OPTIMAL TRANSPORT FOR LONG-TAILED RECOGNITION WITH LEARNABLE COST MATRIX

Hanyu Peng, Mingming Sun &Ping, Li
{penghanyu,sunmingming01,liping11}@baidu.com

ABSTRACT

It is attracting attention to the long-tailed recognition problem, a burning issue that has become very popular recently. Distinctive from conventional recognition is that it posits that the allocation of the training set is supremely distorted. Predictably, it will pose challenges to the generalisation behaviour of the model. Approaches to these challenges revolve into two groups: firstly, training-aware methods, with the aim of enhancing the generalisability of the model by exploiting its potential in the training period; and secondly, post-hoc correction, liberally coupled with training-aware methods, which is intended to refine the predictions to the extent possible in the post-processing stage, offering the advantages of simplicity and effectiveness. This paper introduces an alternative direction to do the post-hoc correction, which goes beyond the statistical methods. Mathematically, we approach this issue from the perspective of optimal transport (OT), yet, choosing the exact cost matrix when applying OT is challenging and requires expert knowledge of various tasks. To overcome this limitation, we propose to employ linear mapping to learn the cost matrix without necessary configurations adaptively. Testing our methods in practice, along with high efficiency and excellent performance, our method surpasses all previous methods and has the best performance to date.

1 INTRODUCTION

Classification problems in the real world are generally challenged by the long-tailed label distribution, i.e., having a small number of samples for a majority of labels, and a dominant number of samples for a minority of labels (Van Horn & Perona, 2017; Buda et al., 2018; Liu et al., 2019). It is also known as imbalanced recognition, which has been widely studied in the past decades (Cardie & Howe, 1997; Chawla et al., 2002; Qiao & Liu, 2009; Cui et al., 2019). These distribution biases pose a significant challenge to predictive modeling; conceivably, models often suffer from poor generalisation and undesirable estimation bias (Cao et al., 2019; Kang et al., 2019; Zhou et al., 2020).

Recently, a renewed interest in the problem of long-tail recognition has emerged following the context of neural networks, as numerous publications in the literature endeavour to resolve the problem albeit in different ways including decouple (Kang et al., 2019), meta-learning (Ren et al., 2020; Wang et al., 2020a; Li et al., 2021), post-hoc correction (Tang et al., 2020; Hong et al., 2020), etc (Liu et al., 2019; Cao et al., 2019; Tang et al., 2020). One of the representative methods of post-hoc correction, Logit Adjustment Menon et al. (2020), provides a statistical correction to the prediction, receiving widespread attention for its simplicity and validity. But the downside is that its optimisation is conducted on individual samples, without any indication that rectified marginal distribution satisfies the desired distribution we are striving for.

Figuring out exact flaws of Logit Adjustment, our explicit modeling of the problem mathematically turns into an equational constraint, meanwhile to minimise the difference between refined distribution and the original one, this minimisation is motivated upon the inner-product similarity. A little further, the resulting problem can be linked to OT. Drawing on this linkage, we develop it further by proposing a linear mapping to automatically learn cost matrix, thereby circumventing the requirement for expert knowledge to configure this matrix.

In short, we contribute as follows:

- We propose an alternative direction based on convex optimisation to do post-hoc correction, which goes beyond previous direction from the statistical view.
- Imposing marginal distributions to align ideal marginal distributions, we derive an optimisation problem tied to the OT that is solved using Sinkhorn, a highly efficient algorithm. More further, for better learning of the cost matrix, we present a linear mapping enabling very elegant learning with one-layer neural network.
- The experimental evidence shows the high efficiency of the algorithm, and the best performance achieved on three benchmarks. It also supports our claim that addressing the post-hoc problem via OT is helpful and effective.

2 PRELIMINARIES

In this section, we begin by giving a notational definition, followed by an introduction to the long-tailed recognition problem, and finally, we briefly review the OT, and give a quick recap of Logit Adjustment Menon et al. (2020).

Notations: In what follows, for two matrices $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{N \times K}$, we denote $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \sum_{n=1}^{N} \sum_{k=1}^{K} \boldsymbol{X}_{nk} \boldsymbol{Y}_{nk}$ as the Frobenius dot-product. $\delta(\cdot)$ stands for the Dirac function, $p(\cdot)$ represents the probability distribution. $U(\boldsymbol{r}, \boldsymbol{c}) = \{\boldsymbol{P} \in \mathbb{R}_{+}^{N \times K} | \boldsymbol{P} \mathbf{1}_K = \boldsymbol{r}, \boldsymbol{P}^{\intercal} \mathbf{1}_N = \boldsymbol{c}\}$, where $\mathbf{1}_N$ and $\mathbf{1}_K$ are N-dimension and K-dimension vector whose elements are all 1. $\boldsymbol{r}$ and $\boldsymbol{c}$ refer to the vectors of size N and K respectively, $U(\boldsymbol{r}, \boldsymbol{c})$ include all matrices with row and column sums $\boldsymbol{r}$ and $\boldsymbol{c}$ respectively.

2.1 PROBLEM FORMULATION

Having a collection of training samples $\{(\boldsymbol{x}_n^s, y_n^s)\}_{n=1}^{N_s}$, validation samples $\{(\boldsymbol{x}_n^v, y_n^v)\}_{n=1}^{N_v}$ and test samples $\{(\boldsymbol{x}_n^t, y_n)^t\}_{n=1}^{N_t}$ for classification with K labels and input $\boldsymbol{x} \in \mathbb{R}^d$, long-tailed recognition assumes that the class-prior distribution for training data $p(y^s)$ is different from that for validation data $p(y^v)$ and test data $p(y^t)$. Specifically, *long-tailed recognition means the distribution $p(y^s)$ is highly skewed, that is, some classes have the dominant number of samples, while tailed labels own a very small number of samples*. We can use imbalance ratio to measure the skewness in training data set, which can be defined as $R = \frac{N_{max}^s}{N_{min}^s}$, where N_{max}^s and N_{min}^s denote the largest and smallest number of samples in the training data set, respectively. *In this paper, we assume that the marginal distribution of the test set is known, we consider it as an implicit prior knowledge to be applied*. Stepping back, even if we do not know the marginal distribution of the test dataset in advance. There are still ways to estimate the marginal distribution of the test dataset relatively precisely, such as methods in Hendrycks et al. (2018); Azizzadenesheli et al. (2019).

