# SUPPLEMENTARY MATERIAL FOR REVISIT FINETUNING STRATEGY FOR FEW-SHOT LEARNING TO STRENGTHEN THE EQUIVARIANCE OF EMDEDDINGS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

In this supplementary material, 1) we present the gradients of Firth Bias Reduction loss to analyze its effectiveness; 2) the analytic formulation of LP-FT-FB's gradients is presented for understanding; 3) the whole flow of the proposed LP-FT-FB is given; 4) the ablation study is given to show the effectiveness of the proposed i-FBR; 5) the related meta-learning-based works are given.

## A THE DERIVATION OF FIRTH BIAS REDUCTION LOSS

The derivative of FBR with respect to $r_i^k$ is given. Here $r_i^k = \beta_k \cdot \mathbf{B}(\mathbf{x_i}, \theta)$. $i$ is the index of sample number ($1 \leq i \leq M$) and $k$ is the index of class number ($1 \leq k \leq C$).

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial r_i^k} &= \frac{\partial \mathcal{L}_{CE}}{\partial r_i^k} + \frac{\partial \mathcal{L}_{Firth}}{\partial r_i^k} \\ &= \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{C}\left[(p_i^k - y_i^k) + \frac{\partial \mathcal{L}_{Firth}}{\partial \mathbf{P}_i}\cdot\frac{\partial \mathbf{P}_i}{\partial r_i^k}\right],\end{aligned} \tag{1}$$

where $\frac{\partial \mathcal{L}_{CE}}{\partial r_i^k}$ is the typical gradient of the cross entropy loss and $y_i^k$ is $k_{th}$ element of $i_{th}$ sample label $\mathbf{y}_i$ (defined in Section 3.1). $\mathbf{P} = \{\mathbf{P}_i\}_{i=1}^M$ and $\mathbf{P}_i = \{p_i^k\}_{k=1}^C$. Then we give the loss of $i_{th}$ sample and its gradients with respect to $r_i^k$. Firstly, the analytic formula of Firth loss of $i_{th}$ sample is

$$\begin{aligned}\mathcal{L}_{Firth} &= -\frac{1}{M}\sum_{i=1}^{M}\left[\lambda \cdot D_{KL}(\mathbf{U}_{[0,C]} \| \mathbf{P}_i)\right] \\ &= -\frac{1}{M}\sum_{i=1}^{M}\left[\frac{\lambda}{(C+1)}\cdot(\log\frac{1}{C+1} + \log \mathbf{P}_i)\right]\end{aligned}. \tag{2}$$

Secondly, we have $p_i^k = \frac{e^{r_i^k}}{1+\sum_{j=1}^{C} e^{r_i^j}}$. The gradient of logistic regression is

$$\frac{p_i^k}{r_i^j} = \begin{cases} -p_i^k \cdot p_i^j, & k \neq j \\ p_i^k - p_i^k \cdot p_i^j, & k = j \end{cases}. \tag{3}$$

With Eq. 3, the gradients of $\mathcal{L}_{Firth}$ with respect to $r_i^k$ is derivated as follows.

$$\begin{aligned}
\frac{\partial \mathcal{L}_{Firth}}{\partial \mathbf{P}_i} \cdot \frac{\partial \mathbf{P}_i}{\partial r_i^k} &= \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{C} -\frac{\lambda}{(C+1)} \cdot \frac{\partial(\log \mathbf{P}_i)}{\partial \mathbf{P}_i} \cdot \frac{\partial \mathbf{P}_i}{\partial r_i^k} \\
&= \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{C} -\frac{\lambda}{(C+1)} \cdot \frac{1}{\mathbf{P}_i} \cdot \left\{ \frac{\partial p_i^k}{\partial r_i^k}, \{\frac{\partial p_i^j}{\partial r_i^k}\}_{j=1,j\neq k}^{C} \right\} \\
&= \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{C} -\frac{\lambda}{(C+1)} \cdot \frac{1}{\mathbf{P}_i} \cdot \left\{ p_i^k(1-p_i^k), \{-p_i^k \cdot p_i^j\}_{j=1,j\neq k}^{C} \right\}, \\
&= \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{C} -\frac{\lambda}{(C+1)} \cdot \frac{p_i^k}{\mathbf{P}_i} \cdot \left\{ 1-p_i^k, \{-p_i^j\}_{j=1,j\neq k}^{C} \right\} \\
&= \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{C} -\frac{\lambda}{(C+1)} \cdot \frac{p_i^k}{\mathbf{P}_i} \cdot (\mathbf{1}_k - \mathbf{P}_i)
\end{aligned} \tag{4}$$

where $\mathbf{1}_k$ is an one-hot manner vector and $k_{th}$ value is 1.

