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

Abstract

Few-Shot Learning (FSL) has been rapidly developed in recent years, potentially eliminating the requirement for significant data acquisition. Few-shot fine-tuning has been demonstrated to be practically efficient and helpful, especially for out-of-distribution datum [7, 13, 17, 29]. In this work, we first observe that the few-shot fine-tuned methods are learned with the imbalanced class marginal distribution, leading to imbalanced per-class testing accuracy. This observation further motivates us to propose the Transductive Fine-tuning with Margin-based uncertainty weighting and Probability regularization (TF-MP), which learns a more balanced class marginal distribution as shown in Fig. 1. We first conduct sample weighting on unlabeled testing data with margin-based uncertainty scores and further regularize each test sample’s categorical probability. TF-MP achieves state-of-the-art performance on in- / out-of-distribution evaluations of Meta-Dataset [31] and surpasses previous transductive methods by a large margin.

1. Introduction

Deep learning has gained vital progress in various architecture designs, optimization techniques, data augmentation, and learning strategies, demonstrating its great potential to be applied to real-world scenarios. However, applications with deep learning generally require a large amount of labeled data, which is time-consuming to collect and costly on manual labeling force. Few-Shot Learning (FSL), learning with only a few training samples, becomes increasingly essential [5, 9, 10, 27, 31, 33] to alleviate the dependence on data acquisition significantly.

The recent attention on FSL over out-of-distribution datum [31] poses a challenge in obtaining efficient algorithms that can perform well on cross-domain situations. Fine-tuning a pre-trained feature extractor with a few samples [5, 7, 13, 17, 29] recently demonstrates its prominent potential to solve this challenge. However, as illustrated in [29], a few training samples would lead to a biased estimation of the true data distribution. The biased learning during few-shot fine-tuning could further mislead the model to learn an imbalanced class marginal distribution. To verify this, we quantify the largest difference (LD) between the number of per-class predictions with a uniform testing set. If the fine-tuned model learns a balanced class marginal distribution, with a uniform testing set LD should approach zero. However, the empirical results show the opposite answer. As shown in Fig. 1, even with state-of-the-art methods [7, 17, 18, 29], LD could be largely over 10 in practice.

Figure 1. We observe that fine-tuned models with current state-of-the-art methods [7, 17, 18, 29] learned an imbalanced class marginal distribution. In the empirical experiments, a uniform testing set is utilized, and the Largest Difference LD between per-class predictions is used to quantify whether the learned class marginal probability is balanced. Data are from sub-datasets in Meta-Dataset [31] with 100 episodes for each dataset and 10 per-class testing samples. With current methods, LD is over 10. TF-MP successfully reduces LD by around 5 points and achieves the best per-class accuracy.
The observation in Fig. 1 demonstrates that the fine-tuned models in FSL suffer from severely imbalanced categorical performance. In other words, the learned class marginal distribution of few-shot fine-tuned models is largely imbalanced and biased. We argue that solving this issue is critical to maintaining the algorithm’s robustness to different testing scenarios. Classes with fewer predictions would carry low accuracy, and this issue of fine-tuned models could yield a fatal failure for testing scenarios in favor of these classes.

In this work, we revisit Transductive Fine-tuning [7] by effectively using unlabelled testing data. Based on the aforementioned analysis, the imbalanced categorical performance in FSL motivates us to propose two solutions: (1) the per-sample loss weighting through Margin-based uncertainty and (2) the probability regularization. For (1), as shown in Fig. 2, using the same number of per-class training data achieves extremely imbalanced prediction results. It indicates that each sample contributes to the final performance differently, which inspires us to weigh the unlabeled testing samples according to their uncertainty scores. Specifically, we address the importance of utilizing margin [26] in entropy computation and demonstrate its supreme ability to compress the utilization of wrong predictions. For (2), as the ideal performance should be categorically balanced, we propose to explicitly regularize the probability for each testing data. Precisely, each testing sample’s categorical probability is adjusted by a scale vector, which quantifies the difference between the class marginal distribution and the uniform. The class marginal distribution is estimated by combining each query sample with the complete support set. Our proposed Transductive Fine-tuning with Margin-based uncertainty and Probability regularization (TF-MP) effectively reduces the largest difference between per-class predictions by around 5 samples and further improves per-class accuracy with 2.1%, shown in Fig. 1. Meanwhile, TF-MP shows robust cross-domain performance boosts on Meta-Dataset, demonstrating its potential in real applications. Our contributions can be summarized as follows:

- We present the observation that: with current state-of-the-art methods [7, 17, 18, 29], the few-shot fine-tuned models are learned with the imbalanced class marginal distribution, which in other words presents imbalanced per-class accuracy. We highlight the importance of solving it to improve the model’s robustness under different testing scenarios.

