[14] Zhining Liu, Pengfei Wei, Jing Jiang, Wei Cao, Jiang Bian, Yi Chang, "MESA: Boost Ensemble imbalanced learning with MEta-SAmpler", in *NeurIPS*, 2020.

[15] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, Deyu Meng, "Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting", in *NeurIPS*, pp. 1917–1928, 2019.

[16] Mengyang Li, Fengguang Su, Ou Wu, Ji Zhang, "Class-Level Logit Perturbation", in *IEEE Transactions on Neural Network and Learning System*, 2023.

[17] K.M. Ting, "A comparative study of cost-sensitive boosting algorithms," in *ICML*, pp.983–990, 2000.

[18] U. Brefeld, P. Geibel, and F. Wysotzki, "Support vector machines with example dependent costs," in *ICML*, pp.23– 34, 2003.

[19] Yoav Freund, Robert E. Schapire, "A decision-Theoretic Generalization of On-line Learning and an Application to Boosting", in *EuroCOLT*, pp. 23-37, 1995.

[20] Xin Wang, Yudong Chen, Wenwu Zhu, "A Survey on Curriculum Learning", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44, 9, 4555 - 4576, 2022.

[21] Anubha Kabra, Ayush Chopra, Nikaash Puri, Pinkesh Badjatiya, Sukriti Verma, Piyush Gupta, Balaji Krishnamurthy, "MixBoost: Synthetic Oversampling using Boosted Mixup for Handling Extreme Imbalance," in *IEEE ICDM*, pp. 1082-1087, 2020.

[22] Y. Wang, G. Huang,S. Song, X. Pan, Y. Xia, and C. Wu, "Regularizing Deep Networks With Semantic Data Augmentation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 3733-3748, 2021.

[23] S. Li, K. Gong, C. H. Liu, Y. Wang, F. Qiao, and X. Cheng, "Metasaug: Meta Semantic Augmentation for Long-tailed Visual Recognition," in *CVPR*, pp. 5212–5221, 2021.

[24] M.A. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown," in *ICML'03 Workshop on Learning from Imbalanced Data Sets*, 2003.

[25] M. Pawan Kumar, B. Packer, and D. Koller, "Self-paced Learning for Latent Variable Models," in *NeurIPS*, pp. 1–9, 2010.

[26] K. Cao, C. Wei, et al., "Learning Imbalanced Datasets with Label Distribution-aware Margin Loss," in *NeurIPS*, pp. 1567–1578, 2019.

[27] B. Zhou, Q. Cui, X. Wei, and Z. Chen, "BBN: Bilateral-branch Network with Cumulative Learning for Long-tailed Visual Recognition," in *CVPR*, pp. 9719–9728, 2020.

[28] T. Hospedales, A. Antoniou, P. Micaelli, A. Storkey, "Meta-Learning in Neural Networks: A Survey", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44, 9, 5149 - 5169, 2021.

[29] H. Xu, X. Liu, Y. Li, A.K. Jain, and J. Tang, "To be Robust or to be Fair: Towards Fairness in Adversarial Training", in *ICML*, 11492–11501, 2021.

[30] Z.-H. Zhou, X.-Y. Liu, "On multi-class cost-sensitive learning", Computational Intelligence, 26(3): 232-257, 2010.

[31] Y. Freund, and R. E. Schapire, "Experiments with a New Boosting Algorithm," in *ICML*, pp. 1–9, 1996.

[32] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum Learning," in *ICML*, pp. 41–48, 2009.

[33] Y. Cui, M. Jia, T. Lin, Y. Song, et al., "Class-Balanced Loss Based on Effective Number of Samples," in *CVPR*, pp. 9260–9269, 2019.

[34] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong, "Rethinking classbalanced methods for long-tailed visual recognition from a domain adaptation perspective," in *CVPR*, pp. 7610–7619, 2020.

[35] Trevor Hastie, Robert Tibshirani, Jerome Friedman, "The Elements of Statistical Learning, Data Mining, Inference, and Prediction," Springer, 2014.

[36] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail Learning Via Logit Adjustment," in *ICLR*, 2022.

[37] T. Castells, P. Weinzaepfel, and J. Revaud, "SuperLoss: A Generic Loss for Robust Curriculum Learning," in *NeurIPS*, pp. 1–12, 2020.

