TOWARDS CLASS-BALANCED TRANSDUCTIVE FEW-SHOT LEARNING

Anonymous authors
Paper under double-blind review

A EXTENDED DISCUSSION ON CLASS-BALANCED NORMALIZATION

In this section, we give a comprehensive overview on the difference between Class-balanced Normalization and Distribution Alignment. The different learning status of classes can be reflected in the learned marginal distribution $\hat{p}(\mathbf{y})$: weakly-learned classes have smaller values compared with the others. During fine-tuning, if the learned marginal distribution is aligned towards uniform, it serves to encourage a fair learning status among classes. The previous work Berthelot et al. (2019) in semi-supervised learning proposes a formulation to conduct distribution alignment as:

$$\tilde{\mathbf{q}} = \text{Normalize}(\mathbf{q}\frac{p(\mathbf{y})}{\hat{p}(\mathbf{y})}), \tag{1}$$

where $p(\mathbf{y})$ is the marginal distribution of the class variable $\mathbf{y}$ and $\hat{p}(\mathbf{y})$ is its estimation. $\mathbf{q}$ refers to the probability of a unlabeled sample. And $\text{Normalize}(x_i) = \frac{x_i}{\sum_j x_j}$. In Eq. 6, the scale vector $\frac{p(\mathbf{y})}{\hat{p}(\mathbf{y})}$ quantifies the difference between the estimation with its expected value, which is further applied on each sample to adjust the predicted probability $\mathbf{q}$. Specifically, scalars of weakly-learned classes are larger compared with others and by multiplying with the scalars, predicted probabilities for weakly-learned classes would be raised accordingly. The overall effect of Eq. 6 works as an alignment to match $\hat{p}(\mathbf{y})$ and $p(\mathbf{y})$.

Class-balanced normalization aims to adjust the predicted probabilities of testing data individually to pursue class-wise balanced fine-tuning. Therefore, testing data assists in achieving a balanced learning status among classes during fine-tuning, and the imbalance issue in predictions is correspondingly solved to a degree. Class-balanced normalization follows the same formulation in Eq. 6. Specifically, the expected marginal distribution $p(\mathbf{y})$ is assigned as a uniform distribution. For $\mathbf{x} \in \mathcal{D}_q$, it is combined with the full support set as $x \cup \mathcal{D}_s$; and the current learned marginal probability is estimated using $x \cup \mathcal{D}_s$, which is further aligned with Uniform. Formally, for $\mathbf{q} = p_\theta(\mathbf{y}|\mathbf{x})$:

$$\tilde{\mathbf{q}} = \text{Normalize}(\mathbf{q}\frac{U}{\hat{E}_{x\cup\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]}) \tag{2}$$

When Distribution Alignment is applied in semi-supervised learning Berthelot et al. (2019), $p(\mathbf{y})$ is estimated from the labeled data and $\hat{p}(\mathbf{y})$ is computed from predictions of the whole unlabeled set. In other words, there is an assumption that the unlabeled data shares the same prior with the labeled training set. However, in transductive FSL, this assumption hardly works. The marginal distribution of a handful of labeled data is biased from the actual marginal distribution. Furthermore, giving an assumption of prior for testing images is not a wise choice that would limit the algorithm's actual application situation. A thorough experimental comparison of DA with our method is provided in the following experimental section.

A.1 TOWARDS A THEORETICAL OVERVIEW

We develop this discussion under the scope of an episode of few-shot classification, namely the support set $\mathcal{D}_s = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_s}$ and the query set $\mathcal{D}_q = \{(\mathbf{x}_i)\}_{i=1}^{N_q}$; N_s and N_q are the total number of samples in support set and query set, respectively. The marginal distribution of class variables can

be estimated separately as:

$$\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})] = \frac{1}{N_s}\sum_{\mathbf{x}\in\mathcal{D}_s} p_\theta(\mathbf{y}|\mathbf{x}) \tag{3}$$

$$\hat{E}_{\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})] = \frac{1}{N_q}\sum_{\mathbf{x}\in\mathcal{D}_q} p_\theta(\mathbf{y}|\mathbf{x}) \tag{4}$$

