Feature Directions Matter: Long-Tailed Learning via Rotated Balanced Representation

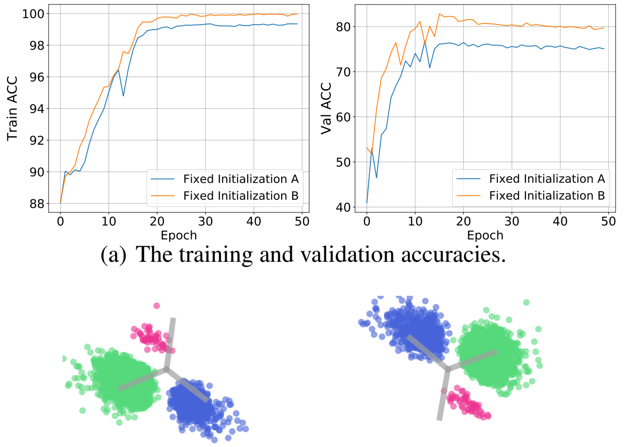
Gao Peifeng [1] Qianqian Xu [2] Peisong Wen [1 2] Zhiyong Yang [3 4] Huiyang Shao [1 2] Qingming Huang [1 2 5 6]

Abstract

There are some studies aiming to solve long-tailed classification from the perspective of feature learning. Recent work proposes to learn the balanced representation by fixing the linear classifier as *Equiangular Tight Frame* (ETF), since they argue what matters in classification is the structure of the feature, instead of their directions. Holding a different view, in this paper, we show that features with fixed directions may be harmful to the generalization of models, even if it is completely symmetric. To avoid this issue, we propose *Representation-Balanced Learning* Framework (RBL), which introduces orthogonal matrices to learn directions while maintaining the geometric structure of ETF. Theoretically, our contributions are two-fold: **1**). we point out that the feature learning of RBL is insensitive toward training set label distribution, it always learns a balanced representation space. **2**). we provide a generalization analysis of proposed RBL through training stability. To analyze the stability of the parameter with orthogonal constraint, we propose a novel training stability analysis paradigm, *Two-Parameter Model Stability*. Finally, our method is extremely simple in implementation but shows great superiority on several benchmark datasets.

1. Introduction

Generally, real-world visual classification tasks suffer from long-tailed distribution data, where a few categories (head class) contribute to major observations of datasets, while other classes (tail class) only contain a few samples. For example, iNaturalist2018 (Van Horn et al., 2018) is a large-scale dataset, which contains more than 8K categories. In this dataset, the head class has several thousand images, whereas the tail class may only have no more than one hundred images. In this setting, training a well-performed model is very hard, because the model will be overwhelmed by head classes and underfit the tail classes.



(a) The training and validation accuracies.

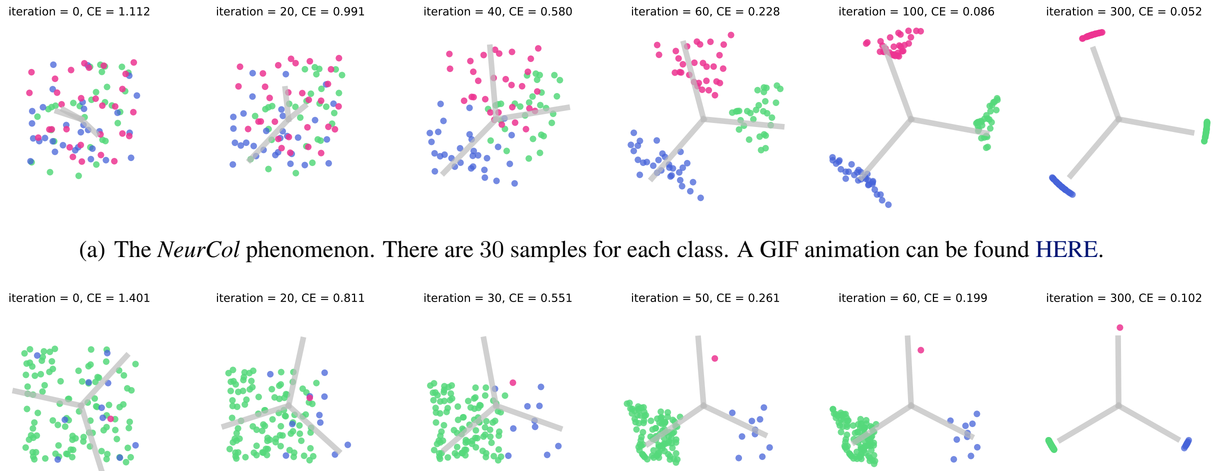
(b) Features visualization on training set at epoch 30. Left: initialization A. Right: initialization B.

Figure 1: Two toy experiments to illustrate feature learning and generalization of *Fixed*. A two-layer deep model and a linear classifier are trained to solve a long-tailed classification problem with 2-dimensional feature and 3 classes. In (b), points and lines indicate sample feature and model weight respectively.

Most previous methods solve the long-tailed problem through data-resampling based and loss-reweighing based methods, which improve the performance of tail classes at the expense of sacrificing head class performance (Kang et al., 2019). Recently, *Neural Collapse* (*NeurCol*) (Papyan et al., 2020) phenomenon has raised increasing attention in deep learning community. It can provide a different perspective for long-tailed learning. *NeurCol* phenomenon happens on classifiers trained over a label-balanced dataset: after the cross entropy loss reaches its minimum, features of the classifier (last layer activations of the deep model) would

[1] School of Computer Science and Technology, UCAS, Beijing, China. [2] Key Laboratory of Intelligent Information Processing, Inst. of Comput. Tech., CAS, Beijing, China. [3] State Key Laboratory of Info. Security (SKLOIS), Inst. of Info. Engin., CAS, Beijing, China. [4] School of Cyber Security, UCAS, Beijing, China. [5] BDKM, UCAS, Beijing, China. [6] Peng Cheng Laboratory, Shenzhen, China. Correspondence to: Qianqian Xu <xuqianqian@ict.ac.cn>, Qingming Huang <qmhuang@ucas.ac.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.



(a) The *NeurCol* phenomenon. There are 30 samples for each class. A GIF animation can be found HERE.

(b) The feature learning of our framework. Each class has $100, 10, 1$ samples respectively. A GIF animation can be found HERE.

Figure 2: Two numerical simulation experiments to illustrate the feature learning of classification. In experiments, a linear classifier was trained to solve a classification problem with 2-dimensional feature and 3 categories. To simulate the model with infinite fitting ability, we directly update the features in $\mathbb{R}^2$. The pictures from left to right record the location of the model weight and sample feature in $\mathbb{R}^2$ during the optimization process. In each picture, points and lines indicate sample feature and model weight respectively. Implementation details of experiments in this figure could be found in Appendices.

learn a completely symmetric structures. Specifically, sample features within the same class and the corresponding weight vector of linear classifiers would collapse to its class center, and centers of every class would form the structure of *Equiangular Tight Frame* (ETF). As the last picture of Fig.2 (a) illustrated, after the training converges, every class is highly symmetric for others in the feature space.

Due to ETF's elegant property, an existing work *Fixed* (Yang et al., 2022b) proposes to learn from ETF directly. They argue that the deep model can learn features with any direction, hence learning directly fixed ETF can achieve satisfactory performances. We take a different view on this. Fig.1 shows two results of *Fixed* with different initialization of ETF. We see the first generalizes better than the second, even though both of them lead to *NeurCol* and learn ETF features. Based on this observation, we argue the bad initialization of ETF is harmful to the generalization of the deep model. Since it only requires the model *push* samples in the randomly generated direction, rather than making it *learn* a direction. To overcome this problem, we propose a *Rrepresentation-Balanced Learning* Framework (RBL). The feature learning of our framework can be divided into two steps: **1)** generate the balanced features space and **2)** use a learnable orthogonal matrix to register the sample features and balanced features.

First Step Before training, we generate the balanced feature space. Directly generating ETF is the best option. Unfortunately, ETF exists only for sparse combinations of a number of class C and feature dimension d (Sustik et al., 2007). We argue that the *equiangular* property is the key point to solving long-tailed problems, so we turn to the second best. We generate the $max(C, d)$-dimensional trivial ETF (trivial ETF always exists), then construct an equiangular structure from it.

Second Step Same as *Fixed*, we fix the linear classifier to be the balanced feature that we generate in advance so that the deep model could learn directly from the ETF. To avoid fixed features damaging the learning, we introduce rotation operation. As shown in Fig.2 (b), during training, we keep balanced feature being a rigid body and only perform rotation on it.

In theory, we analyze our method from the perspectives of feature learning and generalization. We prove our framework could achieve balanced feature space, *Regular Simplex*, regardless of whether the dataset is balanced or not. Meanwhile, we use training stability to analyze generalization performances of our method. The stability analysis of our framework is different from previous analysis since our framework contains two parts of parameters, one is parameterized orthogonal matrices and the another is the weight of deep model. During training, they follow different rules to be updated. To this end, we propose a novel training stability analysis paradigm, *Two-Parameter Model Stability*, which divides the model parameter into two parts to derive model stabilities (Lei & Ying, 2020).

To sum up, our contributions are as follows:

- We propose *Representation-Balanced Learning* Framework, which can lead to *NeurCol* phenomenon even in long-tailed scenarios.
- We propose *Two-Parameter Model Stability*, and present a generalization analysis for our framework.
- A series of empirical studies demonstrate the effectiveness of our method.

2. Preliminary

In this section, we introduce *Neural Collapse* phenomenon and give the definition of Simplex *Equiangular Tight Frame* (Simplex ETF). Papyan et al. (2020) has found *NeurCol* phenomenon in the training for deep classifier. Here, for a classification problem with C classes, a classifier with d-dimensional feature is defined as $\hat{y} = \arg\max_i [Mf(\boldsymbol{x})]_i$, where $M = [M_1, \dots, M_C]^T \in \mathbb{R}^{C\times d}$ is the linear classifier, $\boldsymbol{x}$ is a data point and $f(\cdot) \in \mathbb{R}^d$ is the deep feature extractor. Given a balanced dataset, we denote the i-th sample in y-th category as $\boldsymbol{x}_{y,i}$. They found as the cross entropy loss (over a label balanced dataset) converging to the minimum, the last-layer activations of model, $i.e.$, $f(\boldsymbol{x})$ show amazing simplicity geometrically:

NC1 **Variability Collapse** All samples belonging to the same class converge to the class mean: $\|f(\boldsymbol{x}_{y,i}) - \overline{f(\boldsymbol{x}_y)}\| \to 0, \forall y, \forall i$ where $\overline{f(\boldsymbol{x}_y)} = \text{Ave}_i\,(f(\boldsymbol{x}_{y,i}))$ denotes the class-center of y-th class;

NC2 **Convergence to Self Duality** The samples and classifier belonging to the same class converge to the same: $\|f(\boldsymbol{x}_{y,i}) - M_y\| \to 0, \forall y, \forall i$;

NC3 **Convergence to Simplex ETF** The classifier weight converges to the vertices of Simplex Equiangular Tight Frame (ETF);

NC4 **Nearest Classification** The learned classifier behaves like the nearest classifier, $i.e.$, given any sample $\boldsymbol{x}$ in dataset, $\arg\max_y \langle M_y, f(\boldsymbol{x})\rangle \to \arg\min_y \|f(\boldsymbol{x}) - \overline{f(\boldsymbol{x}_y)}\|$.

NC1-2 and **NC4** tells us the classifier would cluster samples from the same category together in the feature space. And **NC3** states that feature of different classes would collapse to Simplex ETF. Here is the definition of Simplex ETF.

Definition 2.1 (**Simplex Equiangular Tight Frame** (Papyan et al., 2020))**.** A Simplex ETF is a collection of points in $\mathbb{R}^C$ specified by the columns of

$$M^\star = \alpha R \sqrt{\frac{C}{C-1}} \left(I - \frac{1}{C} \mathbb{I}\mathbb{I}^T \right)$$

where $I \in \mathbb{R}^{C\times C}$ is the identity matrix, $\mathbb{I} \in \mathbb{R}^C$ is the all-one vector, $R \in \mathbb{R}^{d\times C} (d \geq C)$ is an orthogonal projection matrix, $\alpha \in \mathbb{R}$ is a scale factor.

3. Methodology

In this section, we propose *Representation-Balanced Learning* Framework, which aims to learn a balanced representation space in long-tailed scenarios. We provide its code implementation in Appendices.E.

3.1. Generation of Balanced Feature

Different from *Fixed* (Yang et al., 2022b), we do not use Simplex ETF as the balanced feature, because Simplex ETF only exists when feature dimension d is larger than class number C. In fact, we argue a sets of feature that meets *equiangular* property is enough for feature learning. Therefore, the requirement of Simplex ETF for any C and d is unnecessary. We define that the Simplex ETF with the same space dimension and vectors number as trivial ETF:

Definition 3.1 (**Trivial Equiangular Tight Frame**)**.** A trivial ETF is a collection of points in $\mathbb{R}^C$

$$M^\star = \sqrt{\frac{C}{C-1}} \left(I - \frac{1}{C} \mathbb{I}\mathbb{I}^T \right)$$

where I is the identity matrix and $\mathbb{I}$ is the ones vector.