$$\begin{aligned}
\frac{\partial \mathcal{L}_{Firth}}{\partial r_i^k} &= \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{C} -\frac{\lambda}{(C+1)} \cdot \frac{p_i^k}{\mathbf{P}_i} \cdot (\mathbf{1}_k - \mathbf{P}_i) \\
&= \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{C} -\frac{\lambda}{(C+1)}(1 - C \cdot p_i^k) \\
&= \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{C} \frac{\lambda}{(C+1)}(C \cdot p_i^k - 1)
\end{aligned} \tag{5}$$

The gradient of $\mathcal{L}$ with respect to $r_i^k$ is

$$\frac{\partial \mathcal{L}}{\partial r_i^k} = \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{C} \left[ (p_i^k - y_i^k) + \frac{\lambda}{(C+1)}(C \cdot p_i^k - 1) \right]. \tag{6}$$

The gradients of $\mathcal{L}$ with respect to $\mathbf{r}_i = \{r_i^k\}_{k=1}^{C}$ is

$$\frac{\partial \mathcal{L}}{\partial \mathbf{r}_i} = \frac{1}{M}\sum_{i=1}^{M} \left[ (\mathbf{P}_i - \mathbf{y}_i) + \frac{\lambda}{(C+1)}(C\mathbf{P}_i - \mathbf{E}) \right], \tag{7}$$

where $\mathbf{E}$ is a one-full vector with the same size as $\mathbf{P}_i$. As given in Eq. 6, we found that when $\mathbf{y}_i = \mathbf{1}_c$ (i.e., $\mathbf{x}_i$ belongs to class $c$) and $\lambda > 0$:

$$\begin{cases} \lambda(C \cdot p_i^k - 1) < 0, & k \neq c \\ \lambda(C \cdot p_i^k - 1) > 0, & k = c \end{cases} \tag{8}$$

because $p_i^c$ is usually larger than $\frac{1}{C}$ and other logit ($\{p_i^k\}_{k\neq c}$) is usually smaller than $\frac{1}{C}$. In other words, FBR increases the gradients with respect to $r_i^c$ and decreases other gradients to strengthen the influence of $\mathbf{y}_i$ to the trained model.

## B THE ANALYTIC GRADIENTS OF LP-FT-FB

We give the analytic formula of LP-FT-FB as follows. The gradients of linear classifier in LP are

$$\begin{aligned}
\beta' &= \beta + \alpha_1 \cdot \frac{\partial \mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})}{\partial \mathbf{r}} \cdot \frac{\partial \mathbf{r}}{\partial \beta} \\
&= \beta + \frac{\alpha_1}{K \cdot C_{few}} \cdot \sum_{i=1}^{K \cdot C_{few}} \frac{\partial \mathcal{L}(\hat{\mathbf{y}}_i, \mathbf{y}_i)}{\partial \mathbf{r}_i} \cdot \frac{\partial \mathbf{r}_i}{\partial \beta} \\
&= \beta + \frac{\alpha_1}{K \cdot C_{few}} \cdot \sum_{i=1}^{K \cdot C_{few}} \left[ (\hat{\mathbf{y}}_i - \mathbf{y}_i) + \frac{\lambda}{C_{few}+1} \cdot (C_{few}\hat{\mathbf{y}}_i - \mathbf{E}) \right] \cdot \frac{\partial \mathbf{r}_i}{\partial \beta}
\end{aligned} \tag{9}$$