And if considering all available data:

$$\hat{E}_{\mathcal{D}_s\cup\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})] = \frac{N_s}{N_s+N_q}\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})] + \frac{N_q}{N_q+N_s}\hat{E}_{\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})] \tag{5}$$

The previous work (Berthelot et al., 2019) in semi-supervised learning proposes a formulation to conduct distribution alignment as:

$$\tilde{\mathbf{q}} = \mathrm{Normalize}(\mathbf{q}\frac{p(\mathbf{y})}{\hat{p}(\mathbf{y})}), \tag{6}$$

where $p(\mathbf{y})$ is the marginal distribution of the class variable $\mathbf{y}$ and $\hat{p}(\mathbf{y})$ is its estimation. $\mathbf{q}$ refers to the probability of a unlabeled sample. And $\mathrm{Normalize}(x_i) = \frac{x_i}{\sum_j x_j}$.

Case 1: $\hat{p}(\mathbf{y}) = \hat{E}_{\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})]$ **and** $p(\mathbf{y}) = \hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]$ refers to *Est. + All Query*. Case 1 refers to the original DA in (Berthelot et al., 2019), which aligns the marginal distribution of unlabeled data to the labeled data.

Case 2: $\hat{p}(\mathbf{y}) = \hat{E}_{\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})]$ **and** $p(\mathbf{y}) = \mathcal{U}$ refers to *Uni. + All Query*.

For both Case 1 and Case 2, an assumption that the testing distribution is the same with the prior distribution $p(\mathbf{y})$ is explicitly made. However this assumption limits the algorithm's generalization by only considering a uniform testing distribution. We discuss these two cases together as the only difference is $p(\mathbf{y})$ as Uniform or not.

For these cases, all query samples share the same scale vector: $\frac{p(\mathbf{y})}{\hat{E}_{x\cup\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]}$, under which the marginal distribution of testing set is changed accordingly:

$$\hat{E}_{\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})] \sim \frac{1}{N_q}\sum_{\mathbf{x}\in\mathcal{D}_q} \frac{p(\mathbf{y})}{\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]} p_\theta(\mathbf{y}|\mathbf{x}) \tag{7}$$

The $\mathrm{Normalize}$ is omitted to simplify the expression ($\sim$ is used accordingly). $\hat{E}_{\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})] \to p(\mathbf{y})$ and the overall estimated marginal distribution is:

$$\hat{E}_{\mathcal{D}_s\cup\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})] \sim \frac{N_s}{N_s+N_q}\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})] + \frac{N_q}{N_q+N_s}p(\mathbf{y}) \tag{8}$$

For Case 1 and Case 2, the estimated marginal distribution of labeled data remains unchanged while the marginal distribution of testing data is forced to approach a prior $p(\mathbf{y})$ either a Uniform distribution or the same marginal distribution with labeled data.

Case 3: $\hat{p}(\mathbf{y}) = \hat{E}_{x\cup\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]$ **and** $p(\mathbf{y}) = \mathcal{U}$ refers to *Uni. + Single Query*. This case is our proposed Class-balanced Normalization, where $\hat{p}(\mathbf{y})$ is estimated by combining each testing data with the full support set, to avert making any assumption on the testing distribution.

Class-balanced Normalization allows a unique scale vector to adjust the predicted probability for each testing data, under which the marginal distribution of testing set is changed as:

$$\hat{E}_{\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})] \sim \frac{1}{N_q}\sum_{\mathbf{x}\in\mathcal{D}_q} \frac{\mathcal{U}}{\hat{E}_{x\cup\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]} p_\theta(\mathbf{y}|\mathbf{x}) \tag{9}$$

The estimated marginal probability with each one query sample x_q and the full support set can be expanded as:

$$\hat{E}_{x_q \cup \mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})] = \frac{1}{1+N_s}[p_\theta(\mathbf{y}|\mathbf{x}_q) + \sum_{x \in \cup \mathcal{D}_s} p_\theta(\mathbf{y}|\mathbf{x})] = \frac{p_\theta(\mathbf{y}|\mathbf{x}_q)}{1+N_s} + \frac{N_s}{1+N_s}\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})] \tag{10}$$