Obviously, a C-dimensional trivial ETF can be seen as a Simplex ETF that has C vectors in $\mathbb{R}^C$. In classification problems, if the number of classes C is smaller than feature dimension d, we select C vectors in d-dimensional trivial ETF as the balanced feature. It is reasonable because the subset of ETF still meets *equiangular* condition. In another case that $C > d$, we directly generate trivial ETF in $\mathbb{R}^C$. When performing feature learning, we use linear transformation to transform the feature dimension d into C. In this way, we obtain features that satisfy *equiangular* property for any number of classes and feature dimension.

3.2. Representation-Balanced Learning

Notations We define symbols first. Denote the sample space as $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the data space and $\mathcal{Y}$ is the label space. We assume $\mathcal{Y} = \{1, \dots, C\}$, where C is the number of classes. Let the long-tailed training set be $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathcal{X}$ is the data point and $y_i \in \mathcal{Y}$ is the label. Denote $f(\cdot; \boldsymbol{w})$ the deep model, where $\boldsymbol{w}$ is the model parameter. Assume the feature $f(\boldsymbol{x}; \boldsymbol{w})$ is a d-dimensional vector.

Framework Remember in *Neural Collapse* phenomenon, the class feature will coincide with the classifier weight of the corresponding category. To make $f(\cdot; \boldsymbol{w})$ learn balanced feature space M^*, the plain idea is fixing the linear classifier as M^* when performing training. However, this approach may harm the feature learning of the model, since $f(\boldsymbol{x};\boldsymbol{w})$ is expected to learn a completely fixed matrix. To avoid this issue, we introduce a learnable orthogonal matrix R (refer to next subsection for optimization in $SO(n)$) to register M^* and $f(\boldsymbol{x};\boldsymbol{w})$.

$$\begin{aligned} \min_{\boldsymbol{w},R} \quad & \mathcal{L}(\boldsymbol{w}, R, S) := -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp([\text{logit}(\boldsymbol{x})]_{y_i})}{\sum_{y=1}^{C}\exp([\text{logit}(\boldsymbol{x})]_{y})} \\ s.t. \quad & \text{logit}(\boldsymbol{x}_i) = M^\star R f(\boldsymbol{x}_i;\boldsymbol{w}) \end{aligned} \tag{1}$$

where $M^\star$ is the balanced feature we generate in advance and R is the orthogonal matrix. According to the number of categories C and feature dimension d, the sizes of $M^\star$ and R are designed specifically. If $C \leq d$, $M^\star$ is a $C \times d$ matrix and R is $d \times d$ orthogonal matrix, where every row of $M^\star$ is the vector in d-dimensional trivial ETF. When $C > d$, we generate C-dimensional trivial ETF as $M^\star$ and let R be the $C \times d$ orthogonal projection matrix. The R is for preservation of *equiangular* of M^*. In this way, $f(\boldsymbol{x};\boldsymbol{w})$ is no longer only learning from M^*, but learning from M^*'s direction.

Post-Hoc Logit Adjustment In the analysis of the next section, we would prove that optimization framework (1) could lead to *NeurCol* even in the long-tailed scenarios. However, feature learning of our framework can only learn balanced features, and it still requires proper decision-making for full play to its abilities (Kang et al., 2019). To this end, we perform Logit Adjustment when the model performs classification.

$$\arg\max_{i\in\mathcal{Y}} \left[M^\star R f(\boldsymbol{x};\boldsymbol{w}) - margin\right]_i \tag{2}$$

Logit Adjustment (Menon et al., 2020) is a simple but effective method for long-tailed learning. It adds *margins* before Softmax to make the loss function Fisher Consistent, which means the model trained by Logit Adjustment over a long-tailed dataset could minimize balanced error consistently. Since *margins* in loss function will influence the feature learning, we use post-hoc Logit Adjustment, which is subtracting *margins* when performing classification. Here, we follow the configuration of Balanced Softmax (Ren et al., 2020), set *margin* as $[\log(N_1/N), \dots, \log(N_C/N)]^T$, where N_i is the number of samples in category i of training set.

3.3. Optimization in Lie Group

So far, we have proposed *Representation-Balanced Learning* Framework. In our framework, we use rotation parameterized by an orthogonal matrix to preserve the *equiangular* property of $M^\star$. In the specific implementation, we can use a block of orthogonal matrices to represent orthogonal projection matrix. Therefore, the question becomes how to optimize an orthogonal matrix. Suppose we need a matrix that lies in $SO(d)$ to represent rotation. $SO(d)$ is the special orthogonal group, $i.e.$, Lie Group

$$SO(d) = \{A \in \mathbb{R}^{d\times d} | A^T A = I, \det A = 1\}$$

Note that the standard SGD can not assure that R always be in $SO(d)$ during training. We address this issue in an algebra way (Lezcano-Casado & Martınez-Rubio, 2019). Consider the Lie Algebra $\mathfrak{so}(d)$ formed by skew-matrices

$$\mathfrak{so}(d) = \{A \in \mathbb{R}^{d\times d} | A + A^T = 0\}$$

In the theory of Lie Group, there exists a well-known conclusion between the structure of the $SO(d)$ and $\mathfrak{so}(d)$, $i.e.$, exponential mapping on matrix $\exp\{\cdot\} : \mathfrak{so}(d) \to SO(d)$ is a homomorphism of Lie Group $SO(d)$. The mapping exponential of matrix $\exp(\cdot)$ is defined as

$$\exp(A) = I + A + \frac{A^2}{2} + \dots$$

Therefore, the optimization in $SO(d)$ could be transformed into optimization in $\mathfrak{so}(d)$:

$$\min_{A\in SO(d)} \text{loss}(A) \overset{A=\exp\{B\}}{\Longleftrightarrow} \min_{B\in\mathfrak{so}(d)} \text{loss}(\exp\{B\}) \tag{3}$$

Note that both sides of (3) have the same minimum. Since $\exp(\cdot)$ is a surjective mapping, once one obtains the solution of right, the other side could be found by the mapping $A = \exp(B)$. Furthermore, the Lie Algebra $\mathfrak{so}(d)$ is isomorphic to a linear space. The isomorphism mapping is given by $\phi(A) : A \mapsto A - A^T$. Consequently, the constraint of $SO(d)$ could be eliminated.

$$\min_{A\in SO(d)} \text{loss}(A) \overset{A=\exp(B-B^T)}{\Longleftrightarrow} \min_{B\in\mathbb{R}^{d\times d}} \text{loss}(\exp(B - B^T)) \tag{4}$$

In the above formulation, the optimization with orthogonal constraint is transformed into the optimization in $\mathbb{R}^{d\times d}$. For the right side of (4), we could use standard optimization techniques such as SGD and Adam.

4. Theoretical Analysis

In this section, we analyze our framework from feature learning and generalization. All proofs in this section could be found in Appendices.

4.1. Feature Learning

Before studying the feature learning of RBL, we have to mention the work of (Graf et al., 2021). Different from the experimental observations (*NeurCol* and ETF feature) of (Papyan et al., 2020), the optimal feature obtained by them through inequalities is also symmetrical, yet it is not ETF, but *Regular Simplex*. The relationship between ETF and *Regular Simplex* in algebra is very complex (Fickus et al., 2018). However, to the best of our knowledge, no works study their relationship in feature learning. The following lemma can fill this void:

Lemma 4.1 (Regular Simplex). *Consider a c-equiangular ETF $\{\zeta_i\}_{i=1}^{C}$, where $\langle \zeta_i, \zeta_j \rangle = c$ for any $i, j (i \neq j)$. Define its means as $\bar{\zeta} = \frac{1}{C}\sum_{i=1}^{C}\zeta_i$. Then the frame $\{\zeta_i - \bar{\zeta}\}_{i=1}^{C}$ forms a Regular Simplex, which means*

$$(\boldsymbol{S1}). \sum_{i=1}^{C}(\zeta_i - \bar{\zeta}) = \mathbf{0} \qquad \textit{(zero mean)}$$

$$(\boldsymbol{S2}). \|\zeta_i - \bar{\zeta}\| = \sqrt{\mathcal{N}(C,c)}, \forall i \qquad \textit{(equalnorm)}$$

$$(\boldsymbol{S3}). \langle \zeta_i - \bar{\zeta}, \zeta_j - \bar{\zeta} \rangle = \mathcal{A}(C,c), \forall i \neq j \qquad \textit{(equiangular)}$$

where

$$\mathcal{A}(C,c) = c - \frac{1 + (C - c)}{C}$$

and

$$\mathcal{N}(C,c) = 1 - \frac{1 + (C - c)}{C}.$$

Lem. 4.1 tells how to construct *Regular Simplex* from ETF geometrically. Based on this conclusion, we give the following theorem:

Theorem 4.2 (Feature Learning of Framework (1)). *Assume the feature $f(\cdot;\cdot)$ has d dimension and maximum norm is ρ. $M^\star = [m_1^T, \cdots, m_C^T]^T$ is a $C \times d$ matrix, where each row vector m_i comes from a c-equiangular ETF. Consider the training set $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ with C categories,*

$$\mathcal{L}(\boldsymbol{w}, R, S) \geq \log\left(1 + (C-1)\exp\left[-\frac{C\rho\mathcal{N}(C,c)}{C-1}\right]\right)$$

holds. The equality holds if and only if for any $(\boldsymbol{x}_i, y_i) \in S$, such that

$$Rf(\boldsymbol{x}_i;\boldsymbol{w}) = \rho\frac{m_{y_i}^T - \bar{M}^\star}{\sqrt{\mathcal{N}(C,c)}}$$

where $\bar{M}^\star = \frac{1}{C}\sum_{j=1}^{C} m_j^T$.

Remark 4.3. During training, the features gradually are pushed against the spherical surface with radius ρ, and angles between any two classes get larger and larger. Finally, once the features in every class achieve the *equiangular* property, $i.e.$, collapse to the corresponding class vector in *Regular Simplex*, the loss function would reach the above lower bound.

Based on the above theorem, we know our framework can learn the balanced feature space, $i.e.$, *Regular Simplex*, and lead to *NeurCol* phenomenon in long-tailed learning.

4.2. Two-Parameter Model Stability

Then, we turn to explore generalization of our framework by stability (Hardt et al., 2016; Lei & Ying, 2020). Recall that a part of weights of RBL is parameterized as an orthogonal matrix by substitution techniques. Therefore, the standard analysis of model stability can not be applied to RBL. To this end, we extend the previous analysis paradigm (Lei & Ying, 2020) to propose *Two-Parameter Model Stability*, which analyzes model stability by splitting the model parameter into two parts. During optimization, two parts of parameters are performed with different update rules.

First, we consider the common case, $i.e.$, both parameters are updated without any constraints. Define the loss function $g(\boldsymbol{x};\boldsymbol{w};T)$, where $\boldsymbol{x}$ is a sample and $\boldsymbol{w}$, T are the model parameters. Given a dataset $S = \{\boldsymbol{x}_i\}_{i=1}^{N}$, the SGD update rule of $\min_{\boldsymbol{w},T} \hat{E}_{x\in S}\, g(\boldsymbol{x};\boldsymbol{w};T)$ is given by

Definition 4.4 (Update Rule of Two-Parameter Model).

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta_t^w \partial_{\boldsymbol{w}_t} g(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;M_t)$$

$$T_{t+1} \leftarrow T_t - \eta_t^T \partial_{T_t} g(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;T_t)$$

In the t-th iteration, the i_t-th data in S are sampled uniformly to perform optimization. The learning rates of $\boldsymbol{w}$ and T are η_t^w and η_t^T respectively. And $\partial_{\boldsymbol{w}_t} g(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;M_t)$ and $\partial_{T_t} g(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;T_t)$ are the sub-gradients of $g(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;T_t)$ w.r.t $\boldsymbol{w}$ and T.

To obtain the model stabilities of $\boldsymbol{w}$ and T, introduce another dataset $S^{(i)} = S \setminus \{\boldsymbol{x}_i\} \bigcup \{\tilde{\boldsymbol{x}}_i\}$, which only differs i-th sample from S and follows from the same distribution with S. We denote $\tilde{S} = \{\tilde{\boldsymbol{x}}_i\}_i^N$. Then for random algorithm Def. 4.4, we use empirical risks $G_S(\boldsymbol{w};T) = \frac{1}{N}\sum_{i=1}^{N} g(\boldsymbol{x}_i;\boldsymbol{w};T)$ to bound its stabilities.

