Appendix: Boosting Transductive Few-Shot Fine-tuning with Margin-based Uncertainty Weighting and Probability Regularization

Ran Tao
Carnegie Mellon University
taoran1@cmu.edu

Hao Chen
Carnegie Mellon University
haoc3@andrew.cmu.edu

Marios Savvides
Carnegie Mellon University
marioss@andrew.cmu.edu

1. Extended Discussion on Probability Regularization

In this section, we give a comprehensive overview on the difference between probability regularization and Distribution Alignment.

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 from the support set and the query set as:

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

$$\hat{E}_{\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})] = \frac{1}{N_q}\sum_{\mathbf{x}\in\mathcal{D}_q} p_\theta(\mathbf{y}|\mathbf{x}) \tag{2}$$

And if considering all available data:

$$\begin{aligned}\hat{E}_{\mathcal{D}_s\cup\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})] = &\frac{N_s}{N_s+N_q}\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})] \\ &+ \frac{N_q}{N_q+N_s}\hat{E}_{\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})]\end{aligned} \tag{3}$$

The previous work [1] in semi-supervised learning proposes a formulation to conduct distribution alignment as:

$$\tilde{\mathbf{q}} = \text{Normalize}(\mathbf{q}\frac{p(\mathbf{y})}{\hat{p}(\mathbf{y})}), \tag{4}$$

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 an unlabeled sample. And $\text{Normalize}(x_i) = \frac{x_i}{\sum_j x_j}$.

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 equals the original DA in [1], 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 as 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:

$$\hat{E}_{\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})] \sim \frac{1}{N_q}\sum_{\mathbf{x}\in\mathcal{D}_q}\frac{p(\mathbf{y})}{\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]}p_\theta(\mathbf{y}|\mathbf{x}) \tag{5}$$

The 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:

$$\hat{E}_{\mathcal{D}_s\cup\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})] \sim \frac{N_s}{N_s+N_q}\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})] + \frac{N_q}{N_q+N_s}p(\mathbf{y}) \tag{6}$$

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 probability regularization, 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.

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

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

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

$$\begin{aligned}\hat{E}_{x_q\cup\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})] &= \frac{1}{1+N_s}[p_\theta(\mathbf{y}|\mathbf{x}_q) + \sum_{x\in\cup\mathcal{D}_s} p_\theta(\mathbf{y}|\mathbf{x})] \\ &= \frac{p_\theta(\mathbf{y}|\mathbf{x}_q)}{1+N_s} + \frac{N_s}{1+N_s}\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]\end{aligned} \quad (8)$$

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

$$\begin{aligned}\hat{E}_{\mathcal{D}_s\cup\mathcal{D}_q}[p_\theta(\mathbf{y}|\mathbf{x})] &\sim \frac{N_s}{N_s+N_q}\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})] \\ &+ \frac{1}{N_q+N_s}\sum_{\mathbf{x}\in\mathcal{D}_q}\frac{\mathcal{U}}{\hat{E}_{x\cup\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]}p_\theta(\mathbf{y}|\mathbf{x}) \\ &\sim \frac{N_s}{N_s+N_q}\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})] \\ &- \frac{1+N_s}{N_q+N_s}\sum_{x\in D_q}\frac{\hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]}{p_\theta(\mathbf{y}|\mathbf{x}) + \hat{E}_{\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]} \\ &+ \frac{1+N_s}{N_q+N_s}\hat{E}_{x_q\cup\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]\end{aligned} \quad (9)$$

By using probability regularization, $\hat{E}_{x\cup\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]$ 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.

2. Re-visiting Transductive Finetuning

In [2], the entropy loss is not directly applied to the feature space but to 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:

$$\mathcal{L}_q(\mathbf{x}) = \lambda H(\hat{\mathbf{y}}, p_\theta(\mathbf{y}|\mathbf{x})), \quad (10)$$

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:

$$p_\theta(y=c|\mathbf{x}) = \frac{\exp z_c}{\sum_{i=1}^{C}\exp z_i}, \quad (11)$$

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

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

$$L = \lambda(-\log p_y) \quad (12)$$

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:

$$L = \lambda(-\sum_{i}^{C} p_i \log p_i) = \lambda(-p_y \log p_y - \sum_{i,i\neq y}^{C}\log p_i) \quad (13)$$

Compared with Eq. 12, the Entropy-loss in Eq. 13 can be viewed as a weighted cross-entropy loss on the ground-truth prediction ($p_y \log p_y$) with the other parts of $\sum_{i,i\neq y}^{C}\log p_i$. Especially 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 than 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. 13. 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. Using pseudo-labels with per-sample weights reduces the effect 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 themselves perform strongly but are weakened by adding per-sample loss weights.