And the overall estimated marginal distribution can be approximately expressed as:

$$\begin{aligned}
\hat{E}_{\mathcal{D}_s \cup \mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})] &\sim \frac{N_s}{N_s+N_q}\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})] + \frac{N_q}{N_q+N_s}\frac{1}{N_q}\sum_{\mathbf{x} \in \mathcal{D}_q} \frac{\mathcal{U}}{\hat{E}_{x \cup \mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]} p_\theta(\mathbf{y}|\mathbf{x}) \\
&\sim \frac{N_s}{N_s+N_q}\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})] - \frac{1+N_s}{N_q+N_s}\sum_{x \in D_q} \frac{\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]}{p_\theta(\mathbf{y}|\mathbf{x}) + \hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]} \\
&+ \frac{1+N_s}{N_q+N_s}\hat{E}_{x_q \cup \mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]
\end{aligned}$$

By using Class-balanced Normalization, $\hat{E}_{x \cup \mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]$ is aligned to Uniform, which doesn't imply a distribution assumption on the testing data and $\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]$ is implicitly adjusted as well. In doing so, the class-wise balance is improved, and the distribution alignment of testing data is achieved without introducing any prior assumption on the overall testing set.

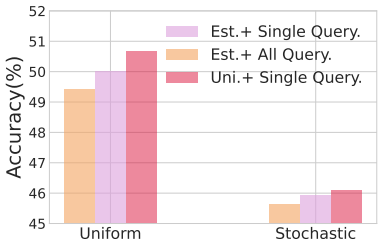


Figure 1: Ablations on Class-balanced Normalization with uniform / stochastic setting of testing data.

A.2 Experimental Results

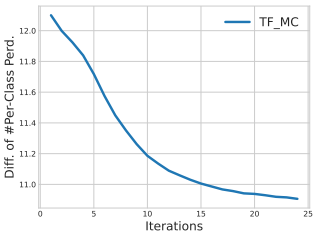
In Fig. 1, we verify the design of CN and its robust performance over different settings of query set using ILSVRC-2012 validation, namely the uniform setting with equal number of per-class testing samples and the stochastic setting with various numbers of per-class testing samples. Note that the performance of stochastic setting is lower than performance on the uniform setting, which is caused by the difficulty level of the setting itself. The stochastic setting is more challenging that the number of per-class testing samples are randomly sampled in the range of [0, 50]. With the challenging stochastic setting, CN could still improve the performance around 0.5%. We ablation the design of Class-balanced Normalization: for the expected marginal distributions $p(\mathbf{y})$, we experiment on uniform distribution (Uni.) and prior distribution from estimating labeled data (Est.); for the estimated marginal distribution $\hat{p}(\mathbf{y})$, using all query set (All Query.) and using one single query set with the support set (Single Query.) are separately experimented. Using Est. as $p(\mathbf{y})$, Single Query. shows better performance than All Query, which indicates the effectiveness of enabling a sample-specific scale vector by Single Query. Meanwhile, using Uni. as $p(\mathbf{y})$ wins over Est as Uni., which indicates applying a stronger regularization like Uniform is beneficial to encourage class-balanced finetuning. Note that DA in (Berthelot et al., 2019) is Est.+ All Query. and Uni.+ Single Query. CN outperforms DA in both uniform and stochastic testing settings.

B Extended Discussion on Class-imbalanced Predictions

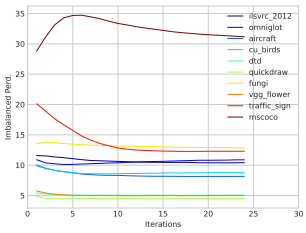
B.1 Implementation Details of the Observation

The Figure.1 showing the class-imbalanced predictions puts together methods of DCMSS (Tao et al., 2022), Transductive-Finetuning (our implementation on feature space) (Dhillon et al., 2019), URL (Li et al., 2021) and TSA (Li et al., 2022). Results from DCMSS(Tao et al., 2022), Transductive-Finetuning (Dhillon et al., 2019) are out-of-domain evaluations for meta-Dataset (Table.2 in the main paper) and the others are in-domain evaluations for meta-Dataset (table.3).