Theorem 4.5 (Two-Parameter ℓ_1 Model Stability). *Consider the two groups of parameters $(\boldsymbol{w}_t, T_t)$ and $(\boldsymbol{w}_t^{(i)}, T_t^{(i)})$ trained on S and $S^{(i)}$ from the same starting point by the update rule Def. 4.4, assume*

- *$g(\boldsymbol{x};\boldsymbol{w};T)$ is nonnegative for any $\boldsymbol{x}$, $\boldsymbol{w}$ and T;*
- *$T \mapsto g(\boldsymbol{x};\boldsymbol{w};T)$ is L_T-smooth for any $\boldsymbol{w}$ and $\boldsymbol{x}$;*
- *$\boldsymbol{w} \mapsto \partial_w g(\boldsymbol{x};\boldsymbol{w};T)$ is ℓ_w-lipschitz for any T and $\boldsymbol{x}$;*
- *$\boldsymbol{w} \mapsto g(\boldsymbol{x};\boldsymbol{w};T)$ is L_w-smooth for any T and $\boldsymbol{x}$;*
- *$T \mapsto \partial_T g(\boldsymbol{x};\boldsymbol{w};T)$ is ℓ_T-lipschitz for any $\boldsymbol{w}$ and $\boldsymbol{x}$.*

We denote $\boldsymbol{v} = [\eta_t^T\sqrt{2L_T}, \eta_t^w\sqrt{2L_w}]^T$ and

$$F = \begin{bmatrix} 1 + \frac{N-1}{N}\eta_t^T L_T & \frac{N-1}{N}\eta_t^T \ell_w \\ \frac{N-1}{N}\eta_t^w \ell_T & 1 + \frac{N-1}{N}\eta_t^w L_w \end{bmatrix}$$

Then, if $\eta_t^T/\eta_t^w = \ell_T/\ell_w$ holds, the ℓ_1 model stabilities of $\boldsymbol{w}$ and T is given by

$$\begin{bmatrix} \mathbb{E}_{S,\tilde{S},A}\left[\frac{1}{N}\sum_{i=1}^N \|T_{t+1} - T_{t+1}^{(i)}\|\right] \\ \mathbb{E}_{S,\tilde{S},A}\left[\frac{1}{N}\sum_{i=1}^N \|\boldsymbol{w}_{t+1} - \boldsymbol{w}_{t+1}^{(i)}\|\right] \end{bmatrix} \leq$$

$$\frac{2}{N}\sum_{j=1}^{t}\left(\lambda_1^{t-j}\boldsymbol{p}_1\boldsymbol{p}_1^T\boldsymbol{v} + \lambda_2^{t-j}\boldsymbol{p}_2\boldsymbol{p}_2^T\boldsymbol{v}\right)\mathbb{E}_{S,A}\left[\sqrt{G_S(\boldsymbol{w}_j;T_j)}\right]$$

where λ_1, λ_2 and $\boldsymbol{p}_1, \boldsymbol{p}_2$ are eigen values and eigen vectors of F respectively.

Then we consider the *Two-Parameter Model Stability* with the one constrained parameter. In our framework, the linear classifier R is restricted as orthogonal matrices. Suppose we parameterize $R \in SO(d)$ (whether R is a square matrix does not affect the analysis) by the mapping $R = \exp\left(B - B^T\right)$, where $B \in \mathbb{R}^{d\times d}$, the SGD update rule of RBL is given by

Definition 4.6 (Update Rule of Framework (1)).

$$\begin{aligned} \boldsymbol{w}_{t+1} &\leftarrow \boldsymbol{w}_t - \eta_t^w \partial_{\boldsymbol{w}_t} f(\boldsymbol{x}_{i_t}; \boldsymbol{w}_t; R_t) \\ B_{t+1} &\leftarrow B_t - \eta_t^T \partial_{B_t} f(\boldsymbol{x}_{i_t}; \boldsymbol{w}_t; R_t) \\ T_{t+1} &\leftarrow B_{t+1} - B_{t+1}^T \\ R_{t+1} &\leftarrow \exp(T_{t+1}) \end{aligned}$$

Here, $f(\boldsymbol{x};\boldsymbol{w};R)$ indicates the loss function, and we use $F_S(\boldsymbol{w},R)$ to represent the empirical risk $\frac{1}{N}\sum_{i=1}^N f(\boldsymbol{x}_i;\boldsymbol{w};R)$ over S. The update rule of R is not standard due to the parameterization, while B performs the standard SGD update. Then ℓ_1 model stability of our framework is as follows.

Theorem 4.7 (Two-Parameter ℓ_1 Model Stability of Framework (1)). *Consider the two groups of parameters $(\boldsymbol{w}_t, T_t)$ and $(\boldsymbol{w}_t^{(i)}, T_t^{(i)})$ trained on S and $S^{(i)}$ from the same starting point by the update rule Def. 4.6. Then let*

$$\boldsymbol{w} \to f(\boldsymbol{x};\boldsymbol{w};e^T), T \to f(\boldsymbol{x};\boldsymbol{w};e^T)$$

of Def.4.6 be

$$\boldsymbol{w} \to g(\boldsymbol{x};\boldsymbol{w};T), T \to g(\boldsymbol{x};\boldsymbol{w};T)$$

of Def.4.4. Assume T_t lies in a bounded space $\Omega \subset \mathfrak{so}(d)$ for all t (see Assumption.D.2 of Appendices for more details) and all assumptions in Thm.4.5 holds. Denote $\lambda_1^{k-j}\boldsymbol{p}_1\boldsymbol{p}_1^T\boldsymbol{v} + \lambda_2^{k-j}\boldsymbol{p}_2\boldsymbol{p}_2^T\boldsymbol{v}$ as $\boldsymbol{p}(k,j)$. Then the ℓ_1 model stability of parameter R is given by

$$\mathop{\mathbb{E}}_{S,\tilde{S},A}\left[\frac{1}{N}\sum_{i=1}^{N}\|R_{t+1} - R_{t+1}^{(i)}\|\right] \leq$$

$$\frac{2\mathcal{H}(\Omega)L_\phi}{N}\sum_{j=1}^{t}(h_1(j) + h_2(j))\mathbb{E}_{S,A}\left[\sqrt{F_S(\boldsymbol{w}_j, R_j)}\right]$$

where $h_1(j) = (N-1)\sum_{k=1}^{t-j+1} N^{k-t-1}\boldsymbol{p}(k,j)_1$ and $h_2(j) = \sqrt{2L_T}N^{j-t}\eta_j$.

Remark 4.8. According to Thm.2 (a) of (Lei & Ying, 2020), once the first order gradient of $f(\cdot;\boldsymbol{w};R)$ is bounded, one can obtain the generalization error, which is proportional to the ℓ_1 model stabilities. Thus, if the training set is large enough, our method can reach a reasonable generalization result.

5. Experiments

To illustrate our method's effectiveness empirically, we conduct a series of experiments.

5.1. Dataset

We use several benchmark datasets in our experiments, including CIFAR10/CIFAR100 (Krizhevsky et al., 2009), long-tailed ImageNet (Liu et al., 2019a) and long-tailed Places (Liu et al., 2019b). We use the imbalanced ratio to represent how imbalanced a dataset is, which is the ratio of the samples between the most-frequent class and the rarest class in the dataset.

Long-Tailed CIFAR CIFAR10/CIFAR100 both contain 60000 images of size 32×32, where 10000 of them are for testing and 50000 for training. Note that the labels in the original CIFAR10/CIFAR100 dataset are uniformly distributed, so we generate long-tailed versions from the original data. Refer to experiments of other studies, we use exponential decay (Cui et al., 2019) to generate long-tailed training set with $50, 100, 200$ imbalance ratios for both datasets, and keep the test sets unchanged.

Long-Tailed ImageNet ImageNet-LT is the long-tailed version of ImageNet-1K (Russakovsky et al., 2015). ImageNet-LT has 1K categories and 115.8K images for training, where the imbalance ratio is 256. Besides, the valid set and test set of ImageNet-LT have 20 and 50 images for each category respectively.

Long-Tailed Places The training set of Places-LT has $62,500$ images for 365 classes, where the most frequently occured classes have $4,980$ images and the least has 5 images. The imbalance ratio is 996. The valid and test sets are balanced and contain 20 and 100 images per class respectively.

5.2. Experimental Setups

Competitors We devide competitors into two technical routes: **1)**. Class Balanced Learning inlcuding Class-Balanced loss (Cui et al., 2019), Calibrated (Xu et al., 2021), Decoupling-NCM, LWS, cRT, τ-norm (Kang et al., 2019), seesaw (Wang et al., 2021a), BalancedSoftmax (Ren et al., 2020), MARC (Wang et al., 2021c), LADE (Hong et al.,

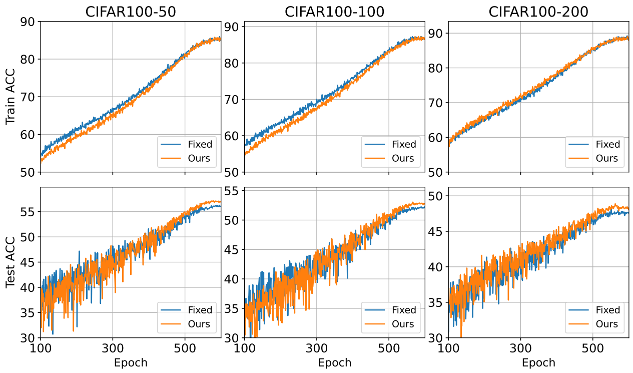


Figure 3: Generalization analysis on CIFAR100. The two rows show the accuracies of *Fixed* and our method on training set and test set in every epoch respectively.

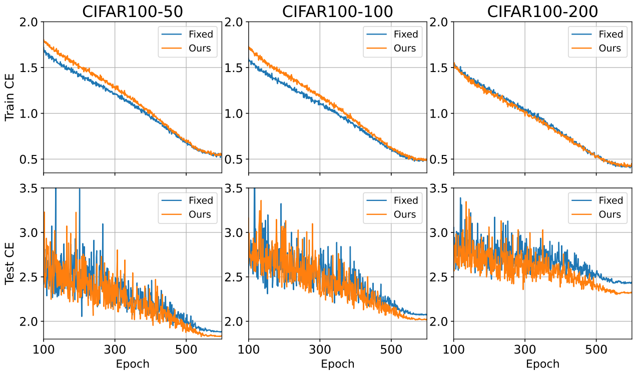


Figure 4: Generalization analysis on CIFAR100. The two rows show the cross entropy loss of *Fixed* and our method on training set and test set in every epoch respectively.

Table 1: Test accuracies on CIFAR10/100-LT. The best and second best results are marked as **bold** and underline. Rows with † denote results borrowed from (Wang et al., 2021c). Results of other competitors are taken from original papers.

Method	CIFAR-10			CIFAR-100		
	50	100	200	50	100	200
CB	79.3	74.6	68.9	45.3	39.6	36.2
LADE	-	-	-	50.5	45.4	-
Calibrated	84.3	82.8	78.5	51.1	45.5	42.1
cRT†	-	82.0	76.6	-	50.0	44.5
LWS†	-	83.7	78.1	-	50.5	45.3
BS†	-	83.1	79.0	-	50.3	45.9
MARC	-	**85.3**	81.1	-	50.8	47.4
HCL	85.4	81.4	-	51.9	46.7	-
TSC	82.9	79.7	-	47.4	43.8	-
Fixed	87.1	84.0	80.2	56.2	52.3	47.2
RBL	**87.6**	84.7	**81.2**	**57.2**	**53.1**	**48.9**

Table 2: Test accuracies on ImageNet-LT. The best and second best results are marked as **bold** and underline. Rows with † denote results borrowed from (Wang et al., 2021c). Results of other competitors are taken from original papers.

Method	Many	Medium	Few	All
Calibrated	-	-	-	48.4
cRT	61.8	46.2	27.4	49.6
LWS	60.2	47.2	30.3	49.9
Seesaw	**67.1**	45.2	21.4	50.4
BS†	62.2	48.8	29.8	51.4
MARC	60.4	50.3	**36.6**	52.3
LADE	65.1	48.9	33.4	53.0
KCL	61.8	49.4	30.9	51.5
TSC	63.5	**49.7**	30.4	52.4
Fixed	64.3	47.6	27.2	51.2
RBL	64.8	49.6	34.2	**53.3**

2021); **2)**. Contrastive Learning inlcuding TSC (Li et al., 2022), HCL (Wang et al., 2021b), KCL (Kang et al., 2020). Besides, we also compare *Fixed* (Yang et al., 2022b) as baseline. We implement (Yang et al., 2022b) with our framework, where the orthogonal matrix is fixed during training and Logit Adjustment is performed during testing.

Implementation details In our methods, dimensions of the feature is important. To obtain feature with different dimensions, a linear layer that transforms feature dimensions is added after backbone. Following previous work, for CIFAR10/100-LT, we use ResNet-32 with 256 feature dimensions as the backbone; for ImageNet-LT, we use ResNext-50 with 512 feature dimensions as the backbone; for Place-LT, we use a pretrained Resnet-152 with 512 feature dimension as the backbone. We utilize SGD optimization for all experiments. For CIFAR10/100-LT, the model is trained for 600 epochs with batch size 256. Besides, the learning rate linearly warm up from 0.05 to 0.1 within the first 8 epochs, and then decays to zero in cosine decay scheme. For ImageNet-LT, we train the model for 200 epochs with batch size 64. The learning rate is set as 0.25 and decays to zeros by cosine decay during training. For Places-LT, we train the model with learning rate 3.5×10^{-3} and batch size 64 for 30 epochs. For all datasets, the weight decay and momentum are set as 0.0005 and 0.9. As for data augmentation, for CIFAR10/100-LT, we perform AutoAugment (Cubuk et al., 2019); for ImageNet-LT, we perform several common augmentation methods including RandomCrop, RandomFlip, and ColorJitter; and for Places-LT, we use RandAug (Cubuk et al., 2020).

5.3. Evaluation protocols

We use balanced accuracy as the evaluation metric. Since the test dataset in experiments have the same number of samples for each class, the standard accuracy calculated on

Table 3: Test accuracies on Places-LT. The best and second best results are marked as **bold** and underline. Results of other competitors are taken from original papers.