Obviously, most models trained on imbalanced training data set would suffer from extremely limited generalisation ability. Hence the ultimate goal is to learn a model that minimises the empirical risk, which can be formulated as:

$$\mathcal{J}\left(\Phi\left(\boldsymbol{x}_n^s\right), y_n^s\right) = \frac{1}{N_s} \sum_{n=1}^{N_s} \mathcal{L}\left(\Phi(\boldsymbol{x}_n^s), y_n^s\right), \tag{1}$$

where $\Phi(\boldsymbol{x}_n^s) \in \mathbb{R}^K$ denotes logits with associated sample, $\Phi(\cdot) : \mathbb{R}^d \to \mathbb{R}^K$ represents the mapping via neural networks, $\mathcal{L}$ stands for the loss function, typically cross entropy loss function for classification problem.

2.2 REMINDERS ON OPTIMAL TRANSPORT

OT is used to calculate the cost of transporting one probability measure to another. We next present a brief introduction to OT with the aim of helping us better view the long-tailed problem from an OT perspective.

Consider probability measures r and c of the two random variables X and Y respectively, when the cost function $\mathcal{C}(X, Y) : X \times Y \to \mathbb{R}_+$ is defined to denote the cost of transporting X to Y, the OT distance can be defined as :

$$d(r, c) = \min_{\boldsymbol{\pi} \in \Pi(r,c)} \int_{X \times Y} \mathcal{C}(\boldsymbol{x}, \boldsymbol{y}) \pi(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y}, \tag{2}$$

where $\Pi(r,c) = \left\{ \int_{\mathcal{Y}} \pi(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{y} = r, \int_X \pi(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x} = c \right\}$ is the joint probability measure with r and c. When we extend the above continuous case to the discrete situation, we consider following discrete distributions:

$$\boldsymbol{r} = \sum_{i=1}^{N} p_i(\boldsymbol{x}_i)\delta(\boldsymbol{x}_i) \quad \boldsymbol{c} = \sum_{j=1}^{K} p_i(\boldsymbol{y}_j)\delta(\boldsymbol{y}_j) \tag{3}$$

where $p_i(\boldsymbol{x}_i)$ and $p_i(\boldsymbol{y}_j)$ represent the probability mass to the sample $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ respectively. In this context, OT distance can be expressed as:

$$d_{\boldsymbol{M}}(\boldsymbol{r},\boldsymbol{c}) = \min_{\boldsymbol{P}\in U(\boldsymbol{r},\boldsymbol{c})} \langle \boldsymbol{P}, \boldsymbol{M} \rangle. \tag{4}$$

where $\boldsymbol{M}$ stands for the cost matrix constructed by $\boldsymbol{M}_{ij} = \mathcal{C}(\boldsymbol{x}_i, \boldsymbol{y}_j)$. The goal of OT is to find a transportation matrix $\boldsymbol{P}$ that minimizes the distance $d_{\boldsymbol{M}}(\boldsymbol{r},\boldsymbol{c})$

As we can see, OT is a distance measure between two probability distributions under some cost matrix (Villani, 2008). However, when we use network simplex or interior point methods to solve the above optimisation problem, it often comes at the cost of heavy computational demands. To tackle this issue, OT with entropy constraint is proposed to allow the optimisation at small computational cost in sufficient smoothness (Cuturi, 2013). By adding a Lagrangian multiplier to the entropy constraint, the new formulation can be defined as follows:

$$d^{\lambda}_{\boldsymbol{M}}(\boldsymbol{r},\boldsymbol{c}) = \langle \boldsymbol{P}^{\lambda}, \boldsymbol{M} \rangle \qquad \text{where} \quad \boldsymbol{P}^{\lambda} = \operatorname*{arg\,min}_{\boldsymbol{P}\in U(\boldsymbol{r},\boldsymbol{c})} \langle \boldsymbol{P}, \boldsymbol{M} \rangle - \lambda h(\boldsymbol{P}), \tag{5}$$

where $\lambda \in [0, +\infty]$, $h(\boldsymbol{P}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} \boldsymbol{P}_{nk} \log \boldsymbol{P}_{nk}$, $d^{\lambda}_{\boldsymbol{M}}(\boldsymbol{r},\boldsymbol{c})$ is also known as dual-Sinkhorn divergence, besides, it can be calculated with matrix scaling algorithms for cheaper computational demand. The following lemma guarantees the convergence and uniqueness of the solution of the algorithm.

Lemma 1 *For $\lambda > 0$, the solution $\boldsymbol{P}^{\lambda}$ is unique and has the form $\boldsymbol{P}^{\lambda} = \boldsymbol{diag}(\boldsymbol{u})\boldsymbol{K}\boldsymbol{diag}(\boldsymbol{v})$, where $\boldsymbol{u}$ and $\boldsymbol{v}$ are two non-negative vectors uniquely defined up to a multiplicative factor and $\boldsymbol{K} = e^{-\boldsymbol{M}/\lambda}$ is the element-wise exponential of $-\boldsymbol{M}/\lambda$.*

The above lemma states the uniqueness of $\boldsymbol{P}^{\lambda}$ (Sinkhorn, 1974), and $\boldsymbol{P}^{\lambda}$ can be efficiently computed via Sinkhorn's fixed point iteration $\boldsymbol{u}, \boldsymbol{v} \leftarrow \boldsymbol{r}./\boldsymbol{K}\boldsymbol{v}, \boldsymbol{c}./\boldsymbol{K}^{\intercal}\boldsymbol{u}$.

2.3 A Quick Recap of Logit Adjustment

We give a brief introduction to Logit Adjustment (Menon et al., 2020; Hong et al., 2020). For the model $\Phi(\cdot)$, it is trained by the standard cross-entropy loss function on imbalanced training data set, and evaluated on test data. In this algorithm, the test logit is adjusted as follows:

$$\Phi(\boldsymbol{x}_n^t) = \Phi(\boldsymbol{x}_n^t) - \log p(y^s) \tag{6}$$

This simple procedure is derived from the Bayes optimal rule. It is apparent that Logit Adjustment involves a post hoc correction on an individual sample, which does not necessarily guarantee that the marginal distribution of the whole dataset matches the desired distribution.