3. 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 probability and 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 $\hat{\Delta} p$.

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

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

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:

$$\begin{cases} \Delta\hat{p} \in [2p_c - 1, p_c - \frac{1-p_c}{C-1}], p_c \geq 0.5 \\ \Delta\hat{p} \in [0, p_c - \frac{1-p_c}{C-1}], p_c < 0.5 \end{cases} \tag{14}$$

The normalized entropy we introduced in the main paper is:

$$e(\mathbf{p}) = -\frac{\sum_i^c (p_i \log p_i)}{\log c} \tag{15}$$

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:

$$e_{(\Delta\hat{p})_{min}} = -\frac{p_c \log p_c + (1 - p_c)\log(1 - p_c)}{\log c} \tag{16}$$

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

$$\begin{aligned} e_{(\Delta p)_{max}} &= -\frac{p_c \log p_c + \sum_i^{c-1}(\frac{1-p_c}{c-1} \log \frac{1-p_c}{c-1})}{\log c} \\ &= -\frac{p_c \log p_c + (1 - p_c)\log(\frac{1-p_c}{c-1})}{\log c} \\ &= e_{(\Delta p)_{min}} + \frac{(1 - p_c)\log(c - 1)}{\log c} \end{aligned} \tag{17}$$

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. 15 can be further formalized with Δp_i:

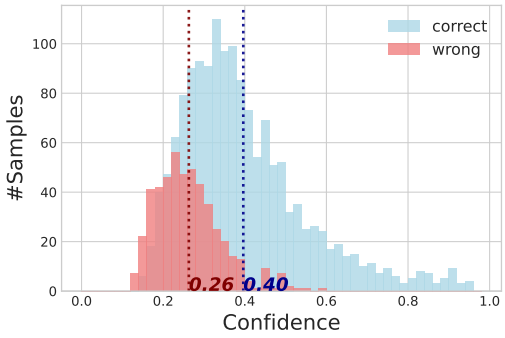
$$\begin{aligned} e(\mathbf{p}) &= -\frac{\sum_i^c (p_i \log p_i)}{\log c} \\ &= -\frac{\sum_i^c (p_c - \Delta p_i)\log(p_c - \Delta p_i)}{\log c} \\ &\geq -\frac{\sum_i^c (p_c - \Delta p_i)\log p_c}{\log c} = -\frac{\log p_c \sum_i^c (p_c - \Delta p_i)}{\log c} \end{aligned} \tag{18}$$

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

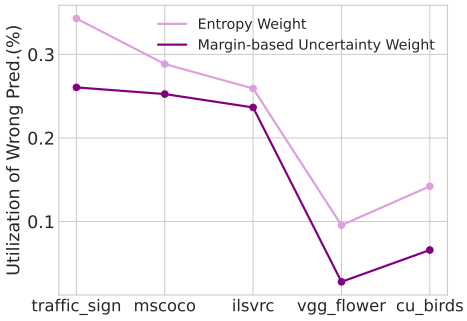
4. Extended Illustration

Q1. Will the usage of Uniform testing set in the observation lead to 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-MP) shows that there are some classes obtain 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-MP)), the accuracy is upper-bounded by the number of predictions (0.2).

Q2. A more balanced prediction does not equal 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 the 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-MP encourages a more balanced prediction during fine-tuning, actively guiding the model to learn classes fairly. Improving the model training is expected to improve the accuracy. The experimental results well support that by solving the class-imbalanced predictions through TF-MP, our method brings a consistent accuracy boost over datasets from different domains(2.39 % on average, Table.1 main paper) and different shots compared with inductive fine-tuning.



(a) Confidence Distribution.



(b) Utilization of Wrong Pred.

Figure 1. (a). The confidence distribution of correct and wrong predictions on the testing set. Results are averaged using 100 episodes for each dataset in Meta-Dataset. For samples with correct predictions, the average confidence is only 0.4, which makes it incapable of applying a hard threshold of confidence to select unlabeled data during transductive finetuning as used in FixMatch [3]. (b). Utilization of wrong predictions using Entropy loss weights and Margin-Based uncertainty loss weights during transductive-finetuning. Compared with entropy loss weights, utilizing margin-based uncertainty weights could largely reduce the utilization of wrong predictions for different datasets. Results are averaged over 600 episodes.

References

[1] 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. 1

[2] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019. 2

[3] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. 4