Method	Many	Medium	Few	All
NCM	40.4	37.1	27.3	36.4
cRT	42.0	37.6	24.9	36.7
LWS	40.6	39.1	28.6	37.6
τ-norm	37.8	**40.7**	31.8	37.9
Marc	39.9	39.8	**32.6**	38.4
BS	41.2	39.8	31.6	38.7
LADE	42.8	39.0	31.2	**38.8**
Fixed	43.7	39.7	23.9	38.0
RBL	**44.1**	**40.7**	24.4	**38.8**

them is balanced. In addition, we also provide accuracies over three different subsets of test set: Many-shot, Medium-shot, and Few-shot. Many-shot subset only consists of the classes that have more than 100 samples in training set, while Medium-shot and Few-shot consist of the class that has 20 $\sim$ 100 samples and less than 20 samples respectively. Since CIFAR10/100 datasets have no valid set, we report the highest accuracy on test set. And for ImageNet-LT, we keep the model with the best performance on valid set, and report its accuracy metric over entire test set.

5.4. Experimental Results

The performance comparisons are shown in Tab.1, 2 and 3. Except that *Fixed* is our implementation, all results of other methods come from the original paper. We have three observations from the results: **1**). Ours generally outperforms previous methods, which validates the effectiveness of our method. **2**). In Tab.2, four accuracy results come from a single experiment, rather than four independent experiments. Therefore, the accuracy result of RBL over ”All” in Tab.2 can be seen as a weighted average of accuracies on each subset. Our results on Many, Medium, and Few shot are not outstanding but still have a high rank compared with other methods. As a result, it has the highest accuracy on all test set. This shows that our method is balanced for head classes and tail classes, could achieve good trade-off between them. **3**). In CIFAR10/100, both *Fixed* and Ours can achieve SOTA level, whereas in the larger scale dataset, ImageNet-LT, *Fixed* is completely inferior to Ours. This phenomenon shows that the fixed feature space is far from enough for long-tailed feature learning.

5.5. Ablation Experiments

We conduct ablation experiments on three long-tailed CIFAR100 datasets, and the results are presented in Tab.4, where Fixed and Learnable Direction denote if the direction of ETF is learnable and LD denotes the post-hoc Logit Adjustment. There are three observations:

Table 4: Ablation study on CIFAR100-LT measured by test accuracies.

Method	CIFAR100-LT		
	200	100	50
CE (Baseline)	42.7	46.7	51.8
Fixed Direction	41.7	46.5	50.5
Learnable Direction	43.4	47.7	52.9
LD	46.6	51.4	55.1
Fixed Direction + LD	47.2	52.3	56.2
Learnable Direction + LD (RBL)	**48.9**	**53.1**	**57.2**

- When LD is removed, the performance rank is Fixed Direction $<$ CE $<$ Learnable Direction.
- Regardless of whether the LD method is used, Learnable Direction always outperforms Fixed Direction.
- Both Fixed and Learnable Direction exhibit substantial improvements when using LD.

The first finding supports our argument that the fixed feature can be detrimental to the model’s performance. The second finding suggests that a learnable rotation matrix is crucial for improving model performance.

We would like to emphasize and explain the third finding. The readers might contend that the state-of-the-art performance of our method is primarily due to the LD. However, LD only works during testing and does not affect the training process. Our approach can only learn a Balanced Feature with a proper direction, which requires appropriate decision-making to harness its power. This point is supported by the work of Kang et al. (2019), which highlights that even if good features are learned, reasonable long-tailed recognition performance cannot be achieved without a good decision-making process in the last layer.

5.6. Generalization Analysis

To study the generalization of our method, we report the accuracies of RBL and *Fixed* on CIFAR100 in every epoch. As shown in Fig.3 and 4, we observe after both methods converge, their accuracies on training set are almost the same. Yet, RBL always has higher accuracy on validation set (CE loss is the same). This indicates that our RBL can generalize better on unseen data. We hold the opinion that the generalization of the model may be harmed by the fixed feature, especially when the model size is limited. *Fixed* requires a more powerful feature extractor to compensate for the fixed feature, since the random generated ETF can hardly be the best.

6. Conclusion

In this paper, we argue that in classification problems, both the structure and direction of features are import. The *Equiangular Tight Frame* with fixed directions is totally inadequate. In order to learn ETF with arbitrary directions, we introduce *Representation-Balanced Learning* Framework, which can register the angle between ETF and model features through learnable orthogonal matrices. Our idea could be devided into two steps: First, according to different cases of class number and feature dimension number, we generate equiangular structrues as the balanced feature to be learned. Next, to avoid fixed features weakening the learning ability, we introduce orthogonal matrices to learn both the structure and direction of ETF. Theoretically, without assuming that the training set is uniform, we prove that our framework could achieve same effect to *NeurCol* phenomenon. To analyze the generalization performance of our framework, we propose *Two-Parameter Model Stability*. It provides a new perspective to analyze the stability of parameters with constraint. Finally, we conduct a series experiments to demonstrate advantages of our method.

7. Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0102000, in part by National Natural Science Foundation of China: 62236008, U21B2038, 61931008, 6212200758 and 61976202, in part by the Fundamental Research Funds for the Central Universities, in part by Youth Innovation Promotion Association CAS, in part by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDB28000000.

References

Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

Cui, J., Zhong, Z., Liu, S., Yu, B., and Jia, J. Parametric contrastive learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 715–724, 2021.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. 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.

Fickus, M., Jasper, J., King, E. J., and Mixon, D. G. Equiangular tight frames that contain regular simplices. *Linear Algebra and its applications*, 555:98–138, 2018.

Graf, F., Hofer, C., Niethammer, M., and Kwitt, R. Dissecting supervised constrastive learning. In *International Conference on Machine Learning*, pp. 3821–3830. PMLR, 2021.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pp. 1225–1234. PMLR, 2016.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Hong, Y., Han, S., Choi, K., Seo, S., Kim, B., and Chang, B. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6626–6636, 2021.

Jiang, Z., Chen, T., Mortazavi, B. J., and Wang, Z. Self-damaging contrastive learning. In *International Conference on Machine Learning*, pp. 4927–4939. PMLR, 2021.

Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., and Kalantidis, Y. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

Kang, B., Li, Y., Xie, S., Yuan, Z., and Feng, J. Exploring balanced feature spaces for representation learning. In *International Conference on Learning Representations*, 2020.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lei, Y. and Ying, Y. Fine-grained analysis of stability and generalization for stochastic gradient descent. In *International Conference on Machine Learning*, pp. 5809–5819. PMLR, 2020.

Lezcano-Casado, M. Trivializations for gradient-based optimization on manifolds. In *Advances in Neural Information Processing Systems, NeurIPS*, pp. 9154–9164, 2019.

Lezcano-Casado, M. and Martınez-Rubio, D. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *International Conference on Machine Learning*, pp. 3794–3803. PMLR, 2019.

Li, T., Cao, P., Yuan, Y., Fan, L., Yang, Y., Feris, R. S., Indyk, P., and Katabi, D. Targeted supervised contrastive learning for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6918–6928, 2022.

Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., and Yu, S. X. 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, 2019a.

Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., and Yu, S. X. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019b.

Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and Van Der Maaten, L. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 181–196, 2018.

Menon, A. K., Jayasumana, S., Rawat, A. S., Jain, H., Veit, A., and Kumar, S. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

Papyan, V., Han, X., and Donoho, D. L. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

Ren, J., Yu, C., Ma, X., Zhao, H., Yi, S., et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33: 4175–4186, 2020.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Sustik, M. A., Tropp, J. A., Dhillon, I. S., and Heath Jr, R. W. On the existence of equiangular tight frames. *Linear Algebra and its applications*, 426(2-3):619–635, 2007.

Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C., and Yan, J. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11662–11671, 2020.

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

Wang, J., Zhang, W., Zang, Y., Cao, Y., Pang, J., Gong, T., Chen, K., Liu, Z., Loy, C. C., and Lin, D. Seesaw loss for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9695–9704, 2021a.

Wang, P., Han, K., Wei, X.-S., Zhang, L., and Wang, L. Contrastive learning based hybrid networks for long-tailed image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 943–952, 2021b.

Wang, T., Li, Y., Kang, B., Li, J., Liew, J., Tang, S., Hoi, S., and Feng, J. The devil is in classification: A simple framework for long-tail instance segmentation. In *European conference on computer vision*, pp. 728–744. Springer, 2020.

Wang, Y., Zhang, B., Hou, W., Wu, Z., Wang, J., and Shinozaki, T. Margin calibration for long-tailed visual recognition. *arXiv preprint arXiv:2112.07225*, 2021c.

Xu, Z., Chai, Z., and Yuan, C. Towards calibrated model for long-tailed visual recognition from prior perspective. *Advances in Neural Information Processing Systems*, 34: 7139–7152, 2021.

Yang, L., Jiang, H., Song, Q., and Guo, J. A survey on long-tailed visual recognition. *International Journal of Computer Vision*, pp. 1–36, 2022a.

Yang, Y., Xie, L., Chen, S., Li, X., Lin, Z., and Tao, D. Do we really need a learnable classifier at the end of deep neural network? *arXiv preprint arXiv:2203.09081*, 2022b.

Zhang, Y., Kang, B., Hooi, B., Yan, S., and Feng, J. Deep long-tailed learning: A survey. *arXiv preprint arXiv:2110.04596*, 2021.

Zhu, J., Wang, Z., Chen, J., Chen, Y.-P. P., and Jiang, Y.-G. Balanced contrastive learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6908–6917, 2022.

A. Related Work

We give a brief overview of long-tail learning from three perspectives: class-balanced methods, supervised contrastive learning, and feature learning. More comprehensive overviews of long-tailed learning could be found in (Zhang et al., 2021; Yang et al., 2022a).

Class-Balanced Methods The sampling-based approach is the most intuitive (Mahajan et al., 2018; Kang et al., 2019; Wang et al., 2020). The common sampling strategies include instance-balanced sampling and class-balanced sampling. Instance-balanced sampling samples every instance in datasets with equal probability, while in class-balanced sampling, every class has an equal probability to be sampled. On the basis of instance-balanced sampling, (Mahajan et al., 2018) proposes to use the square root of the number of samples as the sampling probability. These methods make training set more balanced by different sampling schemes, which could improve accuracy of tail classes. However, it is found that there exists a performance trade-off between head class and tail class for sampling-based methods (Kang et al., 2019). Another idea for solving the long-tailed problem is to balance the cost of different classes by designing a balanced loss function (Tan et al., 2020; Cui et al., 2019; Cao et al., 2019; Ren et al., 2020; Menon et al., 2020). From the perspective of data overlay, (Cui et al., 2019) proposes to measure the volume of datasets by effective number. Effective number is defined as a function of the number of samples. Then they use effective number to re-weigh the losses of different classes. Further, (Cao et al., 2019) proposed LDAM loss. It achieves the lower bound of margin-based generalization bound by adding a set of well-designed margins. Then, (Ren et al., 2020; Menon et al., 2020) point out that existing methods based on loss weighting and margin modification can not achieve Fisher consistency. Inspired by this, they propose Balanced Softmax and Logit Adjustment that could minimize balanced error consistently.

Supervised Contrastive Learning Contrastive Learning is an implementation way of Unsupervised Learning, which aims to learn features by narrowing the distance between similar samples and enlarging the distance between different samples. SCL (Khosla et al., 2020) pioneered the use of contrastive learning in supervised classification problems. Later, a group of methods based on Supervised Contrastive Loss were proposed (Cui et al., 2021; Kang et al., 2020; Li et al., 2022; Zhu et al., 2022; Jiang et al., 2021). The above approaches target to learn balanced feature space to improve the performance in long-tailed scenes. KCL (Kang et al., 2020) fixes the number of positive samples in each batch to learn the balanced representation. Paco (Cui et al., 2021) is a contrastive learning framework based on Maco (He et al., 2020). They introduce parametric class-wise learnable centers, which could enhance the learning for hard examples and imbalanced data. Similarly, (Li et al., 2022; Zhu et al., 2022) are motivated by (Papyan et al., 2020)'research of features, introduces the concept of category center in contrastive loss as well. TSC (Li et al., 2022) generates features uniformly distributed on the sphere in advance as the category center, while BCL (Zhu et al., 2022) uses the classifier weight as the category prototype.

Representation Learning (Kang et al., 2019) pioneers two-stages training in long-tailed learning. They found that only finetuning the classifier could achieve satisfactory performance. Besides, (Papyan et al., 2020) found *Neural Collapse* phenomenon in classification. Specifically, as the cross entropy converges to zeros, the feature that model learned would form an ETF. which does not vary with dataset scales and backbones. Going a step further, (Graf et al., 2021) obtains a similar conclusion in theory. Based on their conclusions, TSC and BCL are proposed to learn the balanced feature for long-tailed learning. The recently proposed *fixed* (Yang et al., 2022b) proposes to learn ETF directly by fixing the classifier as ETF. We argue that only ETF structure is not enough for the feature learning of classification. Therefore, we propose RBL that can learn the ETF with any direction.

B. The proofs of Lem. 4.1 and Thm. 4.2

Before we give the proof of Thm. 4.2, we prove Lem. 4.1 first.