3 Methodology

The first part of this section explores post-hoc correction from an OT perspective, proceeds to the automatic learning of the cost matrix via linear mapping. Lastly, we demonstrate how it can be achieved simply with one-layer neural network.

3.1 Post-hoc Correction Formalised from An OT Perspective

Since Logit Adjustment applies adjustment at the individual sample level. It doesn't assure that the marginal distribution of the overall data set fulfils our desired distribution. In this respect, we clearly put the constraint into an equation:

$$\boldsymbol{Y}^{\intercal}\boldsymbol{1}_N = \boldsymbol{\mu}, \tag{7}$$

where $\boldsymbol{Y} \in \mathbb{R}^{N \times K}$ indicates the refined prediction value in matrix form, $\boldsymbol{\mu}$ represents the expected distribution on the test set. Alternatively, it is desirable to preserve another characteristic of $\boldsymbol{Y}$, namely, remaining almost as similar to the original prediction as possible. We consider inner-product based similarity to measure this, which is a straightforward yet useful similarity measure. In mathematical terms, this can be expressed as follows:

$$\underset{\boldsymbol{Y}}{\text{maximize}} \quad \langle C(\hat{\boldsymbol{Z}}), \boldsymbol{Y} \rangle, \tag{8}$$

where $\hat{\boldsymbol{Z}}$ represents the original prediction in matrix form, $C(\cdot)$ denotes to some transformation to $\hat{\boldsymbol{Z}}$, it can be some simple function, like Logarithmic function $\log(z)$, exponential function z^{α}. Here we select $-\log(\cdot)$ as the cost function. This choice was driven by the requirement that the cost matrix must be positive definite, whereas the transformation of the original prediction by $-\log(\cdot)$ satisfies this condition. In addition, as log likelihood represents the local probability density of the associated samples, it can also be used to substitute $\hat{\boldsymbol{Z}}$ for the similarity approximation. In brief, the resulting numerical form can be put in formal terms as follows:

$$\underset{\boldsymbol{Y}}{\text{minimize}} \quad \langle -\log(\hat{\boldsymbol{Z}}), \boldsymbol{Y} \rangle \tag{9}$$

$$\text{subject to} \quad \boldsymbol{Y}^{\intercal}\mathbf{1}_N = \boldsymbol{\mu}, \quad \boldsymbol{Y}\mathbf{1}_K = \mathbf{1}_N. \tag{10}$$

Extra constraint on $\boldsymbol{Y}$ is imposed simply cos the tuned estimation has to fulfil the basic probabilistic requirement that its sum is one. Comparing Eq. (9-10) with Eq. (4), we can see that if we substitute $\boldsymbol{P}$ with $\boldsymbol{Y}$, and substitute $\boldsymbol{r}$ and $\boldsymbol{c}$ with $\mathbf{1}_N$ and $\boldsymbol{\mu}$ respectively, we find that the above optimisation problem is actually a special case of OT.

In preliminaries, the entropy regularised OT (EOT) is introduced. By adding entropy regularisation to OT, the given equation can be solved efficiently by Sinkhorn algorithm. Specifically, the optimisation problem is:

$$\begin{aligned} \underset{\boldsymbol{Y}}{\text{minimize}} \quad & \langle -\log(\hat{\boldsymbol{Z}}), \boldsymbol{Y} \rangle + \lambda \boldsymbol{Y} \log(\boldsymbol{Y}) \\ \text{subject to} \quad & \boldsymbol{Y}^{\intercal}\mathbf{1}_N = \boldsymbol{\mu}, \quad \boldsymbol{Y}\mathbf{1}_K = \mathbf{1}_N. \end{aligned} \tag{11}$$

The associated algorithmic flow for solving Eq. (11) is outlined in detail in Algorithm 1.

Algorithm 1: Solve OT-related algorithm efficiently in the post-hoc correction via Sinkhorn Algorithm.

Input: Cost matrix $\boldsymbol{M} = -\log(\hat{\boldsymbol{Z}})$, trade-off parameter λ, max number of iterations N_T, iteration number t, error threshold ϵ, current error σ, row and column sums $\boldsymbol{r} = \mathbf{1}_N$ and $\boldsymbol{c} = \boldsymbol{\mu}$, $|\cdot|$ denotes the vector norm.

Result: Refined predictions $\boldsymbol{Y}$

Initialise $\boldsymbol{K} = e^{-\boldsymbol{M}/\lambda}$,$\boldsymbol{u}_{old} = \mathbf{1}_N$, $\boldsymbol{v} = \mathbf{1}_K$, $t = 0$;
while $t \leq N_T$ *and* $\delta \leq \epsilon$ **do**
  $\boldsymbol{u} = \boldsymbol{r}./\boldsymbol{K}\boldsymbol{v}$;
  $\boldsymbol{v} = \boldsymbol{c}./\boldsymbol{K}^{\intercal}\boldsymbol{u}$;
  $\sigma = |\boldsymbol{u}_{old} - \boldsymbol{u}|$;
  $\boldsymbol{u}_{old} = \boldsymbol{u}$;
  $t = t + 1$;
end
Output $\boldsymbol{Y} = \mathbf{diag}(\boldsymbol{u})\boldsymbol{K}\mathbf{diag}(\boldsymbol{v})$

Assigning λ with 1, it is observed that we equate our objective function to the KL divergence, thus illustrating the extensive nature of our approach. DARP (Kim et al., 2020a) has previously applied it to long-tailed semi-supervised classification.

Remark We would like to illustrate the non-applicable scenarios of our method. Firstly, our method requires a large number of samples for evaluation. This is because if the batch size is small, we can not guarantee that the desired marginal distribution can be satisfied within the batch. In some online scenarios, the sample-wise correction method is more suitable. In addition, our method assumes that the marginal distribution is already known. In this paper, we assume in particular that it is consistent with a uniform distribution.