For DCMSS (Tao et al., 2022), we reproduce the results following the same setting mentioned in the paper, which in detail ResNet-18 trained with ILSVRC-2012 training set is used as the



(a) Average over Datasets



(b) Per-Dataset Performance

Figure 3: The Maximum difference between per-class predictions during (Transductive-)finetuning. Results for each dataset are the average from 600 episodes. TF-MC effectively reduces the maximum difference during finetuning.

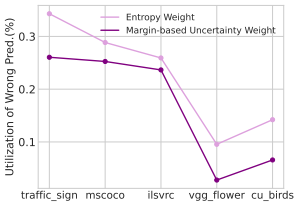


Figure 4: Comparing Utilization of Wrong Predictions with Uncertainty and Margin-based Uncertainty on Meta-Dataset. Results are averaged over 100 episodes.

backbone for finetuning. Meanwhile, we also develop TF using the same backbone. For URL (Li et al., 2021) and TSA (Li et al., 2022), we use the model published on their official github repo (ResNet-18 trained with training set from 8 datasets in Meta-Dataset.) Meanwhile, the performance of URL is done by finetuning the pa part adding before classifier following the official instruction.

B.2 On the *ImageNet-only* Evaluation

We also compare the performance of state-of-the-art methods on *ImageNet-only* evaluation on Meta-Dataset, shown in Fig. 2. "ResNet18-Proto" refers to the performance of ResNet-18 without any fine-tuning. DCMSS (Tao et al., 2022), TF (Dhillon et al., 2019) and TF-MC are further built on the same backbone of ResNet-18. As shown in Fig. 2, although DCMSS and TF could improve the per-class accuracy, the imbalanced prediction still keeps at the similar level with the backbone model. TF-MC could successfully improve accuracy further while largely reduce the class imbalance in predictions comparing with other finetuning methods like DCMSS and TF.

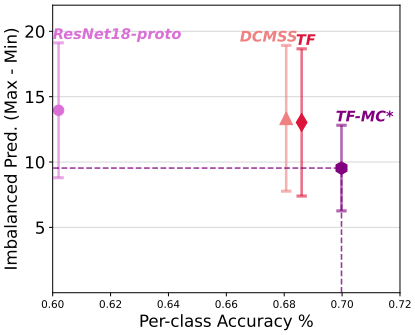


Figure 2: Imbalanced Prediction vs Per-class accuracy in *ImageNet*-only Meta-Dataset Evaluation.

B.3 The Effect of Reducing Classwise Imbalance in Predictions

We compare the change of maximum difference between per-class predictions on the query set for each iteration, shown in Fig. 3. By using TF-MC, the maximum difference decreases during finetuning. For datasets Fungi and MSCOCO, the initial accuracy on the query set are the lowest compared with the other datasets which are around 50% and 60%. The low accuracy indicates that the issue of class-imbalance in predictions could be more serve, which could explain the increase of maximum difference at the very early iterations for these two datasets. It is worth noticing that TF-MC actually effectively control the trend and decreases the maximum difference as the finetuning goes.

C Re-visiting Transductive Finetuning

In (Dhillon et al., 2019), the entropy loss is not directly applied on the feature space but on the predicted probability of base classes (the logit space). We first benchmark the performance of entropy loss directly on the feature space.

As we claimed in the main paper, there are two ways of constructing the entropy loss for unlabeled data, namely using soft-labels or using pseudo-labels. Formally, $\mathcal{L}_q(\mathbf{x})$ for the unlabeled query set is:

Method	weighting	ILSVRC	Omni	Acraft	Birds	DTD	QDraw	Fungi	Flower	Sign	COCO
soft-labels		60.19	78.76	62.71	79.22	77.6	83.33	49.99	91.82	70.54	61.55
pseudo-labels		59.19	73.71	57.56	77.53	75.63	80.83	48.18	90.14	60.42	58.82
soft-labels	✓	59.93	78.26	71.73	78.34	75.96	84.59	48.94	92.48	76.53	59.05
pseudo-labels	✓	61.49	81.64	68.88	80.23	78.55	85.2	50.72	92.67	73.96	60.09