Restatement of Theorem 4.1. *Consider a c-equiangular ETF $\{\zeta_i\}_{i=1}^C$, where $\langle \zeta_i, \zeta_j \rangle = c$ for any $i, j (i \neq j)$. Define its means as $\bar{\zeta} = \frac{1}{C}\sum_{i=1}^C \zeta_i$. Then the frame $\{\zeta_i - \bar{\zeta}\}_{i=1}^C$ forms a Regular Simplex, which means*

$$(\boldsymbol{S1}). \sum_{i=1}^{C}(\zeta_i - \bar{\zeta}) = \mathbf{0} \qquad (\textit{zero mean})$$

$$(\boldsymbol{S2}). \|\zeta_i - \bar{\zeta}\| = \sqrt{\mathcal{N}(C,c)}, \forall i \qquad (\textit{equalnorm})$$

$$(\boldsymbol{S3}). \langle \zeta_i - \bar{\zeta}, \zeta_j - \bar{\zeta} \rangle = \mathcal{A}(C,c), \forall i \neq j \qquad (\textit{equiangular})$$

where

$$\mathcal{A}(C, c) = c - \frac{1 + (C - c)}{C}$$

and

$$\mathcal{N}(C, c) = 1 - \frac{1 + (C - c)}{C}.$$

Proof. (**S1**) is clear. We prove the *equiangular* property (**S3**) first. According to the c-*equiangular* property of ETF, we know $\forall i, j (i \neq j)$, $\langle \zeta_i, \zeta_j \rangle = c$. Given any $i, j (i \neq j)$, we have

$$\begin{aligned}
\langle \zeta_i - \bar{\zeta}, \zeta_j - \bar{\zeta} \rangle &= \langle \zeta_i, \zeta_j \rangle - \langle \bar{\zeta}, \zeta_j \rangle - \langle \zeta_i, \bar{\zeta} \rangle + \langle \bar{\zeta}, \bar{\zeta} \rangle \\
&= c - \frac{1}{C} \sum_{k=1}^{C} \langle \zeta_k, \zeta_j \rangle - \frac{1}{C} \sum_{k=1}^{C} \langle \zeta_i, \zeta_k \rangle + \frac{1}{C^2} \sum_{k=1}^{C} \sum_{v=1}^{C} \langle \zeta_k, \zeta_v \rangle \\
&= c - 2\frac{1 + (C-1)c}{C} + \frac{C(1 + (C-1)c)}{C^2} \\
&= c - \frac{1 + (C-1)c}{C}
\end{aligned}$$

The proof of equalnorm property (**S2**) is similar. Consider a vector $\zeta_i - \bar{\zeta}$, we have

$$\begin{aligned}
\|\zeta_i - \bar{\zeta}\|^2 &= \langle \zeta_i, \zeta_i \rangle - 2\langle \zeta_i, \bar{\zeta} \rangle + \langle \bar{\zeta}, \bar{\zeta} \rangle \\
&= 1 - 2\frac{1 + (C-1)c}{C} + \frac{C(1 + (C-1)c)}{C^2} \\
&= 1 - \frac{1 + (C-1)c}{C}
\end{aligned}$$

□

Then, we will give the proof of Thm. 4.2.

Restatement of Theorem 4.2. *Assume the feature $f(\cdot;\cdot)$ has d dimension and maximum norm is ρ. $M^\star = [m_1^T, \cdots, m_C^T]^T$ is a $C \times d$ matrix, where each row vector m_i comes from a c-equiangular ETF. Consider the training set $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ with C categories,*

$$\mathcal{L}(\boldsymbol{w}, R, S) \geq \log\left(1 + (C-1)\exp\Big[-\frac{C\rho\mathcal{N}(C, c)}{C-1}\Big]\right)$$

holds. The equality holds if and only if for any $(\boldsymbol{x}_i, y_i) \in S$, such that

$$Rf(\boldsymbol{x}_i; \boldsymbol{w}) = \rho\frac{m_{y_i}^T - \bar{M}^\star}{\sqrt{\mathcal{N}(C, c)}}$$

Proof. Recall our optimization objective in (1) is

$$\begin{aligned}
\min_{\boldsymbol{w}, R} \quad & \mathcal{L}(\boldsymbol{w}, R, S) := -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp([\text{logit}(\boldsymbol{x})]_{y_i})}{\sum_{y=1}^{C} \exp([\text{logit}(\boldsymbol{x})]_y)} \\
s.t. \quad & \text{logit}(\boldsymbol{x}_i) = M^\star Rf(\boldsymbol{x}_i; \boldsymbol{w})
\end{aligned}$$

where R, w is the optimization variables and R is restricted as orthogonal matrix. $M^\star$ is pre-computed ETF and keeps fixed during training.

$$
\begin{aligned}
&-\frac{1}{N}\sum_{i=1}^{N}\log\left\{\frac{\exp\Big([\textit{logit}(\boldsymbol{x}_i)]_{y_i}\Big)}{\sum_{j=1}^{C}\exp\Big([\textit{logit}(\boldsymbol{x}_i)]_j\Big)}\right\}\\
=&\frac{1}{N}\sum_{i=1}^{N}\log\left(1+\sum_{j\neq y_i, j\in[C]}\exp\Big([\textit{logit}(\boldsymbol{x}_i)]_j-[\textit{logit}(\boldsymbol{x}_i)]_{y_i}\Big)\right)\\
\overset{\textbf{C1}}{\geq}&\frac{1}{N}\sum_{i=1}^{N}\log\left(1+(C-1)\exp\Big[\frac{1}{C-1}\sum_{j\neq y_i, j\in[C]}[\textit{logit}(\boldsymbol{x}_i)]_j-[\textit{logit}(\boldsymbol{x}_i)]_{y_i}\Big]\right)\\
\overset{\textbf{C2}}{\geq}&\log\left(1+(C-1)\exp\Big[\frac{1}{(C-1)N}\sum_{i=1}^{N}\sum_{j\neq y_i, j\in[C]}[\textit{logit}(\boldsymbol{x}_i)]_j-[\textit{logit}(\boldsymbol{x}_i)]_{y_i}\Big]\right)\\
=&\log\left(1+(C-1)\exp\Big[\frac{1}{(C-1)N}\sum_{i=1}^{N}\sum_{j\in[C]}[\textit{logit}(\boldsymbol{x}_i)]_j-[\textit{logit}(\boldsymbol{x}_i)]_{y_i}\Big]\right)
\end{aligned}
$$

The inequalities **C1** and **C2** follow from Jensen's inequality. For inequality **C1**, we know $t \to \exp(t)$ is a convex function, so the equality in **C1** holds when

$$(\textbf{C1}).\forall i \in [N], \exists M_i \in \mathbb{R}, \forall y \in [C](y \neq y_i), \Big[\textit{logit}(\boldsymbol{x}_i)\Big]_y = M_i$$

Then for inequality **C2** is due to the convex function $t \to \log(1+\exp(t))$. Therefore, the equality in **C2** holds when

$$(\textbf{C2}).\exists M \in \mathbb{R}, \forall i \in [N], \sum_{j\neq y_i, j\in[C]}\Big([\textit{logit}(\boldsymbol{x}_i)]_{y_i}-[\textit{logit}(\boldsymbol{x}_i)]_j\Big) = M$$

The function $t \to \log(1+\exp(t))$ is monotonically increasing, so next we try to bound $\sum_{i=1}^{N}\sum_{j\in[C]}[\textit{logit}(\boldsymbol{x}_i)]_j - [\textit{logit}(\boldsymbol{x}_i)]_{y_i}$. To illustrate the role of ETF $M^\star$, we denote

$$\textit{logit}(\boldsymbol{x}_i) = M^\star Rf(\boldsymbol{x}_i, w) = \Big[m_1 Rf(\boldsymbol{x}_i, w), \ldots, m_C Rf(\boldsymbol{x}_i, w)\Big]^T = \Big[\langle m_1^T, Rf(\boldsymbol{x}_i, w)\rangle, \ldots, \langle m_C^T, Rf(\boldsymbol{x}_i, w)\rangle\Big]^T,$$

then

$$
\begin{aligned}
\sum_{i=1}^{N}\sum_{j\in[C]}[\textit{logit}(\boldsymbol{x}_i)]_j-[\textit{logit}(\boldsymbol{x}_i)]_{y_i} &= \sum_{i=1}^{N}\sum_{j\in[C]}\langle m_j^T, Rf(\boldsymbol{x}_i, w)\rangle - \langle m_{y_i}^T, Rf(\boldsymbol{x}_i, w)\rangle\\
&= C\sum_{i=1}^{N}\langle(\bar{M}^\star - m_{y_i}^T), Rf(\boldsymbol{x}_i, w)\rangle\\
&\overset{\textbf{C3}}{\geq} -C\sum_{i=1}^{N}\|(\bar{M}^\star - m_{y_i}^T)\|\|Rf(\boldsymbol{x}_i, w)\|\\
&\overset{\textbf{C4}}{\geq} -C\rho\sum_{i=1}^{N}\|(\bar{M}^\star - m_{y_i}^T)\|
\end{aligned}
$$

where we denote $\frac{1}{C}\sum_{j=1}^{C} m_j^T$ as $\bar{M}^\star$. The inequality **C3** follows from the Cauchy-Schwarz inequality, equality holds if and only if

$$(\textbf{C3}).\forall i \in [N], \exists \lambda_i \in \mathbb{R}^+, Rf(\boldsymbol{x}_i, w) = \lambda_i(m_{y_i}^T - \bar{M}^\star)$$

The inequality **C4** follows from the assumption for feature extractor, with equality if and only if

$$(\textbf{C4}).\forall i \in [N], \|Rf(\boldsymbol{x}; \boldsymbol{w})\| = \rho$$

To explore what the feature $f(x)$ is look like when the objective function reach its minimum, we check the conditions **C1** - **C4** that make the equality hold. We start from **C3** and **C4**. Given a sample $(\boldsymbol{x}_i, y_i)$, we have

$$\|Rf(\boldsymbol{x}_i;\boldsymbol{w})\| = \lambda_i \|m_{y_i}^T - \bar{M}^\star\| = \rho$$

For the simplicity of result, we denote $1 - \frac{1+(C-1)c}{C}$ by $\mathcal{N}(C,c)$ and $c - \frac{1+(C-1)c}{C}$ by $\mathcal{A}(C,c)$. So, for any $i \in [N]$

$$\lambda_i = \frac{\rho}{\|m_{y_i}^T - \bar{M}^\star\|} = \frac{\rho}{\sqrt{1 - \frac{1+(C-1)c}{C}}} = \frac{\rho}{\sqrt{\mathcal{N}(C,c)}}$$

Recall Lem. 4.1, we know the sequence of vector $\{m_y^T - \bar{M}^\star\}_{y=1}^C$ is also equalnorm and equiangular. Therefore, the learned feature of the framework is equalnorm and equiangular. Then return to the logit of sample $(\boldsymbol{x}_i, y_i)$, we have

$$\begin{aligned}
&\text{if } y \neq y_i, \Big[\text{logit}(\boldsymbol{x}_i)\Big]_y = \langle m_y^T, \lambda_i(m_{y_i}^T - \bar{M}^\star)\rangle = \lambda_i m_y(m_{y_i}^T - \bar{M}^\star) = \lambda_i \mathcal{A}(C,c) = \rho\frac{\mathcal{A}(C,c)}{\sqrt{\mathcal{N}(C,c)}} \\
&\text{if } y = y_i, \Big[\text{logit}(\boldsymbol{x}_i)\Big]_y = \langle m_{y_i}^T, \lambda_i(m_{y_i}^T - \bar{M}^\star)\rangle = \lambda_i m_{y_i}(m_{y_i}^T - \bar{M}^\star) = \lambda_i \mathcal{N}(C,c) = \rho\sqrt{\mathcal{N}(C,c)}
\end{aligned}$$

According to the above formula, we derive the values of M_i in **C1** and M in **C2**.

$$\begin{aligned}
M_i &= \lambda_i \mathcal{A}(C,c) = \rho\frac{\mathcal{A}(C,c)}{\sqrt{\mathcal{N}(C,c)}} \\
M &= \lambda_i(C-1)(\mathcal{A}(C,c) - \mathcal{N}(C,c)) = \rho(C-1)(\frac{\mathcal{A}(C,c)}{\sqrt{\mathcal{N}(C,c)}} - \sqrt{\mathcal{N}(C,c)})
\end{aligned}$$

□

C. The Proof of Thm.4.5

Lemma C.1. *Assume the map $\boldsymbol{w} \mapsto f(\boldsymbol{w})$ is nonnegative and L-smooth. Then for any $\boldsymbol{w}$, we have*

$$\|\partial_{\boldsymbol{w}} f(\boldsymbol{w})\| \leq \sqrt{2Lf(\boldsymbol{w})}$$

where $\partial_{\boldsymbol{w}} f(\boldsymbol{w})$ denote the sub-gradient of $\boldsymbol{w}$.