- Inspired by the observation, we revisit Transductive Fine-tuning and propose TF-MP: (1) Utilizing Margin-based uncertainty to weigh each unlabeled testing data in the loss objective in order to compress the utilization of possibly wrong predictions. (2) Regularizing the categorical probability for each testing sample to pursue a more balanced class marginal during finetuning.

- We empirically verify that models with TF-MP learn a more balanced class marginal distribution as shown in Fig. 1. Furthermore, we conduct comprehensive experiments to show that TF-MP obtains consistent performance boosts on Meta-Dataset [31], demonstrating its efficiency and effectiveness for practical applications.
2. Related Work

Transductive Few-Shot Learning: Transductive few-shot learning uses the unlabeled query set (testing images) along with the support set (training images) to make up for the lack of training data. [21] updates parameters of batch normalization layers using unlabeled query samples. [20] propagates labels for unseen classes through episodic meta-learning and [1] presents the label refinement with a Mahalanobis-distance based classifier. TIM [3] designs a loss to encourage the marginal distribution of the query set to be uniform, and pseudo-labels are directly used without compressing the possibly wrong predictions. α-TIM [32] addresses creating different testing distributions to reflect real-world scenarios better and proposes to enhance TIM [3] by α—convergence. [14] uses the Optimal Transport Algorithm (OTA) for pseudo-label mapping with entropy minimization on the OTA-based mapping. [19] computes a linear projection space on features for each task when utilizing an alternating normalization [16] target at post-processing predictions. In [7], a transductive framework is firstly proposed to involve the testing images during fine-tuning. [7] builds the classification upon predicted logits other than directly on features. Previous works on transductive few-shot learning ignore compressing the utilization of wrong predictions.

Semi-Supervised Learning: Semi-Supervised Learning (SSL) is designed to introduce extra unlabelled data into the training set, which differs from the transductive methods that utilize unlabeled testing samples. Suppressing the influence of possible wrong predictions is also an essential task in SSL. There are methods like assigning per-sample loss weights using entropy-measured probability uncertainty [15], selecting samples with a strictly high confidence threshold [28], and designing functions to adaptive assign loss weights [4]. We compare our margin-based uncertainty weighting with entropy-based weighting thoroughly in Sec. 3. Few-shot learning limits testing samples to have very high confidence, which makes the handcraft high-confidence threshold inapplicable in FSL.

Probability Alignment: Confidence calibration in [12] and Alternating normalization [16] target at post-processing evaluations. [12] calibrates the overall confidence distribution with the true correctness likelihood. [16] normalizes the probability for unconfident samples with the prior distribution of confident samples through multiple steps. Our work proposes probability regularization to adjust probabilities for each testing sample on the fly during fine-tuning. Distribution Alignment in [2] is designed to match the predicted marginal distribution of unlabeled data with the marginal distribution of labeled data.

3. Method

In this section, we first introduce the transductive fine-tuning framework and further discuss the TF-MP.

3.1. Revisiting Transductive Fine-tuning

Firstly, we formally describe the terminology and episode setting in FSL. For one episode in FSL, the training and testing set is referred to as the support and query set, respectively. Let \( (x, y) \) denote the pair of an input \( x \) with its ground-truth one-hot label \( y \in \mathbb{R}^C \), where \( C \) is the number of classes. The support set is then represented as \( D_s = \{(x_i, y_i)\}_{i=1}^{N_s} \). The query set is denoted as \( D_q = \{(x_i)\}_{i=1}^{N_q} \) where the ground-truth labels are unknown if used in a transductive manner; \( N_s \) and \( N_q \) are the total number of samples in support set and query set, respectively.