Table 1: Ablation Results of Soft-labels and Pseudo-labels w/o Margin-based Uncertainty Weighting.

$$\mathcal{L}_q(\mathbf{x}) = \lambda H(\hat{\mathbf{y}}, p_\theta(\mathbf{y}|\mathbf{x})), \tag{11}$$

where λ denotes the loss weight and $\hat{\mathbf{y}}$ is generated from the model's own predictions on the query set. And We denote $p_\theta(\mathbf{y}|\mathbf{x})$ as the softmax probability distribution output from the model on C classes:

$$p_\theta(y = c|\mathbf{x}) = \frac{\exp z_c}{\sum_{i=1}^{C} \exp z_i}, \tag{12}$$

And $p_\theta(y|\mathbf{x}) = [p_1, p_i, ...], i \in [0, C]$.

When $\hat{\mathbf{y}} = \mathrm{argmax}(p_\theta(\mathbf{y}|\mathbf{x}))$, which is referred as pseudo-labels, and under this situation, $\mathcal{L}_q(\mathbf{x})$ is the cross-entropy loss. For a sample $(\mathbf{x}, y)$:

$$L = \lambda(-\log p_y) \tag{13}$$

When $\hat{\mathbf{y}} = p_\theta(\mathbf{y}|\mathbf{x})$, it is noted as soft-labels. And using soft-labels, $\mathcal{L}_q(\mathbf{x})$ is the entropy loss:

$$L = \lambda(-\sum_{i}^{C} p_i \log p_i) = \lambda(-p_y \log p_y - \sum_{i,i\neq y}^{C} \log p_i) \tag{14}$$

Compared with Eq. 13, the Entropy-loss in Eq. 14 can be viewed as a weighted cross-entropy loss on the ground-truth predictions ($p_y \log p_y$) with the other parts of $\sum_{i,i\neq y}^{C} \log p_i$. Specially, for $p_y \log p_y$, the predicted probability p_y serves as "the loss weight" for $\log p_y$.

As shown in Table. 1, directly using soft-labels leads to better performance of directly using pseudo-labels. However, pseudo-labels with per-sample loss weights can indeed boost performance while soft-labels with per-sample loss weights drop the performance. As illustrated above, using soft-labels in entropy loss serves as utilizing the predicted probability to weight the gradient from the ground-truth class, namely the part $p_y \log p_y$ in Eq. 14. Thus further applying the per-sample weights actually makes the gradient from the ground-truth class even smaller while the other part of the loss $\sum_{i,i\neq y}^{C} \log p_i$ weakens the information to lead the optimization towards correct predictions. While using pseudo-labels with per-sample weights, it reduces the affect of possibly wrong predictions while the cross-entropy loss gradients from the possibly correct samples still determine the optimization. This explains why pseudo-labels can work with per-sample loss weights while soft-labels itself performs strongly but are weakened by adding per-sample loss weights.

D Extended Details on Margin-based Uncertainty Weighting

Formally, for one sample we denote $\mathbf{p} = [p_1, p_2, ..., p_c]$ as the simplification of $p_\theta(\mathbf{y}|\mathbf{x})$ and, without losing generalization, we assume $p_1 \leq p_2 \leq ... \leq p_c$. We define the value difference between the maximum of probability with the others as: $\Delta p_i = p_c - p_i, i \in [1, ..., c]$. And the difference between the top-2 maximum probabilities is specifically defined as $\hat{\Delta} p$.