Restatement of Theorem 4.5. *Consider the two groups of parameters $(\boldsymbol{w}_t, T_t)$ and $(\boldsymbol{w}_t^{(i)}, T_t^{(i)})$ trained on S and $S^{(i)}$ from the same starting point by the update rule*

$$\begin{aligned}
\boldsymbol{w}_{t+1} &\leftarrow \boldsymbol{w}_t - \eta_t^w \partial_{\boldsymbol{w}_t} g(\boldsymbol{x}_{i_t}; \boldsymbol{w}_t; M_t) \\
T_{t+1} &\leftarrow T_t - \eta_t^T \partial_{T_t} g(\boldsymbol{x}_{i_t}; \boldsymbol{w}_t; T_t)
\end{aligned}$$

Assume

- *The function $g(\boldsymbol{x};\boldsymbol{w};T)$ is nonnegative for any $\boldsymbol{x}$, $\boldsymbol{w}$ and T;*
- *The function $T \mapsto g(\boldsymbol{x};\boldsymbol{w};T)$ is L_T-smooth for any $\boldsymbol{w}$ and $\boldsymbol{x}$;*
- *The function $\boldsymbol{w} \mapsto \partial_w g(\boldsymbol{x};\boldsymbol{w};T)$ is ℓ_w-lipschitz for any T and $\boldsymbol{x}$;*
- *The function $\boldsymbol{w} \mapsto g(\boldsymbol{x};\boldsymbol{w};T)$ is L_w-smooth for any T and $\boldsymbol{x}$;*
- *The function $T \mapsto \partial_T g(\boldsymbol{x};\boldsymbol{w};T)$ is ℓ_T-lipschitz for any $\boldsymbol{w}$ and $\boldsymbol{x}$.*

We denote

$$\boldsymbol{v} = \begin{bmatrix} \eta_t^T\sqrt{2L_T} \\ \eta_t^w\sqrt{2L_w} \end{bmatrix}, F = \begin{bmatrix} 1 + \frac{N-1}{N}\eta_t^T L_T & \frac{N-1}{N}\eta_t^T \ell_w \\ \frac{N-1}{N}\eta_t^w \ell_T & 1 + \frac{N-1}{N}\eta_t^w L_w \end{bmatrix}, G_S(\boldsymbol{w};T) = \frac{1}{N}\sum_{i=1}^N g(\boldsymbol{x}_i;\boldsymbol{w};T)$$

Then, if $\eta_t^T/\eta_t^w = \ell_T/\ell_w$ holds, the ℓ_1 model stabilities of $\boldsymbol{w}$ and T are given by

$$\begin{bmatrix} \mathbb{E}_{S,\tilde{S},A}\left[\frac{1}{N}\sum_{i=1}^N \|T_{t+1} - T_{t+1}^{(i)}\|\right] \\ \mathbb{E}_{S,\tilde{S},A}\left[\frac{1}{N}\sum_{i=1}^N \|\boldsymbol{w}_{t+1} - \boldsymbol{w}_{t+1}^{(i)}\|\right] \end{bmatrix} \le \frac{2}{N}\sum_{j=1}^{t}\left(\lambda_1^{t-j}\boldsymbol{p}_1\boldsymbol{p}_1^T\boldsymbol{v} + \lambda_2^{t-j}\boldsymbol{p}_2\boldsymbol{p}_2^T\boldsymbol{v}\right)\mathbb{E}_{S,A}\left[\sqrt{G_S(\boldsymbol{w}_j;T_j)}\right]$$

where λ_1, λ_2 and $\boldsymbol{p}_1, \boldsymbol{p}_2$ are eigen values and eigen vectors of F respectively.

Proof. First, we analyze $\|T_{t+1} - T_{t+1}^{(i)}\|$ and $\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_{t+1}^{(i)}\|$ separately. Consider $\|T_{t+1} - T_{t+1}^{(i)}\|$. If $i_t = i$, we have

$$\begin{aligned}
\|T_{t+1} - T_{t+1}^{(i)}\| \le & \|T_t - T_t^{(i)}\| + \eta_t^T\|\partial_{\boldsymbol{T}_t} g(\boldsymbol{x}_i;\boldsymbol{w}_t;T_t) - \partial_{\boldsymbol{T}_t^{(i)}} g(\tilde{x}_i;\boldsymbol{w}_t^{(i)};T_t^{(i)})\| \\
\le & \|T_t - T_t^{(i)}\| + \eta_t^T\|\partial_{\boldsymbol{T}_t} g(\boldsymbol{x}_i;\boldsymbol{w}_t;T_t)\| + \eta_t^T\|\partial_{\boldsymbol{T}_t^{(i)}} g(\tilde{x}_i;\boldsymbol{w}_t^{(i)};T_t^{(i)})\| \\
\le & \|T_t - T_t^{(i)}\| + \sqrt{2L_T}\eta_t^T\left(\sqrt{g(\boldsymbol{x}_i;\boldsymbol{w}_t;T_t)} + \sqrt{g(\tilde{x}_i;\boldsymbol{w}_t^{(i)};T_t^{(i)})}\right)
\end{aligned}$$

where the last inequality follows from Lem.C.1. If $i_t \neq i$, we have

$$\begin{aligned}
\|T_{t+1} - T_{t+1}^{(i)}\| \le & \|T_t - T_t^{(i)}\| + \eta_t^T\|\partial_{\boldsymbol{T}_t} g(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;T_t) - \partial_{\boldsymbol{T}_t^{(i)}} g(\boldsymbol{x}_{i_t};\boldsymbol{w}_t^{(i)};T_t^{(i)})\| \\
\le & \|T_t - T_t^{(i)}\| + \eta_t^T\|\partial_{\boldsymbol{T}_t} g(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;T_t) - \partial_{\boldsymbol{T}_t} g(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;T_t^{(i)})\| + \\
& \eta_t^T\|\partial_{\boldsymbol{T}_t} g(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;T_t^{(i)}) - \partial_{\boldsymbol{T}_t^{(i)}} g(\boldsymbol{x}_{i_t};\boldsymbol{w}_t^{(i)};T_t^{(i)})\| \\
\le & \|T_t - T_t^{(i)}\| + \eta_t^T L_T\|T_t - T_t^{(i)}\| + \eta_t^T\ell_w\|\boldsymbol{w}_t - \boldsymbol{w}_t^{(i)}\| \\
= & (1 + \eta_t^T L_T)\|T_t - T_t^{(i)}\| + \eta_t^T\ell_w\|\boldsymbol{w}_t - \boldsymbol{w}_t^{(i)}\|
\end{aligned}$$

Recall the update rule 4.4 uniformly samples a data from S and $S^{(i)}$ for every iteration. The probability of selecting the i-th data is $\frac{1}{N}$. Combining two cases with expectations, we have

$$\begin{aligned}
\mathbb{E}_{S,S^{(i)},A}\left[\|T_{t+1} - T_{t+1}^{(i)}\|\right] \le & (1 + \frac{N-1}{N}\eta_t^T L_T)\mathbb{E}_{S,S^{(i)},A}\|T_t - T_t^{(i)}\| + \frac{N-1}{N}\eta_t^T\ell_w\mathbb{E}_{S,S^{(i)},A}\|\boldsymbol{w}_t - \boldsymbol{w}_t^{(i)}\| \\
& + \frac{1}{N}\sqrt{2L_T}\eta_t^T\mathbb{E}_{S,S^{(i)},A}\left(\sqrt{g(\boldsymbol{x}_i;\boldsymbol{w}_t;T_t)} + \sqrt{g(\tilde{x}_i;\boldsymbol{w}_t^{(i)};T_t^{(i)})}\right)
\end{aligned} \tag{5}$$

Note that S and $S^{(i)}$ follow from the same distribution, So

$$\begin{aligned}
\mathbb{E}_{S,S^{(i)},A}\left[\|T_{t+1} - T_{t+1}^{(i)}\|\right] \le & (1 + \frac{N-1}{N}\eta_t^T L_T)\mathbb{E}_{S,S^{(i)},A}\|T_t - T_t^{(i)}\| + \frac{N-1}{N}\eta_t^T\ell_w\mathbb{E}_{S,S^{(i)},A}\|\boldsymbol{w}_t - \boldsymbol{w}_t^{(i)}\| \\
& + \frac{2}{N}\sqrt{2L_T}\eta_t^T\mathbb{E}_{S,S^{(i)},A}\sqrt{g(\boldsymbol{x}_i;\boldsymbol{w}_t;T_t)}
\end{aligned} \tag{6}$$

According to the symmetry of two arguments, we know

$$\begin{aligned}
\mathbb{E}_{S,S^{(i)},A}\left[\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_{t+1}^{(i)}\|\right] \le & (1 + \frac{N-1}{N}\eta_t^w L_w)\mathbb{E}_{S,S^{(i)},A}\|\boldsymbol{w}_t - \boldsymbol{w}_t^{(i)}\| + \frac{N-1}{N}\eta_t^w\ell_T\mathbb{E}_{S,S^{(i)},A}\|T_t - T_t^{(i)}\| \\
& + \frac{2}{N}\sqrt{2L_w}\eta_t^w\mathbb{E}_{S,S^{(i)},A}\sqrt{g(\boldsymbol{x}_i;\boldsymbol{w}_t;T_t)}
\end{aligned} \tag{7}$$

We combine (6) and (7) to derive

$$\begin{aligned}
& \begin{bmatrix} \mathbb{E}_{S,S^{(i)},A}\|T_{t+1} - T_{t+1}^{(i)}\| \\ \mathbb{E}_{S,S^{(i)},A}\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_{t+1}^{(i)}\| \end{bmatrix} \\
\le & \overbrace{\begin{bmatrix} 1 + \frac{N-1}{N}\eta_t^T L_T & \frac{N-1}{N}\eta_t^T\ell_w \\ \frac{N-1}{N}\eta_t^w\ell_T & 1 + \frac{N-1}{N}\eta_t^w L_w \end{bmatrix}}^{F}\begin{bmatrix} \mathbb{E}_{S,S^{(i)},A}\|T_t - T_t^{(i)}\| \\ \mathbb{E}_{S,S^{(i)},A}\|\boldsymbol{w}_t - \boldsymbol{w}_t^{(i)}\| \end{bmatrix} + \overbrace{\begin{bmatrix} \eta_t^T\sqrt{2L_T} \\ \eta_t^w\sqrt{2L_w} \end{bmatrix}}^{\boldsymbol{v}}\frac{2}{N}\mathbb{E}_{S,S^{(i)},A}\left[\sqrt{g(\boldsymbol{x}_i;\boldsymbol{w}_t;T_t)}\right]
\end{aligned} \tag{8}$$

Recall that T_t and $T_t^{(i)}$ is equal when $t=1$, solve the recursion to obtain

$$\begin{bmatrix} \mathbb{E}_{S,S^{(i)},A}\|T_{t+1}-T_{t+1}^{(i)}\| \\ \mathbb{E}_{S,S^{(i)},A}\|\boldsymbol{w}_{t+1}-\boldsymbol{w}_{t+1}^{(i)}\| \end{bmatrix} \leq \frac{2}{N}\sum_{j=1}^{t} F^{t-j}\boldsymbol{v}\mathbb{E}_{S,S^{(i)},A}\left[\sqrt{g(\boldsymbol{x}_i;\boldsymbol{w}_j;T_j)}\right] \tag{9}$$

To make F a symmetric matrix, let $\eta_t^T/\eta_t^w = \ell_T/\ell_w$. Then for F, we have the following unitary decomposition

$$\begin{bmatrix} 1+\frac{N-1}{N}\eta_t^T L_T & \frac{N-1}{N}\eta_t^T \ell_w \\ \frac{N-1}{N}\eta_t^w \ell_T & 1+\frac{N-1}{N}\eta_t^w L_w \end{bmatrix} = [\boldsymbol{p}_1,\boldsymbol{p}_2]\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}\begin{bmatrix} \boldsymbol{p}_1^T \\ \boldsymbol{p}_2^T \end{bmatrix} \tag{10}$$

where λ_1,λ_2 and $\boldsymbol{p}_1,\boldsymbol{p}_2$ are eigen values and eigen vectors of F respectively. Then We bring (10) into (9) to yield

$$\begin{aligned} \begin{bmatrix} \mathbb{E}_{S,S^{(i)},A}\|T_{t+1}-T_{t+1}^{(i)}\| \\ \mathbb{E}_{S,S^{(i)},A}\|\boldsymbol{w}_{t+1}-\boldsymbol{w}_{t+1}^{(i)}\| \end{bmatrix} \leq & \frac{2}{N}\sum_{j=1}^{t} F^{t-j}\boldsymbol{v}\mathbb{E}_{S,S^{(i)},A}\left[\sqrt{g(\boldsymbol{x}_i;\boldsymbol{w}_j;T_j)}\right] \\ = & \frac{2}{N}\sum_{j=1}^{t}[\boldsymbol{p}_1,\boldsymbol{p}_2]\begin{bmatrix} \lambda_1^{t-j} & 0 \\ 0 & \lambda_2^{t-j} \end{bmatrix}\begin{bmatrix} \boldsymbol{p}_1^T\boldsymbol{v} \\ \boldsymbol{p}_2^T\boldsymbol{v} \end{bmatrix}\mathbb{E}_{S,S^{(i)},A}\left[\sqrt{g(\boldsymbol{x}_i;\boldsymbol{w}_j;T_j)}\right] \\ = & \frac{2}{N}\sum_{j=1}^{t}\left(\lambda_1^{t-j}\boldsymbol{p}_1\boldsymbol{p}_1^T\boldsymbol{v}+\lambda_2^{t-j}\boldsymbol{p}_2\boldsymbol{p}_2^T\boldsymbol{v}\right)\mathbb{E}_{S,S^{(i)},A}\left[\sqrt{g(\boldsymbol{x}_i;\boldsymbol{w}_j;T_j)}\right] \end{aligned}$$

The final conclusion follows from the sum of $i=1,\ldots,N$ and the Jensen inequality. □

D. The Proof of Thm.4.7

Lemma D.1. *For any two square matrix X and Y with the same dimension, we have*

$$\|e^{X+Y}-e^{X}\| \leq \|Y\|e^{\|Y\|}e^{\|X\|}$$

where $\|\cdot\|$ indicates any matrix norm.