A feature extractor \( f_\theta \) is firstly pre-trained on the meta-training set, and transductive fine-tuning is conducted on the meta-test set within each episode. We denote \( p_0(y|x) \) as the categorical probabilities on \( C \) classes, which is the output from the softmax layer in the model:

\[
p_0(y = c|x) = \frac{e^{z_c}}{\sum_{c=1}^{C} e^{z_c}}, \tag{1}
\]

where \( z_i = \langle \omega_i, f_\theta(x) \rangle, i \in C \), the dot-product between \( \omega_i \) and \( f_\theta(x) \), is the logit for class \( i \). As widely used in [6, 17, 22, 27, 29], \( \omega_i \) is the novel class prototype that is initialized as the mean feature from the support set \( D_s \) for every iteration. A model with parameter \( \theta \) is learnt to classify \( D_s \) and \( D_q \) as measured by the following criterion:

\[
\theta^* = \arg \min_{\theta} \left( \frac{1}{N_s} \sum_{(x, y) \in D_s} L_s(x, y) + \frac{1}{N_q} \sum_{(x) \in D_q} L_q(x) \right) \tag{2}
\]

The loss \( L_s(x, y) \) for the labeled support set is the cross-entropy loss. And the loss \( L_q(x) \) for the unlabeled query set is constructed as entropy minimization:

\[
L_q(x) = \lambda(p_0(y|x)) \times H(p_0(y|x)) \tag{3}
\]

where \( \lambda \) denotes the per-sample loss weight. And \( H(p_0(y|x)) = -p_0(y|x) \log(p_0(y|x)) \) is the entropy loss. Specifically, the entropy loss for unsupervised data can be generally represented as \( H(p_0(y|x)) = -\hat{y} \log(p_0(y|x)) \). As widely used in semi-supervised learning [11, 28, 34], there are two types of \( \hat{y} \): when \( \hat{y} = \arg \max(p_0(y|x)) \), we refer it as pseudo-label, whereas when \( \hat{y} = p_0(y|x) \), it is noted as soft-label.
In the previous work of transductive fine-tuning [7], soft-label is utilized with \( \lambda = 1 \) for every testing image, and the entropy minimization is conducted on the logit space. Different from [7], we directly optimize \( \mathcal{L}_q(x) \) on the feature space and design \( \lambda(p)(y|x) \) to compress the utilization of wrong predictions and probability regularization is applied on \( p(y|x) \) before forwarding it to \( H(p(y|x)) \).

### 3.2. Margin-based Uncertainty Weighting

Margin-based uncertainty is designed to assign low loss weights for wrongly predicted samples and high loss weights for the correct ones. In this section, we first discuss a neglected fact that the generally used entropy-based weighting may not truly reflect whether the sample has the wrong prediction. Furthermore, we propose margin-based uncertainty weighting to compress the utilization of wrongly predicted testing data.

The class with the maximum probability \( p_{max} \) is assigned as the predicted class. Thus \( p_{max} \) is referred to as confidence [12], which indicates the confidence level of the categorical prediction. The other index used to indicate the confidence level of the prediction is the entropy of the predicted probabilities. In semi-supervised learning [15], entropy-based per-sample loss weight is used as:

\[
\lambda(p) = 1 - e(p) \tag{4}
\]

where \( p \) is the abbreviation for \( p(y|x) \). And \( e(p) \) refers to the normalized entropy:

\[
e(p) = \frac{-\sum_{i=1}^{C} (p_i \log p_i)}{\log C} \tag{5}
\]

where \( \sum_{i} p_i = 1 \), \( p = [p_1, p_2, ..., p_C] \) and \( C \) is the number of classes. \( e(p) \) is normalized to \([0, 1]\) as the entropy \( \sum_{i} (p_i \log p_i) \) is scaled by its maximum value \( \log C \). Entropy on \( p(y|x) \) quantifies the uncertainty of probabilities. Larger uncertainty generally refers to a lower confidence level the sample carries towards its class prediction, consequently leading to lower loss weight \( \lambda(p) \). However, when diving into Eq. 5, we discover that the uncertainty on the whole probability distribution may not be ideal for distinguishing whether the predictions are wrong.