With a fixed p_c, the range of $\Delta\hat{p}$ relates to p_c. Specifically, the $max(\Delta\hat{p})$ happens in the situation that except the confidence (the maximum probability p_c), the other probabilities share the same value $p_1 = p_2 = ... = p_{c-1} = \frac{1-p_c}{C-1}$. And $min(\Delta\hat{p})$ is in the situation that the second maximum probability carries the value $p_{c-1} = 1 - p_c$ and the other probabilities are 0. This can be formally expressed as:

$$\begin{cases} \Delta\hat{p} \in [2p_c - 1, p_c - \frac{1-p_c}{C-1}], p_c \geq 0.5 \\ \Delta\hat{p} \in [0, p_c - \frac{1-p_c}{C-1}], p_c < 0.5 \end{cases} \tag{15}$$

The normalized entropy we introduced in the main paper is:

$$e(\mathbf{p}) = -\frac{\sum_i^c (p_i \log p_i)}{\log c} \tag{16}$$

where $\sum_i^c p_i = 1$ and c is the number of classes.

When p_c is fixed, the minimum and maximum value of $\Delta\hat{p}$ are: $(\Delta\hat{p})_{min} = p_c - (1-p_c)$, $(\Delta\hat{p})_{max} = p_c - \frac{1-p_c}{c-1}$. For $(\Delta\hat{p})_{min}$, the entropy uncertainty score is:

$$e_{(\Delta\hat{p})_{min}} = -\frac{p_c \log p_c + (1-p_c)\log(1-p_c)}{\log c} \tag{17}$$

For $(\Delta\hat{p})_{max}$, the entropy uncertainty score is:

$$\begin{aligned} e_{(\Delta p)_{max}} &= -\frac{p_c \log p_c + \sum_i^{c-1}(\frac{1-p_c}{c-1} \log \frac{1-p_c}{c-1})}{\log c} \\ &= -\frac{p_c \log p_c + (1-p_c)\log(\frac{1-p_c}{c-1})}{\log c} \\ &= -\frac{p_c \log p_c + (1-p_c)\log(1-p_c) - (1-p_c)\log(c-1)}{\log c} \\ &= e_{(\Delta p)_{min}} + \frac{(1-p_c)\log(c-1)}{\log c} \end{aligned} \tag{18}$$

Given the same confidence p_c, entropy score refers to larger uncertainty of largest margin $(\Delta\hat{p})_{max}$ compared with smallest margin $(\Delta\hat{p})_{min}$. However, $(\Delta\hat{p})_{max}$ actually refers to the largest difference between top-2 maximum probabilities that the sample is most certain to its prediction. As we discussed in the main paper, the entropy score is contradictory to the uncertainty information given by the margin. We give a theoretical view of the contradiction in the following.

Eq. 16 can be further formalized with Δp_i:

$$\begin{aligned} e(\mathbf{p}) &= -\frac{\sum_i^c (p_i \log p_i)}{\log c} \\ &= -\frac{\sum_i^c (p_c - \Delta p_i)\log(p_c - \Delta p_i)}{\log c} \\ &\geq -\frac{\sum_i^c (p_c - \Delta p_i)\log p_c}{\log c} = -\frac{\log p_c \sum_i^c (p_c - \Delta p_i)}{\log c} \end{aligned} \tag{19}$$

The importance of margin $\hat{\Delta}p$ is weaken by adding $p_c - \Delta p_i$. This is supported by the empirical experimental results of utilizing top-k probabilities in the entropy-related weights.

E Extended Details on the Optimization using Entropy Loss

And the loss $\mathcal{L}_q(\mathbf{x})$ for the unlabeled query set is constructed similarly:

$$\mathcal{L}_q(\mathbf{x}) = \lambda H(\mathbf{p}_\theta(\mathbf{y}|\mathbf{x}), \mathbf{p}_\theta(\mathbf{y}|\mathbf{x})) = \lambda \mathbf{p}_\theta(\mathbf{y}|\mathbf{x}) \log(\mathbf{p}_\theta(\mathbf{y}|\mathbf{x})), \tag{20}$$

where λ denotes the *per-sample* loss weight. $\lambda = 1$ in the following discussion.

We denote $\mathbf{p}_\theta(\mathbf{y}|\mathbf{x})$ as the categorical probabilities on C classes which is the output from the softmax layer in the model:

$$p_\theta(y = c|\mathbf{x}) = \frac{\exp z_c}{\sum_{i=1}^C \exp z_i}, \tag{21}$$

where $z_i = \langle \omega_i, f_\theta(\mathbf{x}) \rangle, i \in C$, is the logit for class i.