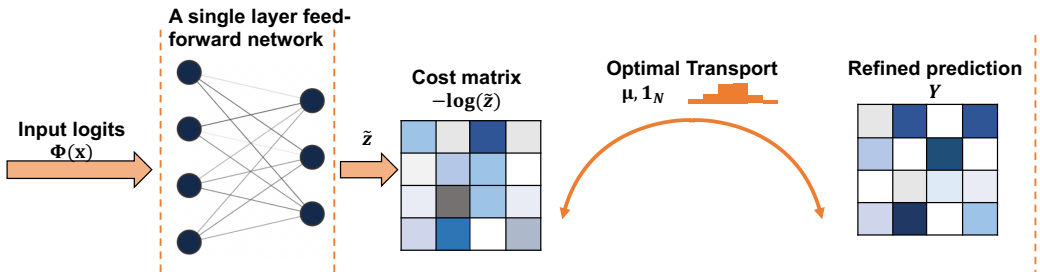

Figure 1: Our proposed framework OTLM. The logit $\Phi(\boldsymbol{x})$ is fed into a single-layer feed-forward network to infer $\tilde{\boldsymbol{Z}}$. The cost matrix $\boldsymbol{M}$ is set to $-\log \tilde{\boldsymbol{Z}}$. By solving the optimal transport problem via Sinkhorn algorithm, we can obtain the refined prediction $\boldsymbol{Y}$.

3.2 Cost Function Learning via Linear Mapping

Simple functions are likely to be sub-optimal for the real data sets; this suggests the design of a better cost function to better fit the long-tailed recognition problem. However, *manually designed cost functions require expert knowledge in different domains. Thus, we propose to use a linear mapping to automatically learn the cost function, which relieves the need to configuration.*

More specifically, for predictions $\tilde{\boldsymbol{Z}}$ generated by Softmax operation via leveraging linear transformation matrix $\boldsymbol{W}$, $\boldsymbol{W} \in \mathbb{R}^{K \times K}$ is learned so that the following objective function is minimised.

$$\begin{aligned} &\underset{\boldsymbol{Y} \in \mathbb{R}}{\text{minimize}} \qquad -\langle \boldsymbol{Y}, \log \tilde{\boldsymbol{Z}} \rangle + \lambda \langle \boldsymbol{Y}, \log \boldsymbol{Y} \rangle, \\ &\boldsymbol{Y} \mathbf{1}_K = \mathbf{1}_N \quad \boldsymbol{Y}^\intercal \mathbf{1}_N = \boldsymbol{\mu}, \quad \tilde{\boldsymbol{Z}}_{nk} = \frac{\exp(\boldsymbol{W}^\intercal \Phi(\boldsymbol{x}_n))_k}{\sum_k^K \exp(\boldsymbol{W}^\intercal \Phi(\boldsymbol{x}_n))_k}. \end{aligned} \tag{12}$$

The resulting formula can be illustrated using a simple one-layer network of weight parameter $\boldsymbol{W}$, together with an error function in Eq. (12). We initialise $\boldsymbol{W}$ with an identity matrix and use small learning rate to learn $\boldsymbol{W}$. One could also absorb the Logit Adjustment term $-\log p(y^s)$ as the fixed bias parameter into the network. Motivated by this description, we can use a general gradient descent algorithm, such as SGD, to optimise the error function. Taking into account the implementation in practice, the direct calculation of Eq. (12) can be done using the Sinkhorn iterations (Cuturi, 2013; Frogner et al., 2015; Peyré et al., 2019) in mini-batch training efficiently. Besides, We term the proposed method as OTLM (***O**ptimal **T**ransport via **L**inear **M**apping*), Figure 1 illustrates the overview of OTLM.

Compute the gradients *w.r.t* $\boldsymbol{W}$ With the optimisation in Algorithm 1, it was conducted on the overall data set with a comparatively large number of data. Quite a different scenario now, as mini-batch training is more favoured when it comes to neural networks. For this reason, the optimisation workflow in Algorithm 1, as pointed by Peyré et al. (2019); Viehmann (2019), will have a problem of batch stablisation. To this end, we perform logarithmic transformation to step 3-4 in Algorithm 1.

$$\log \boldsymbol{u} = \log \boldsymbol{r} - \log(\boldsymbol{M}\boldsymbol{v}) = \log \boldsymbol{r} - \operatorname{logsumexp}\left(\frac{1}{\lambda}\boldsymbol{M} - \log \boldsymbol{v}\right) \tag{13}$$

$$\log \boldsymbol{v} = \log \boldsymbol{c} - \log(\boldsymbol{M}^\intercal \boldsymbol{u}) = \log \boldsymbol{c} - \operatorname{logsumexp}\left(\frac{1}{\lambda}\boldsymbol{M} - \log \boldsymbol{u}\right). \tag{14}$$

with log-sum-exp operation $\operatorname{logsumexp}(\boldsymbol{x}) = \log(\sum_i \exp \boldsymbol{x}_i)$. Assuming convergence is achieved in the Sinkhorn's loops, it is not necessary for us manually to compute the derivative of the loss function in Eq. (12) *w.r.t* $\boldsymbol{W}$. As a matter of fact, as long as we make sure that the entire optimisation process is differentiable, modern deep learning libraries can then automatically implement end-to-end derivations, as it allows us to differentiate between the results of Sinkhorn's loops as a mere composition of elementary operations.

Performance degradation caused by imbalanced distributions can be addressed with meta-learning based methods (Ren et al., 2020; Wang et al., 2020a; Li et al., 2021). Recent works have illustrated neural networks can learn more meaningful representations from the validation data set $\{(\boldsymbol{x}_n^v, y_n^v)\}_{n=1}^{N_v}$. Inspired by the methods based on meta learning, we also take a validation data set to optimise the parameter $\boldsymbol{W}$, *but with the still significant difference in that we require no labelling information, thus avoiding a large expanse of labeling samples*.