[38] Q. A. Wang, "Probability Distribution and Entropy as a Measure of Uncertainty," *J. Phys. A*, vol. 41, no. 6, pp. 065004, 2008.

[39] Kaihua Tang, Mingyuan Tao, Jiaxin Qi, Zhenguang Liu, and Hanwang Zhang, "Invariant Feature Learning for Generalized Long-tailed Classification", in *ECCV*, pp. 709–726, 2022.

[40] Z. Liu, P. Wei, Z. Wei, B. Yu, J. Jiang, W. Cao, J. Bian, and Y. Chang, "Towards Inter-class and Intra-class Imbalance in Class-imbalanced Learning," in CoRR abs/2111.12791, 2021.

[41] Yuxi Xie, Min Qiu, Haibo Zhang, Lizhi Peng, and Zhenxiang Chen, "Gaussian Distribution Based Oversampling for Imbalanced Data Classification," in *IEEE Transactions on Knowledge and Data Engineering*, Vol. 32, No. 2, pp. 667-679, 2022.

[42] Michael J. Siers, Md Zahidul Islam, "Class Imbalance and Cost-Sensitive Decision Trees: A Unified Survey Based on a Core Similarity," ACM Transactions on Knowledge Discovery from DataVolume, Issue 1, No. 4, pp. 1–31, 2020.

[43] Xiaoling Zhou, Ou Wu, Weiyao Zhu, Ziyang Liang, "Understanding Difficulty-Based Sample Weighting with a Universal Difficulty Measure," in *ECML/PKDD*, pp. 68-84, 2022.

[44] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, Geoffrey J. Gordon, "An Empirical Study of Example Forgetting during Deep Neural Network Learning," in *ICLR*, 2019.

[45] Y. Cui, Y. Song, et al., "Large Scale Fine-grained Categorization and Domain-specific Transfer Learning," in *CVPR*, pp. 4109–4118, 2018.

[46] M. Li, Y.-M. Cheung, and Z. Hu, "Key Point Sensitive Loss for Long- tailed Visual Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4812–4825, 2023

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, pp. 770–778, 2016.

[48] T. Wu, Q. Huang, Z. Liu, Y. Wang, and D. Lin, "Distribution-balanced Loss for Multi-label Classification in Long-tailed datasets," in *ECCV*, pp. 162–178, 2020.

[49] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The iNaturalist Species Classification and Detection Dataset," in *CVPR*, pp. 8769–8778, 2018.

[50] "iNaturalist 2018 competition dataset," https://github.com/visipedia/inat comp, 2018.

[51] A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features from Tiny Images," 2009.

[52] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin Softmax Loss for Convolutional Neural Networks." in *ICML*, pp. 507–516, 2016.

[53] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian, "DisturbLabel: Regularizing CNN on the Loss Layer," in *CVPR*, pp. 4753–4761, 2016.

[54] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A Discriminative Feature Learning Approach for Deep Face Recognition," in *ECCV*, pp. 499–515, 2016.

[55] Y. Chen, Z. Hong and X. Yang, "Cost-Sensitive Online Adaptive Kernel Learning for Large-Scale Imbalanced Classification," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 10, pp. 10554-10568, 2023.

[56] Z. Zhang and M. R. Sabuncu, "Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels," in *NeurIPS*, pp. 8778–8788, 2018.

[57] M. Friedman,"The use of ranks to avoid the assumption of normality implicit in the analysis of variance," in *J. Am. Stat. Assoc.*, 1937.

[58] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," in *BMVC*, pp. 87.1–87.12, 2016.

Ou Wu received the B.Sc. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2003, and the M.Sc. and Ph.D. degrees in computer science from the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2006 and 2012, respectively. In 2007, he joined NLPR as an Assistant Professor. In 2017, he became a Full Professor at the Center for Applied Mathematics, Tianjin University, China. His research interests include data mining and machine learning.

Mengyang Li received his B.Eng. degree from Zhengzhou University of Aeronautics, China, in 2015, and his M.Eng. degree from Civil Aviation University of China in Tianjin, China, in 2019. Currently, he is pursuing a Ph.D. degree at Tianjin University in Tianjin, China, under the supervision of Professor Ou Wu. His research interests include data mining and deep learning.

