

# APPENDIX

## TOWARDS UNDERSTANDING THE CAUSE OF ERROR IN FEW-SHOT LEARNING

**Anonymous authors**

Paper under double-blind review

### 1 ALGORITHM

#### 1.1 ALGORITHM

---

##### Algorithm 1 Reducing Classifier Discrepancy (RCD)

---

**Require:**  $C^{base}$ : base classes;  $D^{base}$ : dataset of base classes

---

##### Phase 1: Pre-training in conventional way

---

**Require:**  $\alpha$ : learning rate

**Initialize:**

Initialize feature extractor  $F_\theta$  and predictor  $h_W$

**for** batch = 1, 2, . . . **do**

Sample a batch of samples  $\{(x, y)\}$  from  $D^{base}$

Predict labels of samples  $\hat{y} = h_W(F_\theta(x))$

Compute cross-entropy loss  $L_{ce}(y, h(F(X)))$

Update parameter of feature extractor  $\theta \leftarrow \theta - \alpha \nabla L_{ce}$

Update classification weights  $W \leftarrow W - \alpha \nabla L_{ce}$

**end for**

---

##### Phase 2: Meta-train with proposed auxiliary loss

---

**Require:**  $\alpha$ : learning rate;  $\beta$ : weight of auxiliary loss

**Require:**  $h^* = \operatorname{argmin}_h L_{ce}(h, F; D^{base})$ : the ideal predictor obtained in Phase 1

**Initialize:**

Initialize feature extractor  $F_{\theta'}$

**for** iteration = 1, 2, . . . **do**

Compose a task  $\tau$ : sample  $N$  categories from  $C^{base}$ ; sample a support set  $D_s$  and a query set  $D_q$  for  $N$  categories from  $D^{base}$

Estimate task-specific predictor on  $D_s$ :  $h = \arg \min_h L_{ce}(y, h(F(D_s)))$

Compute cross-entropy loss on  $D_q$ :  $L_{ce}(y, h(F(D_q)))$

Compute classifier discrepancy  $L_{dis}(h^*, h)$

Update the feature extractor  $\theta' \leftarrow \theta' - \alpha \nabla (L_{ce} + \beta L_{dis})$

**end for**

---

### 2 PROOFS

Notations used in this section are first given in Table 1. For simplification, notations in this part are subtly different with those in paper.

Table 1: Notations.

$h$	A linear hypothesis
$h^*$	The ideal linear hypothesis
$F$	Feature extractor
$\epsilon_B$	Error rate on base classes
$\epsilon_N$	Error rate on novel classes
$D_B$	Dataset of base classes
$D_N$	Dataset of novel classes
$dis$	Disagreement
$\Lambda$	A linear transform

## 2.1 PROOF OF PROPOSITION 1

*Proof.* From the definition of  $\epsilon$ , we can split it into two parts as follow:

$$\begin{aligned}\epsilon(h; F) &= E_{(x,y) \in D}[\mathbb{1}(y \neq h(F(x)))] \\ &= E_{(x,y) \in D}[\mathbb{1}(y \neq h(F(x)) \wedge h^*(F(x)) \neq h(F(x)))] \\ &\quad + E_{(x,y) \in D}[\mathbb{1}(y \neq h(F(x)) \wedge h^*(F(x)) = h(F(x)))]\end{aligned}\tag{1}$$

The disagreement can be decomposed as:

$$\begin{aligned}dis(h, h^*; F) &= E_{(x,y) \in D}[\mathbb{1}(h(F(x)) \neq h^*(F(x)))] \\ &= E_{(x,y) \in D}[\mathbb{1}(h(F(x)) \neq h^*(F(x)) \wedge y \neq h(F(x)))] \\ &\quad + E_{(x,y) \in D}[\mathbb{1}(h(F(x)) \neq h^*(F(x)) \wedge y = h(F(x)))]\end{aligned}\tag{2}$$