Then, at the FT stage, the gradients of feature extractor are

$$\begin{aligned}\hat{\theta} &= \theta_0 + \frac{\partial \mathcal{L}(\hat{\mathbf{y}}', \mathbf{y})}{\partial \mathbf{r}'} \cdot \frac{\partial \mathbf{r}'}{\partial \theta_0} \\ &= \theta_0 + \frac{\alpha_2}{K \cdot C_{few}} \cdot \sum_{i=1}^{K \cdot C_{few}} \frac{\partial \mathcal{L}(\hat{\mathbf{y}}'_i, \mathbf{y}_i)}{\partial \mathbf{r}'_i} \cdot \frac{\partial \mathbf{r}'_i}{\partial \theta_0} \\ &= \theta_0 + \frac{\alpha_2}{K \cdot C_{few}} \cdot \sum_{i=1}^{K \cdot C_{few}} \left[ (\hat{\mathbf{y}}'_i - \mathbf{y}_i) + \frac{\lambda_{inv}}{C_{few}+1} \cdot (C_{few}\hat{\mathbf{y}}'_i - \mathbf{E}) \right] \cdot \frac{\partial \mathbf{r}'_i}{\partial \theta_0}\end{aligned} . \tag{10}$$

The gradients of linear classifier in FT are

$$\begin{aligned}\hat{\beta} &= \beta' + \alpha_2 \cdot \frac{\partial \mathcal{L}(\hat{\mathbf{y}}', \mathbf{y})}{\partial \mathbf{r}'} \cdot \frac{\partial \mathbf{r}'}{\partial \beta'} \\ &= \beta' + \frac{\alpha_2}{K \cdot C_{few}} \cdot \sum_{i=1}^{K \cdot C_{few}} \frac{\partial \mathcal{L}(\hat{\mathbf{y}}'_i, \mathbf{y}_i)}{\partial \mathbf{r}'_i} \cdot \frac{\partial \mathbf{r}'_i}{\partial \beta'} \\ &= \beta' + \frac{\alpha_2}{K \cdot C_{few}} \cdot \sum_{i=1}^{K \cdot C_{few}} \left[ (\hat{\mathbf{y}}'_i - \mathbf{y}_i) + \frac{\lambda_{inv}}{C_{few}+1} \cdot (C_{few}\hat{\mathbf{y}}'_i - \mathbf{E}) \right] \cdot \frac{\partial \mathbf{r}'_i}{\partial \beta'}\end{aligned} . \tag{11}$$

## C ALGORITHM

We give the whole flow as follows for a better understanding of the proposed method and reproducing it.

**Algorithm 1** LP-FT-FB

**Require:** $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{1 \le i \le M}$, Episodic Sampler Operator (ES), pre-trained $\mathbf{B}_0(\cdot, \theta_0)$, $\lambda$, $\lambda_{inv}$, $\alpha_1$, and $\alpha_2$;

**for** $1 \le p \le P$ **do**
- $\mathcal{T}_p = ES(\mathcal{D}) = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{K \cdot C_{few}}$ for $C_{few}$-*way*-K-*shot* FSL tasks;
- $\mathbf{z} = \mathbf{B}_0(\mathbf{x}, \theta_0)$;
- Initialize randomly a linear classifier $\mathbf{v}(\cdot, \beta)$;
- $\mathbf{r} = \beta \cdot \mathbf{z}$, $\hat{\mathbf{y}} = \mathbf{v}(\mathbf{z}, \beta)$;
- **LP**: $\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{C_{few} \cdot K} \sum_{i=1}^{K \cdot C_{few}} \mathcal{L}(\hat{\mathbf{y}}_i, \mathbf{y}_i)$;
- $\mathbf{v}(\cdot, \beta)$ is updated by Eq. 9 to obtain $\mathbf{v}'(\cdot, \beta')$;
- $\hat{\mathbf{y}}'$ is computed;
- **FT**: The loss of FT is computed;
- The feature extractor $\mathbf{B}_0(\cdot, \theta_0)$ is updated by Eq. 10 to obtain $\hat{\mathbf{B}}(\cdot, \hat{\theta})$;
- The classifier is updated by Eq. 11 to obtain $\hat{\mathbf{v}}(\cdot, \hat{\beta})$;
- **Output:** $\{\hat{\mathbf{B}}(\cdot, \hat{\theta}), \hat{\mathbf{v}}(\cdot, \hat{\beta})\}$;
- **Evaluated by on Output on test samples**

**end for**

## D EXTRA EXPERIMENTS

### D.1 CROSS-DOMAIN RESULTS

Besides the cross-domain tasks with DC Yang et al. (2021), we also give some experimental results without DC to show the equivariance brought by LP-FT-FB. The evaluated tasks are 5-way tasks. As given in Table 1, LP-FT-FB outperforms S2M2 by a large margin. This verifies the equivariance adapting the novel features to target domain.