Supplementary Materials to Revisiting the Effective Number Theory for Imbalanced Learning

Ou Wu, Mengyang Li

1 Proof of Theorems 1 and 2

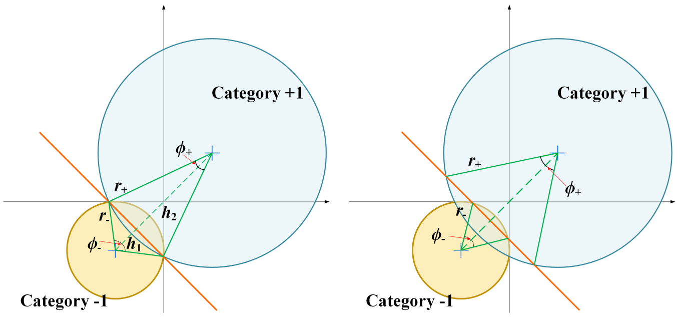


Fig. 1: The geometric relationships for error calculation when $p_+ : p_- = K^2 : 1$ (left) and $p_+ = p_-$ (right).

Assuming that the linear boundary is $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$, where $\boldsymbol{w} = [1, \cdots, 1]^T \in R^d$. Denoting $h_1 = \frac{|b - dr_+|}{\sqrt{d}}$ and $h_2 = \frac{b + dr_+}{\sqrt{d}}$. According to the formula of arched area, the error rate is as follows:

$$Err = \frac{r_+^2(\phi_+ - \sin\phi_+) + r_-^2(\phi_- - \sin\phi_-)}{2\pi(r_+^2 + r_-^2)}, \tag{1}$$

where $r_+ = \eta$, $r_- = K\eta$, $\phi_+ = 2arc\cos\frac{h_1}{r_+}$, $\phi_- = 2arc\cos\frac{h_2}{r_-}$, as shown in Fig. (1) left.

In Eq. (1), b is the variable. Therefore, we first derive the partial derivative as follows:

$$\frac{\partial(\phi_+ - \sin\phi_+)}{\partial b} = \frac{4\sqrt{1 - (\frac{h_1}{r_+})^2}}{r_+\sqrt{d}}, \tag{2}$$

and

$$\frac{\partial(\phi_- - \sin\phi_-)}{\partial b} = \frac{4\sqrt{1 - (\frac{h_2}{r_-})^2}}{r_-\sqrt{d}}, \tag{3}$$

Therefore, if $\frac{\partial Err}{\partial b} = 0$, then

$$\sqrt{r_+^2 - {h_1}^2} = \sqrt{r_-^2 - {h_2}^2}, \tag{4}$$

Eq. (4) denotes that the optimal linear boundary passes through the two intersection points of the two boundaries of the two categories as shown in Fig. 1 left. As $r_+ < r_-$, it is easy to conclude that

$$\phi_+ > \phi_-. \tag{5}$$

As the accuracy of each category only depends on the arc angle, so the following is obtained:

$$\mathcal{A}(f^*, +1) > \mathcal{A}(f^*, -1). \tag{6}$$

That is, Theorem 1 holds.

In Theorem 2, $p_+ = p_-$. If $\frac{\partial Err}{\partial b} = 0$, then we have

$$\frac{\sqrt{1 - (\frac{h_1}{r_+})^2}}{r_+} = \frac{\sqrt{1 - (\frac{h_2}{r_-})^2}}{r_-}, \tag{7}$$

Based on Eq. (7), we obtain

$$\frac{\sqrt{r_+^2 - h_1^2}}{r_+^2} = \frac{\sqrt{r_-^2 - h_2^2}}{r_-^2}. \tag{8}$$

The above equation indicates that

$$\frac{\sin\frac{\phi_+}{2}}{r_+} = \frac{\sin\frac{\phi_-}{2}}{r_-}. \tag{9}$$

Note that $r_+ < r_-$. Consequently, $\phi_+ < \phi_-$ is obtained as shown in Fig. 1 right. Likewise, as the accuracy of each category only depends on the arc angle, so the following is obtained:

$$\mathcal{A}(f^*, +1) < \mathcal{A}(f^*, -1). \tag{10}$$

That is, Theorem 2 is established.

2 An illustrative example for Section 3.2.2

Fig. 2 illustrates the construction of the distribution for category ′+1′. The circle is bounded by r_+. The distribution for category ′+1′ is derived from the original Gaussian distribution through the following steps. Firstly, data points that lie outside the circle but between lines l_a and l_b are uniformly relocated to the area within the circle and also between lines l_a and l_b. Secondly, data points situated in the lower-left region of line l_e and in the upper-right region of line l_c are uniformly moved to the upper-right semicircular area.