Intuitively, wrong predictions are more likely to be made when the model produces similar probabilities between two classes. In other words, the margin between the maximum and second maximum probability \( \Delta p \) can largely reflect how uncertain an example is with its prediction. A smaller margin indicates larger uncertainty on the prediction, which indicates a higher possibility that the prediction is wrong [26].

We further analyze how margin information is reflected in the entropy-based uncertainty measurement. When \( p_{max} \) is fixed, margin \( \Delta p \) is in the range of: \( \min(\Delta p) = p_{max} - (1-p_{max}) \), \( \max(\Delta p) = p_{max} - \frac{1-p_{max}}{c-1} \). Samples with the largest margin \( (\Delta p)_{max} \) are expected to be assigned with the least uncertainty on decisions. However, the entropy score gives the opposite answer. For \( \max(\Delta p) \), the entropy is:

\[
e_{\max}(\Delta p) = e_{\min}(\Delta p) + \frac{(1-p_{max}) \log(c-1)}{\log c} \tag{6}
\]

As \( \frac{(1-p_{max}) \log(c-1)}{\log c} \) is non-negative, Eq. 6 reveals that samples with largest margin \( \max(\Delta p) \) carry larger entropy-based uncertainty scores than samples with \( \min(\Delta p) \), which is contradictory to the information implied by the margin.

To solve this contradiction, we address the importance of only using top-2 probabilities in Eq. 5. The maximum and second maximum probabilities are first normalized by dividing the sum to satisfy the requirement of \( \sum_{i} p_i = 1 \) in Eq. 5. We refer \( \hat{p}_{max} \) and \( \hat{\Delta p} \) as the normalized results, which is further used in Eq. 7. The margin-based uncertainty is defined as:
This simple modification can unify the information carried by confidence, margin, and entropy. When margin $\Delta p$ is fixed, $\hat{e}(p)$ is non-decreasing with confidence $p_{\text{max}}$; when confidence $p_{\text{max}}$ is fixed, $\hat{e}(p)$ is as well non-decreasing with $\Delta p$. In doing so, the margin-based entropy score could consistently reflect the confidence level $p_{\text{max}}$ as well as the margin $\Delta p$, as shown in Fig. 3. By focusing on the uncertainty delivered by the margin in $p$, it achieves more substantial compression on utilization of wrong predictions compared with entropy-based loss weights.

### 3.3. Probability Regularization

As illustrated in Sec. 1, the learned class marginal distribution from a few-shot fine-tuned model is severely imbalanced. Motivated by this, We emphasize the importance of explicitly regularizing the categorical probability for each testing sample, as introduced in the following. The probability regularization is explicitly conducted on the predicted probability $p(y|x)$ for each testing data. Firstly with $x \in D_p$, the learned class marginal distribution is estimated using the set $x \cup D_s$, which is constructed by combining each testing data with the whole support set. And a unique scale vector $v \in R^C$ is obtained for each testing sample by aligning the estimated marginal probability with a uniform prior:

$$v = \frac{U}{\sum_{x \cup D_s} p_{\theta}(y|x)}$$

where $U \in R^C$ represents the uniform prior and the scale vector $v$ quantities the difference between estimated marginal distribution with the uniform prior. Furthermore, $v$ is used to conduct probability regularization on $q = p_{\theta}(y|x)$ as:

$$\hat{q} = \text{Normalize}(q \ast v)$$

where $\text{Normalize}(x_i) = \frac{x_i}{\sum_j x_j}$ and $\ast$ denotes the element-wise multiplication. $q \ast v$ applies re-scaling on $q$ to reduce the difference between estimated marginal distribution with the uniform prior.

In doing so, each sample from the query set obtains a unique scale vector $v$, which allows per-sample probability regularization. Meanwhile, aligning the estimated marginal probability of $x \cup D_s$ to Uniform avoids direct regularization on the class marginal probability of the whole query set. This allows the probability regularization to be theoretically effective when the actual testing set is not uniform. Last but not least, the uniform prior serves as a solid regularization to enforce the class balance during fine-tuning. We conduct comprehensive ablation experiments to verify the design’s effectiveness under different testing scenarios in Sec. 4.

### 4. Experimental Validation

In this section, we first conduct comprehensive ablation experiments to verify the effectiveness of margin-based uncertainty weighting and probability regularization. Meanwhile, we highlight the essential role of transductive fine-tuning for extreme few-shot cases compared with purely fine-tuning. Furthermore, we evaluate and compare our results with the latest techniques on Meta-Dataset [31] Imagenet-only and All-datasets evaluations.