4 Experiments

In this section, we first conduct experiments comparing our approach versus extant post-hoc correction methods on three data sets, including CIFAR-100-LT (Cao et al., 2019), ImageNet-LT (Liu et al., 2019), and iNaturalist (Van Horn et al., 2018) with varying backbones. Finally, we empirically make a comparison of our algorithm with alternative cutting-edge long-tailed recognition methods. Observe that, for one thing, *a plausible coupling of our algorithm to any training-aware long-tailed recognition method exists*. As an illustration of the potency and strong generalisation of our approach, we conducted post-hoc correction of the pseudo-predictions of both methods, which are fairly typical for the training stage of all data sets: The first is based on the *cross entropy (CE)*, and the other is the recently proposed *RIDE* (Wang et al., 2020b), which is based on multi experts. *We conducted experiments in the appendix for semi-supervised learning, to some extent to imitate the online situation.*

4.1 Data Sets and Baselines

The baseline methods and data set are briefly described here, with implementation details placed in the appendix. *In our experiments, we assume that the marginal distribution of the test set is uniform due to the characteristics of the three data sets.*

Baselines: We compare our methods with (i) post-hoc correction methods including Logit Adjustment (Menon et al., 2020), DARP (Kim et al., 2020a), (ii) state-of-the-art methods including Focal Loss (Lin et al., 2017), LDAM (Cao et al., 2019), BBN (Zhou et al., 2020), Balanced Softmax (Ren et al., 2020), Causal Norm (Tang et al., 2020), LADE (Hong et al., 2020), M2m (Kim et al., 2020b), Decouple (Kang et al., 2019), LFME (Xiang et al., 2020), RIDE (Wang et al., 2020b)

Long-tailed data set: We take experiments on three data sets including CIFAR-100-LT, ImageNet-LT, and iNaturalist. We build the imbalanced version of CIFAR-100 by downsampling samples per class following the profile in Liu et al. (2019); Kang et al. (2019) with imbalanced ratios 10, 50, and 100. For all the benchmarks, we evaluate the performance on the test data set using a model trained on the training data set, and report the results using top-1 accuracy.

4.2 Main Results

Table 1 illustrates the results on the CIFAR-100-LT data set. For CE-based training-aware methods, it first shows that Logit Adjustment is indeed a simple yet effective method for post-hoc correction.

Table 1: Comparison on the top-1 accuracy with post-hoc correction methods on CIFAR-100-LT data set using ResNet-32 backbone, ImageNet-LT and iNaturalist data set using ResNeXt-50-32x4d and ResNet-50 backbones respectively. Better results are marked in bold, the small red font indicates the increase in accuracy when our method is compared to Logit Adjustment and RIDE. '-' means the results are not reported. CE indicates that a cross-entropy loss function is used in the training stage.

Data set	CIFAR-100			ImageNet-LT	iNaturalist
Imabalced Ratio	10	50	100	-	-
Baseline (CE)	59.00	45.50	41.00	43.10	65.00
Logit Adjustment+CE	<u>61.63</u>	<u>49.62</u>	<u>45.11</u>	<u>51.62</u>	<u>69.44</u>
DARP+CE	61.78	49.52	45.13	51.20	71.18
OT+CE	62.25$_{+0.62}$	50.05$_{+0.43}$	45.81$_{+0.70}$	51.87$_{+0.25}$	71.20$_{+1.76}$
OTLM+CE	62.18$_{+0.55}$	**50.35**$_{+0.73}$	**46.10**$_{+0.99}$	**52.35**$_{+0.73}$	**72.30**$_{+2.86}$
RIDE	-	-	<u>50.20</u>	<u>57.50</u>	<u>72.90</u>
OT+RIDE	-	-	50.82$_{+0.62}$	58.45$_{+0.95}$	75.85$_{+2.95}$
OTLM+RIDE	-	-	**51.43**$_{+1.23}$	**58.71**$_{+1.21}$	**76.12**$_{+3.22}$

Remarkably, it outperforms the baseline methods by 2.6%, 4.1%, and 4.1% under the imbalance ratio of 10, 50, and 100 respectively. With a quick look at the results based on the OT approach, the superior results stress the advantages of our approach over the DARP and Logit adjustments. OT algorithm can further improve the performance by 0.62%, 0.43%, 0.70%, for OTLM, it outperforms Logit Adjustment by 0.55%, 0.73%, 0.99% under the three imbalance ratios on CIFAR-100-LT data set. For RIDE-based training-aware methods, there is also a 0.6% and 1.23% improvement in the accuracy of our method.

Table 1 also provides the results on ImageNet-LT and iNaturalist data sets. The fact again attracts our attention that for CE-based training-aware methods, OT, which is based on convex optimisation, consistently outperforms Logit Adjustment by a large margin (1.76%) on iNaturalist data set. The accuracy boosting indicates the great potential of methods based on optimisation for post-hoc correction. Besides, not surprisingly, OTLM can further enhance the prediction accuracy, it outperforms Logit Adjustment by 0.73% and 2.86% on ImageNet-LT and iNaturalist respectively. As for RIDE, the gain in accuracy is also significant, about 3% on iNaturalist.

4.3 Comparison with the State-of-the-art Methods

Armed with a demonstration of the validity of OT and OTLM, we switched to a comparison of our performance with existing methods that achieved the most advanced results on three benchmarks. As shown in Table 2, in particular, on ImageNet-LT and iNaturalist data sets, the experimental results are remarkable and impressive, our method outperforms RIDE by 1.0% and 3.0%, respectively. *Since iNaturalist is a very difficult and fine-grained data set consisting of 8,142 categories. These huge performance gains come at a fraction of the cost*, as we show in the following subsection.

Table 2: Comparison on the top-1 accuracy with state-of-the-arts on CIFAR-100-LT data set using ResNet-32 backbone, on ImageNet-LT and iNaturalist data set using ResNeXt-50-32x4d and ResNet-50 backbones respectively. $-$ denotes the results are not reported, results underlined are the ones being compared, best results are marked in bold, the small red font denotes performance gain.

Data Set	CIFAR-100			ImageNet-LT	iNaturalist
Imabalced Ratio	10	50	100	–	–
Cross Entropy (CE)	55.7	45.5	38.3	44.4	61.7
Focal Loss	55.8	44.3	38.4	43.7	–
CB-Focal	–	–	–	–	61.1
OLTR	–	–	–	46.3	63.9
NCM	–	–	–	47.3	63.1
norm	–	–	–	49.4	69.3
cRT	–	–	–	49.6	67.6
LWS	–	–	–	49.9	69.5
LDAM	–	–	–	–	64.6
LDAM+DRW	58.7	46.6	42.0	–	68.0
BBN	59.1	47.0	42.6	–	69.3
Causal Norm	59.6	50.3	44.1	51.8	–
M2m	58.2	–	43.5	–	–
LFME	–	–	43.8	–	–
Balanced Softmax	61.6	49.9	45.1	–	–
LADE	61.7	50.5	45.4	51.9	70.0
RIDE (6 experts)	–	–	50.2	57.5	72.9
Logit Adjustment	61.6	49.6	45.1	51.6	69.4
DARP	61.8	49.5	45.1	51.2	71.2
OT+CE	**62.3**$_{+0.7}$	50.1	45.8	51.9	71.2
OTLM+CE	62.2	**50.4**$_{+0.8}$	46.1	52.4	72.3
OT+RIDE (6 experts)	–	–	**50.8**$_{+0.6}$	**58.5**$_{+1.0}$	**75.9**$_{+3.0}$
OTLM+RIDE (6 experts)	–	–	**51.4**$_{+1.2}$	**58.7**$_{+1.2}$	**76.1**$_{+3.2}$