Similarly, category ′-1′ follows the same construction approach as described above. It is important to note that the

- *Ou Wu and Mengyang Li are with the Center for Applied Mathematics, Tianjin University, Tianjin, China, 300072.*
E-mail: {wuou, limengyang}@tju.edu.cn

Manuscript received July 31, 2023.

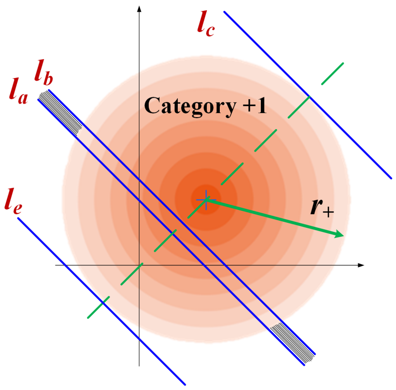


Fig. 2: The illustration for the construction of a bounded Gaussian distribution.

bounds for both categories are sufficiently large to ensure that the optimal linear classifier lies between the centers of the two circles. Consequently, as the distance between lines l_a and l_b approaches infinitesimal values, it becomes evident that the optimal linear classifier coincides with that for classification tasks where both categories conform to Gaussian distributions.

3 Inference for Eq. (13) in the paper

First we have $p_1 = E_{n-1}/V_c$. Based on

$$E_n = p_1E_{n-1} + (1-p_1)[p_2E_{n-1} + (1-p_2)(E_{n-1}+1)] \quad , \tag{11}$$

it is easy to obtain

$$\begin{aligned} E_n &= E_{n-1} + (1-p_1)(1-p_2) \\ &= (1 - \frac{1-p_2}{V_c})E_{n-1} + (1-p_2), \\ &= \beta'_c E_{n-1} + (1-p_2) \end{aligned} \tag{12}$$

As $E_1 = 1 - p_2$ and according to Eq. (12), assuming $E_{n-1} = (1-p_2)\frac{1-{\beta'_c}^{n-1}}{1-\beta'_c}$ holds, then

$$\begin{aligned} E_n &= \beta'_c(1-p_2)\frac{1-{\beta'_c}^{n-1}}{1-\beta'_c} + (1-p_2) \\ &= (1-p_2)\frac{\beta'_c - {\beta'_c}^{n} + 1 - \beta'_c}{1-\beta'_c} \\ &= (1-p_2)\frac{1-{\beta'_c}^{n}}{1-\beta'_c} \end{aligned} \quad . \tag{13}$$

4 Proof of Lemma 1

The condition is that when the newly sampled data is $\boldsymbol{x}$ and its volume is englarged by $\frac{1}{p(y|\boldsymbol{x})}$. Then the volume of the new sample $\boldsymbol{x}$ is $\frac{1}{p(y|\boldsymbol{x})}$.

The second formula of Eq. (12) in the paper becomes

$$\begin{aligned} E_n &= p_1E_{n-1} + (1-p_1)\mathbb{E}_x[(1-p(y|\boldsymbol{x}))E_{n-1} \\ &+ p(y|\boldsymbol{x})(E_{n-1} + \frac{1}{1-p(y|\boldsymbol{x})})] \\ &= p_1E_{n-1} + (1-p_1)\mathbb{E}_x[E_{n-1}+1] \\ &= E_{n-1} + (1-p_1) \end{aligned} \quad , \tag{14}$$

which is exactly the iterative formula of the existing effective number-based theory. Consequently, the effective number of category y is still $\frac{1-\beta_y^{n_y}}{1-\beta_y}$.

5 Hyper-parameters setting for the competing methods

The rest hyper-parameters of the competing methods are as follows:

- **Class-balanced CE loss** [6] The algorithmic parameter β is searched in $\{0.9, 0.99, 0.999, 0.9999\}$. The training hyper-parameters on CIFAR10 and CIFAR100 are set as follows. The epoch, batch size, weight decay, and momentum are set as 200, 128, 0.0005, and 0.9, respectively. The learning rate is initialized as 0.1 and decayed by 0.01 at the 160 epochs and again at 180 epochs. The training hyper-parameters on iNat2017 and iNat2018 are set as follows. The epoch, batch size, weight decay, and momentum are set as 90, 1024, 0.0005, and 0.9, respectively. The learning rate is initialized as and decayed by 0.01 at the 30 epochs and again at 60 epochs. We used linear warm-up of learning rate [2] in the first 5 epochs.
- **Class-balanced fine-tuning** [3] The training hyper-parameters on CIFAR10 and CIFAR100 are set as follows. The epoch, batch size, weight decay, and momentum are set as 200, 128, 0.0005, and 0.9, respectively. On iNat2017 and iNat2018, the epoch, batch size, weight decay, and momentum are set as 90, 1024, 0.0005, and 0.9, respectively.
  The learning rate is set as follows. In the first stage, the neural network is trained on the entire imbalanced training set. The initial learning rate is set to 0.045, with exponential decay of 0.94 after every two epochs. In the second stage, the neural network is fine-tuned on a balanced subset of the training set[1]. The initial learning rate is lowered to 0.0045 with the learning rate decay of 0.94 after every 4 epochs.
- **Meta-weight net** [4] The training hyper-parameters on CIFAR10 and CIFAR100 are set as follows. The epoch, batch size, weight decay, and momentum are set as 100, 100, 0.0005, and 0.9, respectively. The learning rate for the backbone network is initialized as 0.1 and decayed by 0.1 at the 80 epochs and again at 90 epochs. The learning rate in meta learning is set as 0.00001. The training hyper-parameters on iNat2017 and iNat2018 are set as follows. The epoch, batch size, weight decay, and momentum are set as 20, 100, 0.0005, and 0.9, respectively. The learning rate for the backbone network is set as 0.0001, and the learning rate for meta learning is set as 0.00001.
- **Focal loss** [5] The algorithmic parameter γ is searched in $\{0.5, 1, 2\}$. The training hyper-parameters on CIFAR10 and CIFAR100 are set as follows. The epoch, batch size, weight decay, and momentum are set as 200, 128, 0.0005, and 0.9, respectively. The learning rate for the backbone network is initialized as 0.1 and decayed by 0.1 at the 160 epochs and again at 180 epochs. The training hyper-parameters on iNat2017 and iNat2018 are set as follows. The epoch, batch size, weight decay, and momentum are set as

[1] The paper [3] does not describe how to construct the balanced subset. Our results for this method are directly borrowed from previous studies mentioned in the paper.

90, 1024, 0.0005, and 0.9, respectively. The learning rate is initialized as and decayed by 0.01 at the 30 epochs and again at 60 epochs. We also used linear warm-up of learning rate [2] in the first 5 epochs.