Before we begin the proof of theorem, we make a necessary assumption as follow.

Assumption D.2. Consider the optimization of Def.4.6, assume $T_t \in \Omega$ for any t, where $\Omega \subset \mathfrak{so}(n)$ is the auxiliary parameter space. Suppose Ω is a bounded set, satisfying

$$\max\left(\max_{\substack{T_t\in\Omega \\ B_t\in\{B_t\in\mathbb{R}^{n\times n}\mid T=B_t-B_t^T\}}}\frac{\phi(\|\partial_{B_t}f(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;R_t)-\partial_{B_t}(f(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;R_t))^T\|)}{\|\partial_{B_t}f(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;R_t)-\partial_{B_t}(f(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;R_t))^T\|},\ \max_{\substack{T_t\in\Omega \\ T_t^{(i)}\in\Omega}}\frac{\phi(\|T_t-T_t^{(i)}\|)}{\|T_t-T_t^{(i)}\|}\right)=L_\phi,$$

for any $\boldsymbol{x}_{i_t}$ and $\boldsymbol{w}_t$, where $\phi(a)$ is the mapping $a\mapsto ae^a$. And we denote $\max_{T\in\Omega}e^{\|T\|}$ as $\mathcal{H}(\Omega)$.

Then, formally, we propose our stability analysis for *Representation-Balanced Learning*.

Lemma D.3. *Consider the two groups of parameters $(\boldsymbol{w}_t,R_t)$ and $(\boldsymbol{w}_t^{(i)},R_t^{(i)})$ trained on S and $S^{(i)}$ from the same starting point by the update rule*

$$\begin{aligned} \boldsymbol{w}_{t+1} &\leftarrow \boldsymbol{w}_t-\eta_t^w\partial_{\boldsymbol{w}_t}f(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;R_t) \\ B_{t+1} &\leftarrow B_t-\eta_t^T\partial_{B_t}f(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;R_t) \\ T_{t+1} &\leftarrow B_{t+1}-B_{t+1}^T \\ R_{t+1} &\leftarrow \exp(T_{t+1}) \end{aligned}$$

Assume

- *Assumption D.2 holds.*
- *The function $T\mapsto f(\boldsymbol{x};\boldsymbol{w};e^T)$ is nonnegative and L_T-smooth for any $\boldsymbol{x}$ and $\boldsymbol{w}$.*

We have

$$\mathop{\mathbb{E}}_{S,\tilde{S},A}\left[\frac{1}{N}\sum_{i=1}^{N}\|R_{t+1}-R_{t+1}^{(i)}\|\right] \leq \mathcal{H}(\Omega)L_\phi(N-1)\sum_{j=1}^{t}N^{j-t-1}\mathop{\mathbb{E}}_{S,\tilde{S},A}\Big[\frac{1}{N}\sum_{i=1}^{N}\|T_{j+1}-T_{j+1}^{(i)}\|\Big]+$$
$$2\sqrt{2L_T}\mathcal{H}(\Omega)L_\phi\sum_{j=1}^{t}N^{j-t-1}\eta_j^T\mathop{\mathbb{E}}_{S,A}\Big[\sqrt{\frac{1}{N}\sum_{i=1}^{N}f(\tilde{\boldsymbol{x}}_i;\boldsymbol{w}_j;R_j)}\Big]$$

Proof. First, we consider $\|R_{t+1}-R_{t+1}^{(i)}\|$. When $i_t=i$, we have

$$\begin{aligned}\|R_{t+1}-R_{t+1}^{(i)}\| &= \|R_{t+1}-R_t+R_t-R_t^{(i)}+R_t^{(i)}-R_{t+1}^{(i)}\| \\ &\leq \|R_t-R_t^{(i)}\|+\|R_{t+1}-R_t\|+\|R_t^{(i)}-R_{t+1}^{(i)}\|\end{aligned}$$

Then we turn to $\|R_{t+1}-R_t\|$,

$$\begin{aligned}\|R_{t+1}-R_t\| &= \|e^{T_{t+1}}-e^{T_t}\| \\ &= \|\exp\big(B_t-B_t^T-\eta_t^T\partial_{B_t}f(\boldsymbol{x}_i;\boldsymbol{w}_t;R_t)+\eta_t^T(\partial_{B_t}f(\boldsymbol{x}_i;\boldsymbol{w}_t;R_t))^T\big)-e^{B_t-B_t^T}\| \\ &\leq \|\eta_t^T(\partial_{B_t}f(\boldsymbol{x}_i;\boldsymbol{w}_t;R_t))-\eta_t^T(\partial_{B_t}f(\boldsymbol{x}_i;\boldsymbol{w}_t;R_t))^T\| \\ &\qquad \exp\big[\|\eta_t^T(\partial_{B_t}f(\boldsymbol{x}_i;\boldsymbol{w}_t;R_t))-\eta_t^T(\partial_{B_t}f(\boldsymbol{x}_i;\boldsymbol{w}_t;R_t))^T\|\big]\exp\big[\|B_t-B_t^T\|\big]\end{aligned}$$

where the inequality is due to Lem.D.1. Then, we denote $\max_{T\in\Omega}e^{\|T\|}$ as $\mathcal{H}(\Omega)$ and $\phi(a)$ as the map $a\to ae^{2a}$

$$\|R_{t+1}-R_t\|^2 \leq \phi\Big(\eta_t^T\|\partial_{B_t}f(\boldsymbol{x}_i;\boldsymbol{w}_t;R_t)-(\partial_{B_t}f(\boldsymbol{x}_i;\boldsymbol{w}_t;R_t))^T\|\Big)\mathcal{H}(\Omega) \leq \mathcal{H}(\Omega)L_\phi\eta_t^T\|\partial_{T_t}f(\boldsymbol{x}_i;\boldsymbol{w}_t;R_t)\|$$

The above inequality follows from the bounded parameter space Ω in Assumption.D.2:

$$\max_{B_t\in\{B_t\in\mathbb{R}^{d\times d}|T=B_t-B_t^T\}}\frac{\phi(\|\partial_{B_t}f(\boldsymbol{x}_i;\boldsymbol{w}_t;R_t)-\partial_{B_t}(f(\boldsymbol{x}_i;\boldsymbol{w}_t;R_t))^T\|)}{\|\partial_{B_t}f(\boldsymbol{x}_i;\boldsymbol{w}_t;R_t)-\partial_{B_t}(f(\boldsymbol{x}_i;\boldsymbol{w}_t;R_t))^T\|}\leq L_\phi,\ \text{for any } T_t\in\Omega$$

And $\|R_t^{(i)}-R_{t+1}^{(i)}\|$ is similar, we have

$$\|R_{t+1}^{(i)}-R_t^{(i)}\| \leq \mathcal{H}(\Omega)L_\phi\eta_t^T\|\partial_{T_t^{(i)}}f(\tilde{\boldsymbol{x}}_i;\boldsymbol{w}_t^{(i)};R_t^{(i)})\|$$

Next, we consider the case that $i_t\neq i$, we have

$$\begin{aligned}\|R_{t+1}-R_{t+1}^{(i)}\| &= \|\exp(T_{t+1})-\exp(T_{t+1}^{(i)})\| \\ &= \|\exp(T_{t+1}-T_{t+1}^{(i)}+T_{t+1}^{(i)})-\exp(T_{t+1}^{(i)})\| \\ &\leq \|T_{t+1}-T_{t+1}^{(i)}\|\exp\Big[\|T_{t+1}-T_{t+1}^{(i)}\Big]\exp\Big[\|T_{t+1}^{(i)}\|\Big] \\ &\leq \phi\Big(\|T_{t+1}-T_{t+1}^{(i)}\|\Big)\mathcal{H}(\Omega) \\ &\leq \mathcal{H}(\Omega)L_\phi\|T_{t+1}-T_{t+1}^{(i)}\|\end{aligned} \tag{11}$$

where the last inequality is due to the assumption that $\max_{T_t,T_t^{(i)}\in\Omega}\frac{\phi(\|T_t-T_t^{(i)}\|)}{\|T_t-T_t^{(i)}\|}\leq L_\phi$. We combine two cases with expectations,

$$\begin{aligned}
\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\|R_{t+1}-R^{(i)}_{t+1}\|\right] \le{}& \frac{1}{N}\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\|R_t-R^{(i)}_t\|\right]+\frac{N-1}{N}\mathcal{H}(\Omega)L_\phi\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\|T_{t+1}-T^{(i)}_{t+1}\|\right]+\\
&\frac{\mathcal{H}(\Omega)L_\phi\eta^T_t}{N}\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\|\partial_{T_t}f(\boldsymbol{\tilde{x}}_i;\boldsymbol{w}_t;R_t)\|+\|\partial_{T^{(i)}_t}f(\boldsymbol{\tilde{x}}_i;\boldsymbol{w}^{(i)}_t;R^{(i)}_t)\|\right]\\
={}&\frac{1}{N}\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\|R_t-R^{(i)}_t\|\right]+\frac{N-1}{N}\mathcal{H}(\Omega)L_\phi\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\|T_{t+1}-T^{(i)}_{t+1}\|\right]+\\
&\frac{2\mathcal{H}(\Omega)L_\phi\eta^T_t}{N}\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\|\partial_{T_t}f(\boldsymbol{\tilde{x}}_i;\boldsymbol{w}_t;R_t)\|\right]\\
\le{}&\frac{1}{N}\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\|R_t-R^{(i)}_t\|\right]+\frac{N-1}{N}\mathcal{H}(\Omega)L_\phi\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\|T_{t+1}-T^{(i)}_{t+1}\|\right]+\\
&\frac{2\sqrt{2L_T}\mathcal{H}(\Omega)L_\phi\eta^T_t}{N}\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\sqrt{f(\boldsymbol{\tilde{x}}_i;\boldsymbol{w}_t;R_t)}\right]
\end{aligned}$$

where the last inequality follows from Lem.C.1. To solve the recursion, we multily N^{t+1} on both sides

$$\begin{aligned}
N^{t+1}\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\|R_{t+1}-R^{(i)}_{t+1}\|\right] \le{}& N^t\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\|R_t-R^{(i)}_t\|\right]+N^t(N-1)\mathcal{H}(\Omega)L_\phi\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\|T_{t+1}-T^{(i)}_{t+1}\|\right]+\\
&N^t 2\sqrt{2L_T}\mathcal{H}(\Omega)L_\phi\eta^T_t\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\sqrt{f(\boldsymbol{\tilde{x}}_i;\boldsymbol{w}_t;R_t)}\right]
\end{aligned}$$

And then, take a summation for t from 1 to t,

$$\begin{aligned}
\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\|R_{t+1}-R^{(i)}_{t+1}\|\right] \le{}& \mathcal{H}(\Omega)L_\phi(N-1)\sum_{j=1}^{t}N^{j-t-1}\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\|T_{j+1}-T^{(i)}_{j+1}\|\right]+\\
&2\sqrt{2L_T}\mathcal{H}(\Omega)L_\phi\sum_{j=1}^{t}N^{j-t-1}\eta^T_j\mathop{\mathbb{E}}_{S,S^{(i)},A}\left[\sqrt{f(\boldsymbol{\tilde{x}}_i;\boldsymbol{w}_j;R_j)}\right]
\end{aligned}$$

Then we take a summation for $i=1,\ldots,N$ to derive the on-average model stability,

$$\begin{aligned}
\mathop{\mathbb{E}}_{S,\tilde{S},A}\left[\frac{1}{N}\sum_{i=1}^{N}\|R_{t+1}-R^{(i)}_{t+1}\|\right] \le{}& \mathcal{H}(\Omega)L_\phi(N-1)\sum_{j=1}^{t}N^{j-t-1}\mathop{\mathbb{E}}_{S,\tilde{S},A}\left[\frac{1}{N}\sum_{i=1}^{N}\|T_{j+1}-T^{(i)}_{j+1}\|\right]+\\
&2\sqrt{2L_T}\mathcal{H}(\Omega)L_\phi\sum_{j=1}^{t}N^{j-t-1}\eta^T_j\mathop{\mathbb{E}}_{S,\tilde{S},A}\left[\frac{1}{N}\sum_{i=1}^{N}\sqrt{f(\boldsymbol{\tilde{x}}_i;\boldsymbol{w}_j;R_j)}\right]\\
\le{}& \mathcal{H}(\Omega)L_\phi(N-1)\sum_{j=1}^{t}N^{j-t-1}\mathop{\mathbb{E}}_{S,\tilde{S},A}\left[\frac{1}{N}\sum_{i=1}^{N}\|T_{j+1}-T^{(i)}_{j+1}\|\right]+\\
&2\sqrt{2L_T}\mathcal{H}(\Omega)L_\phi\sum_{j=1}^{t}N^{j-t-1}\eta^T_j\mathop{\mathbb{E}}_{S,\tilde{S},A}\left[\sqrt{\frac{1}{N}\sum_{i=1}^{N}f(\boldsymbol{\tilde{x}}_i;\boldsymbol{w}_j;R_j)}\right]
\end{aligned} \tag{12}$$

The last inequality follows from Jensen inequality. □

Through the Lem.D.3, we give the model stability of orthogonal matrix R in our framework.