For one sample $\mathbf{x} \in \mathcal{D}_q$, the gradient for feature $f_\theta(\mathbf{x})$ from the entropy loss is:

$$\frac{\partial \mathcal{L}_q}{\partial f_\theta(\mathbf{x})} = \sum_i^C \frac{\partial \mathcal{L}_q}{\partial z_i} \omega_i \tag{22}$$

And for each $\frac{\partial \mathcal{L}_q}{\partial z_i}$, $i \in C$:

$$\frac{\partial \mathcal{L}_q}{\partial z_i} = \frac{\partial \mathcal{L}_q}{\partial p_i} \frac{\partial p_i}{\partial z_i} \tag{23}$$

And for entropy loss (not cross-entropy loss) soft-label is used with no gradients on the label part:

$$\frac{\partial \mathcal{L}_q}{\partial p_i} = -p_i \frac{1}{p_i} \tag{24}$$

$$\frac{\partial p_i}{\partial z_i} = p_i(1 - p_i) \tag{25}$$

$$\frac{\partial \mathcal{L}_q}{\partial z_i} \Rightarrow -p_i(1 - p_i) \tag{26}$$

$p_i = p_\theta(i|\mathbf{x})$ for simple notations. By summing over all samples, the gradient on z_i, $i \in C$ is:

$$\frac{\partial \mathcal{L}_q}{\partial z_i} \Rightarrow -\frac{1}{N_q} \sum_j^{N_q} p_{ij}(1 - p_{ij}) \tag{27}$$

F Possible Questions

Q1. Will the usage of Uniform testing set in the observation lead to the imbalanced predictions?: Using uniform testing distribution ensures that the testing distribution will not affect the quantification of pre-class predictions. Meanwhile, the imbalanced prediction would be more severe when the test distribution is non-uniform. The observation in Figure 2 (before TF-MC) shows that there are some classes obtaining much fewer predictions than others. If a testing scenario is constructed by samples from the class of the least number of predictions (2 predictions for 10 testing samples in Figure 2 (before TF-MC)), the accuracy is upper-bounded by the number of predictions (0.2).

Q2. Will the Uniform used in Class-balanced Normalization degrades the performance if the query set is not balanced? The uniform prior in CN is a strong regularization to align the learned marginal probability. To avoid its effect on regularizing the marginal distribution of the whole testing set to be uniform, we designed to compute the learned marginal probability by using each sample from the query set with the whole support set. Meanwhile, we verify that this design effectively makes the uniform regularization also works well when the query set is not balanced (the stochastic setting in Appendix.A.

Q3. A more balanced prediction is not equal to a higher accuracy: As the practical testing environment could involve different data distributions, solving class-imbalanced predictions would make the algorithm more robust to different testing scenarios. For example: if all images from the testing set are from those classes with least predictions (e.g. 2 predictions for 10 testing samples), the accuracy is upper-bounded by the number of predictions (only 0.2). In this case, improving class-imbalanced predictions is beneficial to improve accuracy. Meanwhile, TF-MC encourages a more balanced prediction during finetuning which actively guides the model to learn classes fairly, and improving the model training is expected to improve the accuracy. The experimental results well support that by solving the class-imbalanced predictions through TF-MC, our method brings a consistent accuracy boost over datasets from different domains(2.39 $\%$ on average, Table.1 main paper) and different shots (Figure.3(c)) compared with inductive fine-tuning.

REFERENCES

David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019.

Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019.

Wei-Hong Li, Xialei Liu, and Hakan Bilen. Universal representation learning from multiple domains for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9526–9535, 2021.

Weihong Li, Xialei Liu, and Hakan Bilen. Cross-domain few-shot learning with task-specific adapters. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition 2022*, 2022.

Ran Tao, Han Zhang, Yutong Zheng, and Marios Savvides. Powering finetuning in few-shot learning: Domain-agnostic feature adaptation with rectified class prototypes. *arXiv preprint arXiv:2204.03749*, 2022.