- **Class-balanced focal loss** [6] The algorithmic parameters γ and β are set as 0.5 and 0.999, respectively. The training parameters are set with the same setting as Class-balanced CE loss.
- **LDAM** [7] On the CIFAR corpora, the epoch, batch size, weight decay, and momentum are set as 200, 128, 0.0004, and 0.9, respectively. The learning rate is initialized as 0.1 and decayed by 0.1 at the 160 epochs and again at 180 epochs. The training hyper-parameters on iNat2017 and iNat2018 are set as follows. The epoch, batch size, weight decay, and momentum are set as 90, 1024, 0.0005, and 0.9, respectively. The learning rate is initialized as and decayed by 0.01 at the 30 epochs and again at 60 epochs. We also used linear warm-up of learning rate [2] in the first 5 epochs.
- **ISDA + Dropout** [8] TThe dropout rate is set as 0.3. On the CIFAR corpora, the epoch, batch size, weight decay, and momentum are set as 160, 128, 0.0004, and 0.9, respectively. The learning rate is initialized as 0.1 and decayed by 0.1 at the 80 epochs and again at 120 epochs. On iNat2017 and iNat2018, the training hyper-parameters on CIFAR10 and CIFAR100 are set as follows. The epoch, batch size, weight decay, and momentum are set as 300, 512, 0.0004, and 0.9, respectively. The learning rate is set as 0.2 and cosine schedule.
- **LA** [9] The training hyper-parameters on CIFAR10 and CIFAR100 are set as follows. The epoch, batch size, weight decay, and momentum are set as 200, 128, 0.0004, and 0.9, respectively. The learning rate is initialized as 0.1 and decayed by 0.1 at the 160 epochs and again at 180 epochs. The training hyper-parameters on iNat2017 and iNat2018 are set as follows. The epoch, batch size, weight decay, and momentum are set as 90, 512, 0.0004, and 0.9, respectively. The learning rate is set as 0.4. We also used linear warm-up of learning rate in the first 5 epochs.
- **MetaSAug** [10] The training hyper-parameters on CIFAR10 and CIFAR100 are set as follows. The epoch, batch size, weight decay, and momentum are set as 200, 100, 0.0004, and 0.9, respectively. The learning rate is initialized as 0.1 and decayed by 0.1 at the 160 epochs and again at 180 epochs. The training hyper-parameters on iNat2017 and iNat2018 are set as follows. The epoch, batch size, weight decay, and momentum are set as 20, 64, 0.0004, and 0.9, respectively. The learning rate is set as 0.4.
- **LPL** [11] For CIFAR10 and 100, the epoch, batch size, weight decay, and momentum are set as 200, 100, 0.0004, and 0.9, respectively. The learning rate is initialized as 0.1 and decayed by 0.1 at the 160 epochs and again at 180 epochs. For the iNaturalist corpora, the epoch, batch size, weight decay, and momentum are set as 90, 512, 0.0004, and 0.9, respectively. The learning rate is set as 0.4. We also used linear warm-up of learning rate in the first 5 epochs.
- **KPS** [13] For the CIFAR corpora, the epoch, batch size, weight decay, and momentum are set as 200, 64, 0.0004, and 0.9, respectively. The learning rate is initialized as 0.1 and decayed by 0.1 at the 160 epochs and again at 180 epochs. linear warm-up of learning rate in the first 5 epochs. For the iNaturalist corpora, the epoch, batch size, weight decay, and momentum are set as 180, 512, 0.0004, and 0.9, respectively. The learning rate is initialized as 0.2 and decayed by 0.1 at the 160 epochs and again at 180 epochs. We also used linear warm-up of learning rate in the first 5 epochs.
- **NENum loss** The rest hyper-parameters follow the setting used in LPL [11].
- **Meta-ENum** The rest hyper-parameters for the backbone network follow the setting used in LPL [11]. The rest hyper-parameters for meta learning follow the setting used in Meta-weight net [4].

For the experiments on the standard datasets, all method are implemented with almost the same training configurations. The epoch, batch size, weight decay, and momentum are set as 160, 128, 0.0001, and 0.9, respectively. The learning rate is initialized as 0.1 and decayed by 0.1 at the 80 epochs and again at 120 epochs. For the center loss, its initial learning rate is set as 0.5.

References

[1] Zhi-Hua Zhou and Xu-Ying Liu, "Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 63-77, 2006.

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

[3] Y. Cui, Y. Song, et al., "Large Scale Fine-grained Categorization and Domain-specific Transfer Learning," in *CVPR*, pp. 4109–4118, 2018.

[4] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, Deyu Meng, "Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting", in *NeurIPS*, pp. 1917–1928, 2019.

[5] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *CVPR*, pp. 2999–3007, 2017.

[6] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, Serge Belongie, "Class-Balanced Loss Based on Effective Number of Samples", in *CVPR*, pp. 9260–9269, 2019.

[7] K. Cao, C. Wei, et al., "Learning Imbalanced Datasets with Label Distribution-aware Margin Loss," in *NeurIPS*, pp. 1567–1578, 2019.

[8] Y. Wang, G. Huang,S. Song, X. Pan, Y. Xia, and C. Wu, "Regularizing Deep Networks With Semantic Data Augmentation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 3733-3748, 2021.

[9] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail Learning Via Logit Adjustment," in *ICLR*, 2022.

[10] S. Li, K. Gong, C. H. Liu, Y. Wang, F. Qiao, and X. Cheng, "Metasaug: Meta Semantic Augmentation for Long-tailed Visual Recognition," in *CVPR*, pp. 5212–5221, 2021.

[11] Mengyang Li, Fengguang Su, Ou Wu, Ji Zhang, "Class-Level Logit Perturbation", in *IEEE Transactions on Neural Network and Learning System*, 2023.

[12] R. Dror, S. Shlomov, and R. Reichart, "Deep Dominance - How to Properly Compare Deep Neural Models," in *Proc. ACL*, pp. 2773–2785, 2019.

[13] M. Li, Y.-M. Cheung, and Z. Hu, "Key Point Sensitive Loss for Long- tailed Visual Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4812–4825, 2023.