Restatement of Theorem 4.7. *Consider the two groups of parameters $(\boldsymbol{w}_t, R_t)$ and $(\boldsymbol{w}^{(i)}_t, R^{(i)}_t)$ trained on S and $S^{(i)}$ from the same starting point by update rule*

$$\begin{aligned}
\boldsymbol{w}_{t+1} &\leftarrow \boldsymbol{w}_t-\eta^w_t\partial_{\boldsymbol{w}_t}f(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;R_t)\\
B_{t+1} &\leftarrow B_t-\eta^T_t\partial_{B_t}f(\boldsymbol{x}_{i_t};\boldsymbol{w}_t;R_t)\\
T_{t+1} &\leftarrow B_{t+1}-B^T_{t+1}\\
R_{t+1} &\leftarrow \exp(T_{t+1})
\end{aligned}$$

Then let $\boldsymbol{w} \to f(\boldsymbol{x}; \boldsymbol{w}; e^T)$ and $T \to f(\boldsymbol{x}; \boldsymbol{w}; e^T)$ in Lem.D.3 be $\boldsymbol{w} \to g(\boldsymbol{x}; \boldsymbol{w}; T)$ and $T \to g(\boldsymbol{x}; \boldsymbol{w}; T)$ in Thm.4.5. Assume Assumption.D.2 and all assumptions in Thm.4.5 hold. Denote $\mathbf{p}(k, j) = \lambda_1^{k-j} \boldsymbol{p}_1 \boldsymbol{p}_1^T \boldsymbol{v} + \lambda_2^{k-j} \boldsymbol{p}_2 \boldsymbol{p}_2^T \boldsymbol{v}$. Then the ℓ_1 model stability of parameter R is given by

$$\mathop{\mathbb{E}}_{S,\tilde{S},A}\left[\frac{1}{N}\sum_{i=1}^{N}\|R_{t+1} - R_{t+1}^{(i)}\|\right] \leq \frac{2\mathcal{H}(\Omega)L_\phi}{N}\sum_{j=1}^{t}\big(h_1(j) + h_2(j)\big)\mathbb{E}_{S,A}\left[\sqrt{F_S(\boldsymbol{w}_j, R_j)}\right]$$

where $h_1(j) = (N-1)\sum_{k=1}^{t-j+1} N^{k-t-1}\boldsymbol{p}(k, j)_1$ and $h_2(j) = \sqrt{2L_T}N^{j-t}\eta_j$.

Proof. The main idea of this theorem is to plug Thm.4.5 into Lem.D.3. Consider the unbounded $\mathbb{E}_{A,S,\tilde{S}}\left[\|T_{j+1} - T_{j+1}^{(i)}\|\right]$, accoring to the Thm.4.5, we know,

$$\begin{aligned}
&\sum_{j=1}^{t} N^{j-t-1}\mathop{\mathbb{E}}_{S,\tilde{S},A}\Big[\frac{1}{N}\sum_{i=1}^{N}\|T_{j+1} - T_{j+1}^{(i)}\|\Big] \\
\leq & 2\sum_{j=1}^{t} N^{j-t-2}\sum_{k=1}^{j}\left[\lambda_1^{j-k}\boldsymbol{p}_1\boldsymbol{p}_1^T\boldsymbol{v} + \lambda_2^{j-k}\boldsymbol{p}_2\boldsymbol{p}_2^T\boldsymbol{v}\right]_1 \mathbb{E}_{S,A}\left[\sqrt{\frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}_i; \boldsymbol{w}_k; R_k)}\right]
\end{aligned} \tag{13}$$

Combine the conclusion of Thm.D.3 with (13) to derive the result.

$$\begin{aligned}
\mathop{\mathbb{E}}_{S,\tilde{S},A}\left[\frac{1}{N}\sum_{i=1}^{N}\|R_{t+1} - R_{t+1}^{(i)}\|\right] \leq\ & 2\sqrt{2L_T}\mathcal{H}(\Omega)L_\phi\sum_{j=1}^{t} N^{j-t-1}\eta_j\mathop{\mathbb{E}}_{S,A}\Big[\sqrt{\frac{1}{N}\sum_{i=1}^{N} f(\tilde{\boldsymbol{x}}_i; \boldsymbol{w}_j; R_j)}\Big] + \\
& 2\mathcal{H}(\Omega)L_\phi(N-1)\sum_{j=1}^{t} N^{j-t-2}\sum_{k=1}^{j}\left[\lambda_1^{j-k}\boldsymbol{p}_1\boldsymbol{p}_1^T\boldsymbol{v} + \lambda_2^{j-k}\boldsymbol{p}_2\boldsymbol{p}_2^T\boldsymbol{v}\right]_1 \mathbb{E}_{S,A}\left[\sqrt{\frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}_i; \boldsymbol{w}_k; R_k)}\right] \\
=& 2\sqrt{2L_T}\mathcal{H}(\Omega)L_\phi\sum_{j=1}^{t} N^{j-t-1}\eta_j\mathop{\mathbb{E}}_{S,A}\Big[\sqrt{\frac{1}{N}\sum_{i=1}^{N} f(\tilde{\boldsymbol{x}}_i; \boldsymbol{w}_j; R_j)}\Big] + \\
& 2\mathcal{H}(\Omega)L_\phi(N-1)\sum_{j=1}^{t}\sum_{k=1}^{t-j+1} N^{k-t-2}\left[\lambda_1^{k-j}\boldsymbol{p}_1\boldsymbol{p}_1^T\boldsymbol{v} + \lambda_2^{k-j}\boldsymbol{p}_2\boldsymbol{p}_2^T\boldsymbol{v}\right]_1 \mathbb{E}_{S,A}\left[\sqrt{\frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}_i; \boldsymbol{w}_j; R_j)}\right]
\end{aligned}$$

We label $h_1(j) = (N-1)\sum_{k=1}^{t-j+1} N^{k-t-1}\boldsymbol{p}(k, j)_1$ and $h_2(j) = \sqrt{2L_T}N^{j-t}\eta_j$. By merging coefficient, we derive

$$\mathop{\mathbb{E}}_{S,\tilde{S},A}\left[\frac{1}{N}\sum_{i=1}^{N}\|R_{t+1} - R_{t+1}^{(i)}\|\right] \leq \frac{2\mathcal{H}(\Omega)L_\phi}{N}\sum_{j=1}^{t}\big(h_1(j) + h_2(j)\big)\mathbb{E}_{S,A}\left[\sqrt{F_S(\boldsymbol{w}_j, R_j)}\right]$$

□

E. PyTorch implementation of Our Method

To show all details of our method, we release the source code of our framework implemented by Pytorch. Our approach is very simple in implementation, only need more than 20 lines code. Here, we provide two versions of implementation, as shown in Code.1 and 2. The first is implemented by the `geotorch` library (Lezcano-Casado, 2019), which could perform optimization on manifolds easily. Another version is implemented without using other third-party libraries. Both versions are valid to reproduce experimental results. We recommend using the first since the former is more concise.

F. Details of Fig.2

In Fig.2, we design a toy experiment to simulate feature learning in classification, where the classification problem contains 3 categories. In the experiment of *NeurCol Phenomenon*, we set every class to have 30 samples. And in the experiment of

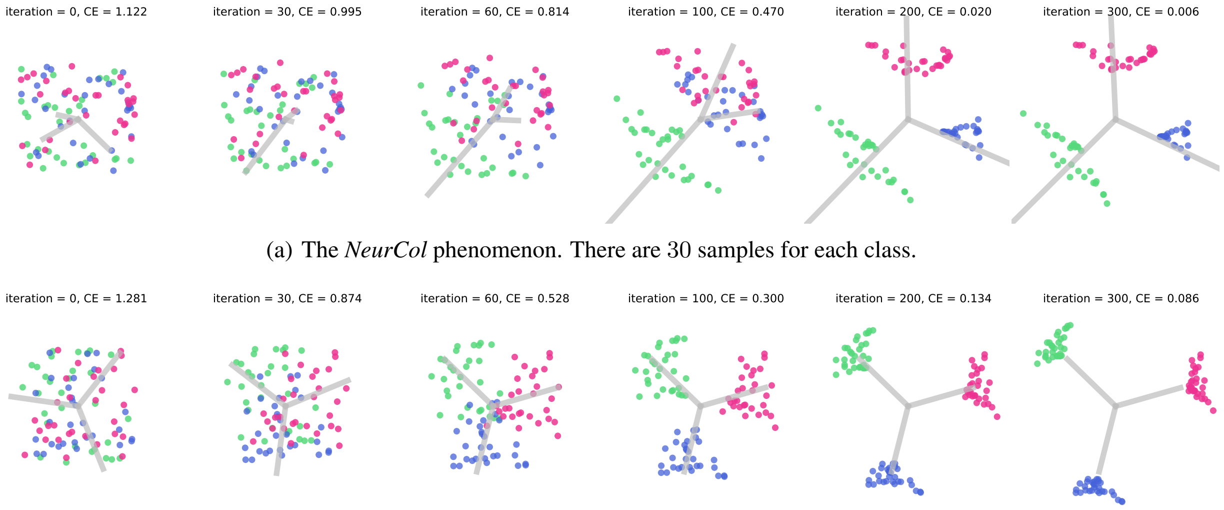
```python
class RBL(nn.Module):
    r"""
    Args:
        backbone (nn.Module) : deep model for feature
        feature_num (int) : backbone's feature dimension
        class_num (int) : the number of classes
        _cls_num_list (list) : numbers of sample in each class

    Examples:
        >>> import torchvision.models as models
        >>> feature_dim = 512
        >>> class_dim = 1000
        >>> resnet18 = models.resnet18(num_classes=feature_dim).cuda()
        >>> model = RBL(resnet18, feature_dim, class_dim, torch.arange(class_dim, 0, -1)).cuda()
        >>> pred = model(torch.randn(1, 3, 224, 224).cuda())
        >>> print(pred.shape)
        torch.Size([1, 1000])
    """
    def __init__(self, backbone, feature_num, class_num, _cls_num_list):
        super(RBL, self).__init__()
        self.feature_num = feature_num
        self.class_num = class_num
        self.backbone = backbone
        self.margin = torch.log(torch.Tensor(_cls_num_list) / sum(_cls_num_list)).cuda()

        if feature_num < class_num:
            self.rotate = nn.Linear(class_num, feature_num, bias=False)
            self.register_buffer("ETF", self.generate_ETF(dim=class_num))
        else:
            self.rotate = nn.Linear(feature_num, feature_num, bias=False)
            self.register_buffer("ETF", \
                self.generate_ETF(dim=feature_num)[:, :self.class_num])
        geotorch.orthogonal(self.rotate, "weight")

    def generate_ETF(self, dim):
        return torch.eye(dim, dim) - torch.ones(dim, dim) / dim

    def forward(self, x):
        logit = self.backbone(x) @ self.rotate.weight @ self.ETF
        return logit if self.training else logit - self.margin
```

Code 1: PyTorch implementation of our framework using `geotorch` library (Lezcano-Casado, 2019).

```python
class PLPostHocModel(nn.Module):
def __init__(self, backbone, triv, feature_num, class_num, _cls_num_list):
    super(PLPostHocModel, self).__init__()
    self.feature_num = feature_num
    self.class_num = class_num
    self.backbone = backbone
    _cls_num_list = torch.Tensor(_cls_num_list)
    self.margin = torch.log(_cls_num_list / torch.sum(_cls_num_list)).cuda()

    if feature_num < class_num:
        self.register_buffer("ETF", self.generate_ETF(dim=class_num))
        self.rotate = nn.Linear(class_num, class_num, bias=False)
    else:
        self.register_buffer("ETF", self.generate_ETF(dim=feature_num)\
            [:, :self.class_num])
        self.rotate = nn.Linear(feature_num, feature_num, bias=False)

def generate_ETF(self, dim):
    return torch.eye(dim, dim) - torch.ones(dim, dim) / dim

def encode_rotate(self):
    if self.feature_num < self.class_num:
        return torch.linalg.matrix_exp(self.rotate.weight - self.rotate.weight.T)\
            [:self.feature_num, :]
    return torch.linalg.matrix_exp(self.rotate.weight - self.rotate.weight.T)

def forward(self, x):
    logit = self.backbone(x) @ self.encode_rotate() @ self.ETF
    return logit if self.training else logit - self.margin

def forward_feature(self, x):
    return self.backbone(x)
```

Code 2: PyTorch implementation of our framework without third-party libraries.

our framework, we set three classes has $100, 10, 1$ samples respectively. Both experiments follow the common paradigm of classification, using cross entropy loss and SGD optimization. To simulate the deep feature extractor, we let every sample also could be optimized. Besides, when performing optimization updates, we projection the sample feature and classifier's weight into a sphere to constrain their maximum norm. This constraint for tne maximal norm is needed, which makes *NeurCol* phenomenon more obvious. As shown in Fig.5 (a), if we eliminate this constraint, the feature may not achieve complete *NeurCol* phenomenon. Before it reach the symmetry structrues, the excessively large norm makes the loss fast converge, and feature learning stops due to the vanishing gradient. Besides, one can find that in Fig.5 (b), our framework still could learn relatively balanced features. We attribute this to the fixed norm of classifier weights in our framework. In the situation that the norm of classifier weights is fixed, our framework needs to reduce losses through ”pushing” sample features as far as possible in the direction of corresponding classifier weight.



(a) The *NeurCol* phenomenon. There are 30 samples for each class.

(b) The feature learning of our framework. Each class has $100, 10, 1$ samples respectively.

Figure 5: Experiments without constraint of maximal norm.