Table 1: 5-way experiments for cross-domain FSL tasks.

| Methods | *mini*-Imagenet → CUB | | *tiered*-Imagenet → CUB | |
|---|---|---|---|---|
| | 1-shot | 5-shot | 1-shot | 5-shot |
| S2M2 Mangla et al. (2020) | 48.24 | 70.44 | 72.46 | 88.02 |
| LP-FT-FB | **50.47** | **71.62** | **74.56** | **89.42** |

### D.2 Experiments on 2-layer MLP

We added experiments with a 2-layer-MLP linear layer. The results are given as follows. The experimental settings: we add a fully-connected layer and a ReLU layer before the cosine normalized layer Chen et al. (2019). The added fully-connected layer contains $d \times d$ weight parameters without bias. d is the length of the extracted features. The experiments are conducted on *mini*-Imagenet for the 5-way-1-shot task. The 2-layer MLP improved the performance of FBR by 0.2%. And the proposed LP-FT-FB outperforms it by 1.38%. It still works well.

Table 2: The experiments on 2-layer MLP.

| Method | 5-way-1-shot |
|---|---|
| FBR Ghaffari et al. (2022) | 65.59 |
| LP-FT-FB | 67.04 |
| FBR + 2-layer-MLP | 65.80 |
| LP-FT-FB + 2-layer-MLP | 67.18 |

## E Related Meta-learning-based Methods

The meta-learning FSL methods used meta-learning methods to learn the learning patterns of the models from different N-way-K-shot tasks instead of learning the pattern of the target tasks. It can be viewed as a special regularization method to avoid the learned models being over-fitted on the pre-training samples. The learning patterns contain the gradients for updating the models and the metrics of the responding features, i.e., gradient-based and metric-based methods.

The gradient-based methods are original from Model-Agnostic Meta-Learning (MAML) Finn et al. (2017). MAML learned second-order gradients to update the pre-trained model for each task. Following MAML, Nichol *et al.* proposed Reptile Nichol et al. (2018) to use the learned first-order gradients to approximate the second-order gradients to avoid complex computation of Hessian matrices. Song *et al.* proposed using Evolution Strategies Song et al. (2020) to obtain an algorithm, which avoids the problem of estimating second derivatives for solving the complex computation problem.

However, gradient-based methods are at high computation costs, Snell proposed Prototypical Networks Snell et al. (2017) to use Euclidean distance to replace the linear classifier, which is oversensitive to the base samples. Then sung *et al.* proposed Relation Networks Sung et al. (2018) to improve the distance by learning the similarity between the support and query features. Recently, Zhang *et al.* proposed a variational method Zhang et al. (2019) to address the biased point estimation problem of metric-based methods.

## References

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. 2017. URL http://arxiv.org/abs/1703.03400.

Saba Ghaffari, Ehsan Saleh, David Forsyth, and Yu-Xiong Wang. On the importance of firth bias reduction in few-shot classification. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=DNRADop4ksB.

Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *The IEEE Winter Conference on Applications of Computer Vision*, pp. 2218–2227, 2020.

Alex Nichol, Joshua Achiam, and J. Schulman. On first-order meta-learning algorithms. *ArXiv*, abs/1803.02999, 2018.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf.

Xingyou Song, Wenbo Gao, Yuxiang Yang, Krzysztof Choromanski, Aldo Pacchiano, and Yunhao Tang. Es-maml: Simple hessian-free meta learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1exA2NtDB.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=JWOiYxMG92s.

Jian Zhang, Chenglong Zhao, Bingbing Ni, Minghao Xu, and Xiaokang Yang. Variational few-shot learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1685–1694, 2019. doi: 10.1109/ICCV.2019.00177.