### 4.1. Implementation Details

**A Briefing on Datasets.** We evaluate our method on Meta-Dataset [31], which is so far the most comprehensive benchmark for few-shot learning composed of multiple existing datasets in different domains. More specifically, there are two evaluation protocols in Meta-Dataset. The in-distribution evaluation, referred to as All-datasets evaluation, allows using available training sets from 8 of 10 datasets, and the out-of-distribution evaluation, referred to as Imagenet-only evaluation, allows only using the training set from ILSVRC-2012 [24].

<table>
<thead>
<tr>
<th>Method</th>
<th>PR</th>
<th>MW</th>
<th>ILSVRC</th>
<th>Omni</th>
<th>Acraft</th>
<th>Birds</th>
<th>DTD</th>
<th>QDraw</th>
<th>Fungi</th>
<th>Flower</th>
<th>Sign</th>
<th>COCO</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td></td>
<td></td>
<td>58.57</td>
<td>68.13</td>
<td>53.32</td>
<td>76.55</td>
<td>74.38</td>
<td>57.03</td>
<td>44.34</td>
<td>89.11</td>
<td>49.61</td>
<td>56.4</td>
</tr>
<tr>
<td>TF</td>
<td>✓</td>
<td></td>
<td>59.96</td>
<td>78.7</td>
<td>72.32</td>
<td>78.30</td>
<td>76.96</td>
<td>68.11</td>
<td>47.51</td>
<td>91.95</td>
<td>76.39</td>
<td>57.32</td>
</tr>
<tr>
<td>TF</td>
<td>✓</td>
<td>✓</td>
<td>59.19</td>
<td>73.71</td>
<td>57.56</td>
<td>77.53</td>
<td>75.63</td>
<td>66.42</td>
<td>48.18</td>
<td>90.14</td>
<td>60.42</td>
<td>58.82</td>
</tr>
<tr>
<td>TF</td>
<td>✓</td>
<td>✓</td>
<td>59.63</td>
<td>74.58</td>
<td>56.45</td>
<td>76.07</td>
<td>75.92</td>
<td>67.5</td>
<td>46.04</td>
<td>91.26</td>
<td>62.21</td>
<td>59.17</td>
</tr>
<tr>
<td>TF</td>
<td>✓</td>
<td>✓</td>
<td>61.49</td>
<td>81.64</td>
<td>68.88</td>
<td>80.23</td>
<td>78.55</td>
<td>69.29</td>
<td>50.72</td>
<td>92.67</td>
<td>73.96</td>
<td>60.09</td>
</tr>
<tr>
<td>TF</td>
<td>✓</td>
<td>✓</td>
<td>62.18</td>
<td>83.78</td>
<td>70.9</td>
<td>81.25</td>
<td>79.15</td>
<td>70.5</td>
<td>51.17</td>
<td>93.3</td>
<td>78.23</td>
<td>62.46</td>
</tr>
</tbody>
</table>

Table 1. Ablation studies using ResNet18. Results are reported using an average of 600 episodes. The first row corresponds to the performance of the Proto-classifier. Fine-tuning (F) the backbone is first evaluated. Transductive Finetune (TF), Margin-based Uncertainty Weighting (MW) and probability regularization (PR) separately or combined are verified. TF with MW and PR achieve the best results in the ablation study.

\[
\hat{e}(p) = -\frac{1}{\log 2} (\hat{p}_{\text{max}} \log \hat{p}_{\text{max}} \\
+ (\hat{p}_{\text{max}} - \Delta p) \log(\hat{p}_{\text{max}} - \Delta p))
\]

\[
(7)
\]
Pre-training the Backbone: Choice of the Network and Training Setting. For Imagenet-only evaluation, the ILSVRC-2012 [23] in Meta-Dataset is split into 712 training, 158 validation, and 130 test classes. We use the training set of 712 classes to train two feature extractors with backbones: ResNet18 and ResNet34. For All-Datasets evaluation, Traffic-sign and MSCOCO are excluded from training, and the training sets from the other datasets in Meta-Dataset are used in (pre)-training the feature extractor. We use the same ResNet18 in [17, 18] as the feature extractor for All-Datasets evaluation. For ResNet18, we follow the same protocol in [6], which is: the images are randomly resized and cropped to 128x128, horizontally flipped, and normalized. For ResNet34, we follow the same structure modification in [8] which uses stride 1 and dilated convolution for the last residual block, and the input image size is 224x224. For the training of feature extractors, we use the same setting: the initial learning rate is set to 0.1 with 1e-4 weight decay and decreases by a factor of 0.1 every 30 epochs with a total of 90 epochs. Both models are trained using the SGD optimizer with batch size 256.