The error rate of the best hypothesis can be rewritten as:

$$\begin{aligned}\epsilon(h^*; F) &= E_{(x,y) \in D}[\mathbb{1}(y \neq h^*(F(x)))] \\ &= E_{(x,y) \in D}[\mathbb{1}(y \neq h^*(F(x)) \wedge h^*(F(x)) \neq h(F(x)))] \\ &\quad + E_{(x,y) \in D}[\mathbb{1}(y \neq h^*(F(x)) \wedge h^*(F(x)) = h(F(x)))]\end{aligned}\tag{3}$$

The second term of Eqn. 3 can be further rewritten as:

$$\begin{aligned}E_{(x,y) \in D}[\mathbb{1}(y \neq h^*(F(x)) \wedge h^*(F(x)) = h(F(x)))] \\ = E_{(x,y) \in D}[\mathbb{1}(y \neq h(F(x)) \wedge h^*(F(x)) = h(F(x)))]\end{aligned}\tag{4}$$

Notice that the first term of Eqn. 2 and the second term of Eqn. 3 are identical to the two terms of Eqn. 1 respectively. And the rest terms of Eqn. 2 and Eqn. 3 are positive numbers. So the sum of  $\epsilon(h^*; F)$  and  $dis(h, h^*; F)$  is larger equal than  $\epsilon(h; F)$ . The Proposition 1 is proved hereto.

## 2.2 PROOF OF LEMMA 1

*Proof.* Let  $h^*$  be the ideal hypothesis for the novel classes and  $h'^*$  be the ideal hypothesis for the base classes. There exists a linear transformation  $h'^* = \Lambda(h^*)$ . The classification weight of the hypothesis has  $W_n = W_b \tilde{W}$ .

From the triangular inequality, and the definition of  $\mathcal{H}\Delta\mathcal{H}$  divergence we have:

$$\begin{aligned}\epsilon_N(h^*) &\leq \epsilon_N(h'^*) + \epsilon_N(h^*, h'^*) \\ &\leq \epsilon_N(h'^*) + \epsilon_B(h^*, h'^*) + |\epsilon_B(h^*, h'^*) - \epsilon_N(h^*, h'^*)| \\ &\leq \epsilon_B(h'^*) + \epsilon_N(h'^*) + \epsilon_B(\Lambda^{-1}(h'^*)) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_B, D_N) \\ &\leq \epsilon_B(h'^*) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_B, D_N) + \lambda\end{aligned}\tag{5}$$

where  $\lambda = \min_h \epsilon_N(h) + \epsilon_B(\Lambda^{-1}(h))$ .

### 2.3 PROOF OF LEMMA 2

*Proof.* From the definition of  $\mathcal{H}\Delta\mathcal{H}$  divergence (Ben-David et al. (2010)) we have:

$$|dis_t(h, h') - dis_s(h, h')| \leq \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_s, D_t) \quad (6)$$

where  $D_s$  is the source domain and  $D_t$  is the target domain.

Consider any hypothesis  $h$  and the ideal hypothesis  $h^*$  on novel classes. We define an intermediate set  $D_I = \Lambda(D_B)$ . In this paper, we assume the novel set as the target domain and the intermediate set as the source domain. Then we have:

$$dis_N(h, h^*) \leq \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\Lambda(D_B), D_N) + dis_I(h, h^*) \quad (7)$$

$h'^* = \Lambda(h^*)$ , then we can rewrite  $dis_T(h, h^*)$  as:

$$dis_I(h, h^*) = dis_B(\Lambda(h), \Lambda(h^*)) = dis_B(\Lambda(h), h'^*) \quad (8)$$

Then Eqn. 7 can be rewritten as:

$$\begin{aligned} dis_N(h, h^*) &\leq \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_I, D_N) + dis_B(\Lambda(h), h'^*) \\ &\leq dis_B(h', h'^*) + dis_B(h', \Lambda(h)) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\Lambda(D_B), D_N) \end{aligned} \quad (9)$$

## 3 EXTENDED DISCUSSION

In this section, we discuss the relation among our paper and state-of-the-arts in few-shot learning.