4.4 Computation Cost of OT and OTLM

As we have highlighted, the additional computational cost of OT and OTLM is particularly small, compared to that of training-aware methods. In Table 3, exact time of the evaluations on the ImageNet-

Table 3: Time for different methods to execute on the ImageNet-LT and iNaturalist data sets. The times here are measured from the start of each method to the final best performance. Their running time are counted in seconds. Coupled with the prior results, an observation can be made that OT can be the best coordinate to trade off performance and efficiency in post-hoc correction methods.

Data Set	ImageNet-LT	iNaturalist
Logit Adjustment	0.3	1.2
OT	32.7	82.5
OTLM	134.4	256.5

LT and iNaturalist data sets is provided. Please note, the time here are measured from the start of each method to the final best performance. For the comparison between OT and OTLM, because OTLM runs on the GPU, one optimisation iteration of OTLM is in fact much faster than OT. Except for OTLM, which was run on an NVidia card (V100), the results come from a 28-core machine (2.20 Ghz Xeon). *Firstly, we can observe that all post-hoc correction methods here are not quite time-consuming*. For comparison, ImageNet-LT and iNaturalist can be trained on 4 NVidia cards (V100) for extremely long periods of time. Using ResNeXt-50-32x4d on the iNaturalist training data set, for example, with a batch size of 256, the amount of training time (in seconds) to perform 1 iteration is approximately 850. *Also, if one differs the performance of each method, we gather that OT and OTLM provide the best balance of performance and running costs to match*.

4.5 Convergence Behavior of OTLM

To verify that our adopted OTLM algorithm is really efficient. With ResNeXt-50-32x4d as the backbone, we plotted curves depicting the decreasing absolute error σ with the number of iterations t on the ImageNet-LT validation data set. Image on the left in Figure 2 gives us a visualisation of the optimisation process. It is found that the absolute error σ converges at around 60 iterations. In fact, we discovered that 10 or even fewer iterations were enough to yield satisfactory performance.

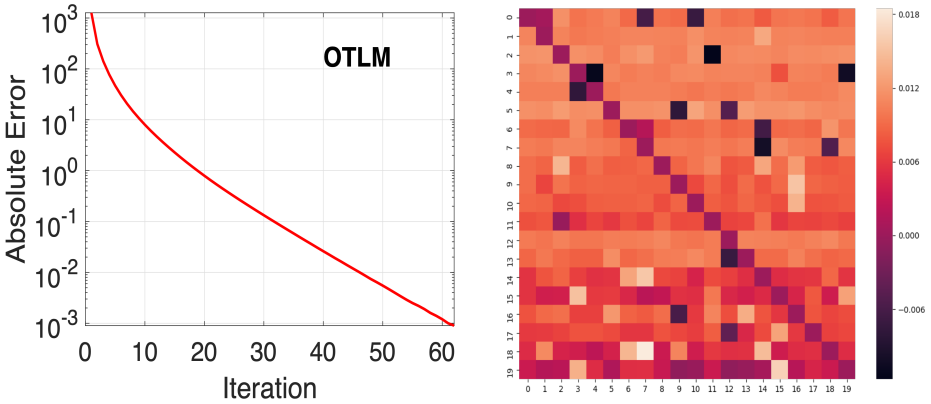

Figure 2: The left image shows the convergence of OTLM: The absolute error σ decreases with iterations on the ImageNet-LT validation data set (20k samples), and we used ResNet-50 as the backbone to obtain the original predictions $\boldsymbol{Z}$. The right image demonstrates the block in $\boldsymbol{W}$, we trained on CIFAR-100 to produce the original predictions with OTLM to minimise the difference between the ideal and actual distributions. For ease of display, the values of the diagonal elements are relatively large and we will replace these values with zeros.

4.6 Detailed Analysis on Distribution Matching

Upon distribution fitting, preferably the estimation bias is lessened, which implies that the KL divergence distance between the refined prediction and the target distribution is decreased. Table 4 displays the KL distances on the ImageNet-LT and iNaturalist data sets, and we can find that OT possesses a smaller KL distance than Logit Adjustment, which suggests that OT decreases the estimation bias more than Logit Adjustment by a much larger margin.

Table 4: KL distance between refined probability distribution and target distribution computed on ImageNet-LT and iNaturalist data sets. Smaller distance indicates two distributions are more similar.

Data Set	ImageNet-LT	iNaturalist
Logit Adjustment	1.78e-05	5.81e-06
DARP	8.49e-08	1.07e-08
OT	8.62e-10	4.72e-10

4.7 Visualisation on Weight Parameter in OTLM

As a natural consequence, one wonders about the exact structure of this weight matrix $\boldsymbol{W}$, and no better way to present its results than through visualisation. Towards this end, we applied OTLM to refine the original predictions on CIFAR-100 data set with an imbalanced ratio $R = 100$ and ResNet-32 backbone, and trained a one-layer model until it converged to extract the value of $\boldsymbol{W}$, where the shape of is 100×100. We visualised a sub-block of size 20×20, the illustration is presented on the right image in Figure 2. The visualised matrix is very dense, this non-sparsity suggests that the OTLM manages to correct the predictions of the model in the training stage, whilst minimising the distance between the desired and the actual distribution.