### 1. *Rethinking Few-Shot Image Classification: a Good Embedding Is All You Need?* (Tian et al. (2020))

On the one hand, Rethinking-FSC indicates that learning a supervised feature representation can achieve strong performance. Objective in Rethinking-FSC can be formulated as  $\min_{F, h} L_{ce}(h, F; D^{base})$ . In our paper, we theoretically point out that few-shot performance is effected by feature separability. Feature separability can be measured by the error rate of the ideal linear predictor, represented by  $\min_h \epsilon(h, F; D^{base})$ . Since this measurement is non-differentiable, we can use cross-entropy loss  $\min_h L_{ce}(h, F; D^{base})$  for estimation. Our paper aims to maximizing linear separability by  $\min_F \min_h L_{ce}(h, F; D^{base})$  which is consistent with the objective in supervised training. On the other hand, Rethinking-FSC uses knowledge distilling to boost few-shot performance. Objective in distilling can be formulated as  $\min_{h, F} L_{KL}(h(F(D^{base})), h'(F'(D^{base})))$  where  $F$  and  $F'$  are feature extractors in different generations. In our paper, we theoretically analyze the relation of error and classifier discrepancy and further propose to use KL divergence as auxiliary constraint. Our objective is  $\min_F L_{KL}(h(F(D^{base})), h^*(F'(D^{base})))$  where  $h^*$  is approximated by  $h' = \operatorname{argmin}_h L_{ce}(h, F; D^{base})$ . Objectives in two methods are consistent. In summary, our paper theoretically explains aforementioned two concepts in Rethinking-FSC.

### 2. *Unraveling Meta-Learning: Understanding Feature Representations for Few-Shot Tasks* (Cao et al. (2019))

Cao et al. (2019) propose that variance in the feature space has important effect on few-shot performance. Since classifier in few-shot learning is sample-dependent, the variance of feature representations influences decision boundaries thus affects stability of classifier. Similarly in our paper, we study on discrepancy between task-specific and task-independent classifiers. From experiment results and visualization, we find that in better clustered feature space, classifier discrepancy is relatively small and accuracy is higher. Our paper supports the conclusion proposed by Cao et al. (2019) from another side.

### 3. *Prototype Rectification for Few-Shot Learning* (Liu et al. (2020))

Liu et al. (2020) give a theoretical lower bound of Cosine Similarity based Prototypical Network, demonstrating that intra-class bias and cross-class bias are key influencing factors. The intra-class

bias refers to the difference of the prototype estimated from limited samples and the expected prototype. If the intra-class bias is quantified by  $MSE$ , our proposed classifier discrepancy measured by  $dis_{MSE}$  in PN can approximate to it. Liu et al. (2020) point out that the intra-class bias is larger in fewer-shot scenario which is consistent with the findings in Sec. 3.

## 4 EXPERIMENTS

### 4.1 DETAILS

In this section, we give the details of how to measure classification performance on novel classes in Sec. 4.1. We first train a feature extractor in supervised way. Then we fix the feature extractor and extract features of the samples in novel set. The ideal classifier is defined as  $h^* = \operatorname{argmin}_h \epsilon(D; h, F)$ . To obtain a differentiable criterion, we use cross entropy to approximate it  $h^* = \operatorname{argmin}_h L_{ce}(D; h, F)$ .

For ProtoNet, we can calculate the ideal classifier by the definition:

$$P_i = \frac{\sum F_\theta(x) \cdot \mathbb{1}(y == i)}{\sum \mathbb{1}(y == i)} \quad (10)$$

For LR, we can calculate the ideal classifier by:

$$W = \operatorname{argmin}_w L_{ce}(D^{base}; h_w, F_\theta) \quad (11)$$

For RR, we can calculate the ideal classifier by:

$$W = (X^T X + \gamma I)^{-1} X^T Y \quad (12)$$

where  $Y$  is the one-hot label.

$dis(h, h^*; F)$  is computed from 600 randomly sampled episodes.

### 4.2 HYPERPARAMETERS

In phase 1, we use a fully connected layer as classifier. Hyperparameters are displayed in Table 2.

Table 2: Hyperparameters.

Phase 1	Optimizer	SGD
	Learning Rate	0.001
	Weight Decay	0.0005
	Momentum	0.9
	Max Epoch	500
	LR Decay Milestones	75, 150, 300
	LR Decay Gamma	0.1
	Batch Size	16
	Steps per Epoch	2400
Phase 2	Optimizer	SGD
	Learning Rate	0.0001
	Weight Decay	0.0005
	Momentum	0.9
	Batch Size	1
	Max Episodes	20000
	$\beta$	50

### 4.3 VISUALIZATION

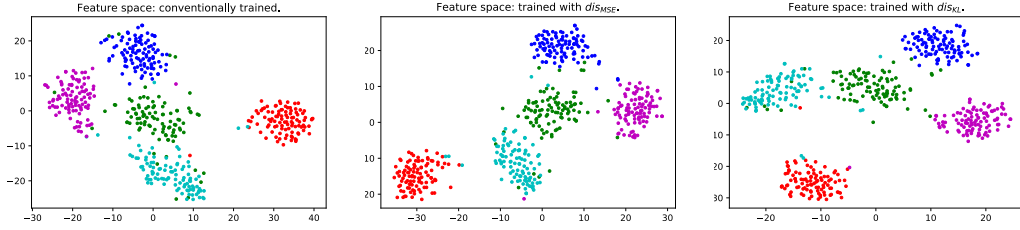


Figure 1: t-SNE visualization on tiered-ImageNet

### 4.4 DATASET

The **mini-ImageNet** (Vinyals et al. (2016)) consists of 100 classes from ImageNet (Deng et al. (2009)) and each class has 600 images of size  $84 \times 84$ . We follow the standard split proposed in (Ravi & Larochelle (2017)): dataset is divided into three subsets which has 64, 16, and 20 classes for training, validation, and test.

The **tiered-ImageNet** (Ren et al. (2018)) has 608 classes which are randomly chosen from the ImageNet (Deng et al. (2009)). As proposed in (Ren et al. (2018)), dataset is split into training, validation, test subsets which contain 351, 97, and 160 classes respectively. It is further split into 34 high-level semantic categories, including 20, 6, 8 classes for training, validation and test. In total, there are 779,165 images with a size of  $84 \times 84$ .

The **CIFAR-FS** includes 100 classes which is derived from CIFAR-100 dataset (Bertinetto et al. (2018)). Each class has 600 images of size  $32 \times 32$ . The whole dataset is divided into three subsets: 64 training classes, 16 validation classes and 20 test classes.

### 4.5 BACKBONE

ConvNet-64 (Snell et al. (2017)) is composed of 4 convolutional modules with  $3 \times 3$  convolutions, each followed by a BatchNorm layer (Ioffe & Szegedy (2015)), a ReLU nonlinearity (Nair & Hinton (2010)), and a  $2 \times 2$  max-pooling unit. With input images of size  $84 \times 84$ , the output feature map has size of  $64 \times 5 \times 5$ . Backbone is followed by a global average pooling layer and outputs a 64-dimension feature vector. ResNet-12 (Lee et al. (2019)) is a shallow residual network. It is composed of 4 residual blocks, and each residual block consists of 3 convolutional layers with  $3 \times 3$  kernel and a max-pooling layer. A global average-pooling layer follows at the end of the backbone. The final output feature vector is 640-dimension.

## REFERENCES

- Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010.
- Luca Bertinetto, Joao F Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2018.
- Tianshi Cao, Marc T Law, and Sanja Fidler. A theoretical analysis of the number of shots in few-shot learning. In *ICLR*, 2019.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255. Ieee, 2009.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pp. 448–456, 2015.

- Kwonjoon Lee, Subhansu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, pp. 10657–10665, 2019.
- Jinlu Liu, Liang Song, and Yongqiang Qin. Prototype rectification for few-shot learning. *ECCV*, 2020.
- Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pp. 807–814, 2010.
- Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.
- Mengye Ren, Sachin Ravi, Eleni Triantafillou, Jake Snell, Kevin Swersky, Josh B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018.
- Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, pp. 4077–4087, 2017.
- Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *ECCV*, 2020.
- Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, pp. 3630–3638, 2016.