5 Related Work

Long-tailed recognition: The great majority of current methods applied to long-tailed recognition can be divided into either training-aware and post-hoc correction methods. The training-aware methods aim to improve the generalisation ability of models trained on imbalanced training data set. Among them, re-sampling and re-weighting are two major methods. Re-sampling controls the class numbers by the means of over-sampling (Chawla et al., 2002) and under-sampling (Barandela et al., 2004; Drummond et al., 2003). The re-weighting approaches achieve balanced learning by giving smaller weights to majority classes and larger weights to minority classes (Tang et al., 2008; Zadrozny et al., 2003; Lin et al., 2017; Lee et al., 2017; Cui et al., 2019; Khan et al., 2019). Other approaches have also been proposed to resolve the imbalanced problem. For example, Cao *et al.* introduces a theoretically-principled loss function motivated by minimizing the generalisation error bound (Cao et al., 2019), Kang *et al.* proposes to decouple the learning phases into representation learning and classifier tuning (Kang et al., 2019). Apart from training-aware methods, post-hoc correction also offers us an avenue for long-tailed recognition problem, it can be applied to the normalisation of classifier weights (Kang et al., 2019; Zhang et al., 2019; Kim & Kim, 2020) or logits based on label frequency (Provost, 2000; Collell et al., 2016). Among them, Logit Adjustment (Menon et al., 2020) is a simple yet powerful method to use. Compared with Logit Adjustment, our approach performs the post-hoc correction from the optimisation view.

Optimal transport: OT distance is a important family of distances for probability measures (Villani, 2008; Santambrogio, 2015). It has been extensively applied in different fields in machine learning. Courty *et al.* (Courty et al., 2016) proposes a regularised unsupervised optimal transportation model to perform the domain adaption. Other applications include 3D shape matching (Su et al., 2015), generative model (Arjovsky et al., 2017; Salimans et al., 2018; Bunne et al., 2019; Deshpande et al., 2019; Bhagoji et al., 2019), graph matching (Xu et al., 2019a;b), model designs Kandasamy et al. (2018); Chizat & Bach (2018). To the best of our knowledge, we are the first to perform the post-hoc correction from the OT perspective.

6 Conclusion

In this paper, we present two techniques for post-hoc corrections from an optimisation perspective. Our approach reviews the basic mathematical formulation of the distribution alignment problem and relates the resulting formulation to OT, which we find, moreover, offers greater possibilities and flexibility to solve the problem. In order to avoid manual configuration of the cost matrix, we propose to learn the cost matrix automatically by linear mapping, which leads to further performance improvements. The experimental results confirm the superiority of our proposed approach. We believe our work can open up more possibilities and imagination for post-hoc corrections. There are potential challenges that remain to be addressed. These include, for example, how to apply OT to correction of predictions under unknown marginal distribution. Meanwhile, an extension of our approach to other applications where label shift is involved is considered possible.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. *arXiv preprint arXiv:1903.09734*, 2019.

Ricardo Barandela, Rosa M Valdovinos, J Salvador Sánchez, and Francesc J Ferri. The imbalanced training sample problem: Under or over sampling? In *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)*, pp. 806–814. Springer, 2004.

Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. Lower bounds on adversarial robustness from optimal transport. *arXiv preprint arXiv:1909.12272*, 2019.

Paula Branco, Luís Torgo, and Rita P Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):1–50, 2016.

Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

Charlotte Bunne, David Alvarez-Melis, Andreas Krause, and Stefanie Jegelka. Learning generative models across incomparable spaces. In *International Conference on Machine Learning*, pp. 851–861. PMLR, 2019.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *arXiv preprint arXiv:1906.07413*, 2019.

Claire Cardie and Nicholas Howe. Improving minority class prediction using case-specific feature weights. 1997.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *arXiv preprint arXiv:1805.09545*, 2018.

Guillem Collell, Drazen Prelec, and Kaustubh Patil. Reviving threshold-moving: a simple plug-in bagging ensemble for binary and multiclass imbalanced data. *arXiv preprint arXiv:1606.08698*, 2016.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9268–9277, 2019.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David Forsyth, and Alexander G Schwing. Max-sliced wasserstein distance and its use for gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10648–10656, 2019.

Chris Drummond, Robert C Holte, et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11, pp. 1–8. Citeseer, 2003.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. Learning with a wasserstein loss. *arXiv preprint arXiv:1506.05439*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. *arXiv preprint arXiv:1802.05300*, 2018.

Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. *arXiv preprint arXiv:2012.00321*, 2020.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5375–5384, 2016.

Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric Xing. Neural architecture search with bayesian optimisation and optimal transport. *arXiv preprint arXiv:1802.07191*, 2018.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

Salman Khan, Munawar Hayat, Syed Waqas Zamir, Jianbing Shen, and Ling Shao. Striking the right balance with uncertainty. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 103–112, 2019.

Byungju Kim and Junmo Kim. Adjusting decision boundary for class imbalanced learning. *IEEE Access*, 8:81674–81685, 2020.

Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. *arXiv preprint arXiv:2007.08844*, 2020a.

Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13896–13905, 2020b.

Wonji Lee, Chi-Hyuck Jun, and Jong-Seok Lee. Instance categorization by support vector machines to adjust weights in adaboost for imbalanced data classification. *Information Sciences*, 381:92–103, 2017.

Shuang Li, Kaixiong Gong, Chi Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng. Metasaug: Meta semantic augmentation for long-tailed visual recognition. *arXiv preprint arXiv:2103.12579*, 2021.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546, 2019.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Foster Provost. Machine learning from imbalanced data sets 101. In *Proceedings of the AAAI'2000 workshop on imbalanced data sets*, volume 68, pp. 1–3. AAAI Press, 2000.

Xingye Qiao and Yufeng Liu. Adaptive weighted learning for unbalanced multicategory classification. *Biometrics*, 65(1):159–168, 2009.

Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. *arXiv preprint arXiv:2007.10740*, 2020.

Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving gans using optimal transport. *arXiv preprint arXiv:1803.05573*, 2018.

Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. ii. *Proceedings of the American Mathematical Society*, 45(2):195–198, 1974.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

Zhengyu Su, Yalin Wang, Rui Shi, Wei Zeng, Jian Sun, Feng Luo, and Xianfeng Gu. Optimal mass transport for shape matching and comparison. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2246–2259, 2015.

Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *arXiv preprint arXiv:2009.12991*, 2020.

Yuchun Tang, Yan-Qing Zhang, Nitesh V Chawla, and Sven Krasser. Svms modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):281–288, 2008.

Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

Thomas Viehmann. Implementation of batched sinkhorn iterations for entropy-regularized wasserstein loss. *arXiv preprint arXiv:1907.01729*, 2019.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Renzhen Wang, Kaiqin Hu, Yanwen Zhu, Jun Shu, Qian Zhao, and Deyu Meng. Meta feature modulator for long-tailed recognition. *arXiv preprint arXiv:2008.03428*, 2020a.

Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X Yu. Long-tailed recognition by routing diverse distribution-aware experts. *arXiv preprint arXiv:2010.01809*, 2020b.

Liuyu Xiang, Guiguang Ding, and Jungong Han. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *European Conference on Computer Vision*, pp. 247–263. Springer, 2020.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Hongteng Xu, Dixin Luo, and Lawrence Carin. Scalable gromov-wasserstein learning for graph partitioning and matching. *arXiv preprint arXiv:1905.07645*, 2019a.

Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin Duke. Gromov-wasserstein learning for graph matching and node embedding. In *International conference on machine learning*, pp. 6932–6941. PMLR, 2019b.

Bianca Zadrozny, John Langford, and Naoki Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Third IEEE international conference on data mining*, pp. 435–442. IEEE, 2003.

Junjie Zhang, Lingqiao Liu, Peng Wang, and Chunhua Shen. To balance or not to balance: A simple-yet-effective approach for learning with long-tailed distributions. *arXiv preprint arXiv:1912.04486*, 2019.

Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9719–9728, 2020.

A Optimal Transport on Long-tailed Semi-supervised Classification

Together, with the extension of our approach, we adapt it to semi-supervised long-tailed classification tasks. We bring our approach together with the advanced semi-supervised learning methods FixMatch Sohn et al. (2020). Adhering to the configuration in DARP Kim et al. (2020a), we also create synthetically long-tailed variants of CIFAR-100 with imbalanced ratio $R = 20$. The sample size of the unlabeled data is twice as large as the labelled data, with the same proportion of imbalance between them. To assess the performance, we report two popular metrics: balanced accuracy (bACC) Huang et al. (2016) and geometric mean scores (GM) Branco et al. (2016). Experimentally, with the exception of an OT to align the marginal distribution of the unlabelled data, all settings are consistent with DARP, and we set λ to 0.005. The results are reported in Table 5. It can be seen that our approach also achieves much better performance on semi-supervised learning tasks. *It can somehow mimic the online scenarios. And the advantages of OT coping with such cases can be demonstrated.*

Table 5: Comparison of classification performance (bACC/GM) on CIFAR-100 with imbalanced ratio $R = 20$, We report standard deviation and mean values for each evaluation metric. Better results are marked in bold.

Method	CIFAR-100	
	bACC	GM
DARP	$54.9_{\pm 0.05}$	$46.4_{\pm 0.41}$
OT	$\mathbf{56.6}_{\pm 0.08}$	$\mathbf{48.4}_{\pm 0.32}$

B Parameter Study of λ

We conduct an additional experiment on iNaturalist data set to study the effect of λ on the final performance. The value of λ is selected from $\{0.01, 05, 0.1, 1.0.2.0, 5.0\}$, all other experimental settings are the same, including various hyperparameters, network structure, etc. As illustrated in Figure 3, we can find that OT achieves the best performance when $\lambda = 1$.

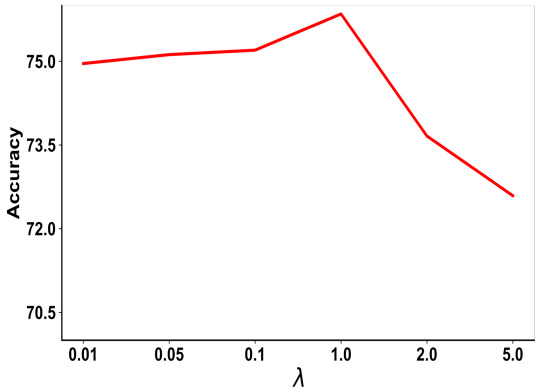

Figure 3: Visualisation of the accuracy change curve with λ on iNaturalist data set.

C Implementation Details

The specific implementation details for each data set under the different methods are described below.

C.1 Implementation Details of Cross Entropy

CIFAR-100-LT: During the training stage, following the recipe of (Tang et al., 2020; Hong et al., 2020), we apply SGD with batch size 256 and weight decay 0.0005 to train a ResNet-32 (He et al.,

2016) model for 200 epochs, we employ the linear warm-up learning rate schedule for the first five epochs. We also set the base learning rate to 0.2 and reduce it at epoch 120 and 160 by a factor of 100. During the post-hoc stage, we use Adam with batch size 10k and weight decay 0.001 to train the one-layer feed-forward network for 20 iterations. We set λ to 0.1, T to 200, Δ to 0.001. The learning rate is constant and set to 0.001.

ImageNet-LT: During the training stage, following the implementation of (Hong et al., 2020), we train the ResNeXt-50-32x4d (Xie et al., 2017) for 90 epochs and perform a cosine learning rate scheme with an initial learning rate of 0.05. SGD is also employed to optimise the neural network with batch size 256, weight decay 0.0005, and momentum 0.9. During the post-hoc stage, we use SGD with batch size 10k, weight decay 0.001 to train the one-layer feed-forward network for 100 iterations. The constant learning rate is 0.2. We set λ to 1.2, T to 7, Δ to 0.001.

iNaturalist: During the training stage, ResNet-50 (He et al., 2016) is chosen as the backbone network. We use SGD with momentum 0.9, batch size 256 to train the network. We utilise the cosine learning rate schedule gradually decaying from 0.1 to 0. During the post-hoc stage, we use SGD with batch size 23,826, weight decay 0.001, λ to 1.0, Δ to 0.001 and T to 10. The constant learning rate is set to 0.05. As iNaturalist has no test data set, we directly report the performance on validation data set.

C.2 Implementation Details of RIDE

The training details we use in RIDE share the same as the original paper, there are 6 experts in our RIDE on all three data sets.

CIFAR-100-LT We experimented with $R = 100$ and ResNet-32 as the base network. SGD is utilised as our solver with momentum 0.9, batch size 128, and epoch 200. The learning rate is initialised as 0.1 and decayed by 0.01 at epoch 120 and 160 respectively.

ImageNet-LT We take experiments with ResNeXt-50 as backbone with 100 epochs. We set the initial learning rate to 0.1 and reduce it by 10 at epoch 60 and 80. SGD with batch size 256 and momentum 0.9 is also adopted as the optimisation solver here.

iNaturalist We conduct experiments with ResNet-50 as backbone. The training details are the same as ImageNet-LT except from batch size 512.