Setting of Evaluation and Fine-tuning: The general evaluation on Meta-Dataset utilizes a flexible sampling of episodes [31], which allows a maximum of 500 images in the support set in one episode. Data argumentation works as resizing and center cropping images to 128x128 (ResNet18) and 224x224 (ResNet34) followed by normalization. We follow the same fine-tuning setting in [7, 29]: learning rate of 5e-5, Adam optimizer, and 25 total episodes. We follow the same metrics in meta-Baseline [6]: for fine-tuning on the (Meta-)test set, features and class prototypes are under normalization for the softmax cross-entropy loss. The temperature in the loss function is initialized to 10. Experiments of fine-tuning run on 1 P6000 GPU. For All-Datasets evaluation, we evaluate our method upon the latest technique TSA [17] and follow the same fine-tuning setting with adadelta but only extend the iterations from 40 to 60. We explain some abbreviations used in this section.

Proto-classifier refers to purely evaluating the (pre-)trained feature extractor with average feature initialized classifier, which is the same evaluation in [6]. Finetune refers to only using the support set to finetune the (pre-)trained feature extractor. TF-MP refers to our methods.

4.2. Ablation Studies

All of our ablation results are based on Imagenet-only evaluation in Meta-Dataset, which only uses the ILSVRC-2012 training set to pre-train the feature extractor ResNet-18.

### 4.2.1 Studies on Margin-based Uncertainty

Margin-based uncertainty weighting shows domain-agnostic performance boosts over all datasets. In Table 1, adding MW with TF helps to compensate for the performance loss using TF. Compared with only TF, adding MW brings performance boosts from 1.27% on MSCOCO to 13.54% on the Traffic sign. Meanwhile, TF with MW surpasses fine-tuning over 7 out of 10 datasets with the most performance gains from 0.72% on VGG-flower to 3.21% on Fungi. This demonstrates the importance of down-weighting samples with the wrong prediction during transductive fine-tuning. The consistent performance gains
Table 2. Results on ImageNet-only evaluation of Meta-Dataset. We provide the statistical results with a 95% confidence interval over 600 episodes. TF-MP brings consistent performance improvements over all ten datasets compared with recent works.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>ILSVRC</th>
<th>Omn</th>
<th>Acr</th>
<th>Birds</th>
<th>DTD</th>
<th>QDraw</th>
<th>Fungi</th>
<th>Flower</th>
<th>Sign</th>
<th>COCO</th>
</tr>
</thead>
<tbody>
<tr>
<td>fT-M [21]</td>
<td>-</td>
<td>51.9 ± 1.1</td>
<td>67.6 ± 1.2</td>
<td>54.1 ± 0.9</td>
<td>70.7 ± 0.9</td>
<td>68.3 ± 0.8</td>
<td>50.3 ± 1.0</td>
<td>41.4 ± 1.1</td>
<td>87.3 ± 0.6</td>
<td>51.8 ± 1.0</td>
<td>48.0 ± 1.0</td>
</tr>
<tr>
<td>BOHB [25]</td>
<td>ResNet18</td>
<td>60.1</td>
<td>64.9</td>
<td>63.1</td>
<td>77.7</td>
<td>78.6</td>
<td>62.5</td>
<td>47.1</td>
<td>91.6</td>
<td>77.5</td>
<td>57.0</td>
</tr>
<tr>
<td>LR [30]</td>
<td>Meta-B [6]</td>
<td>ResNet18</td>
<td>59.2</td>
<td>69.1</td>
<td>54.1</td>
<td>77.3</td>
<td>76.0</td>
<td>57.3</td>
<td>45.4</td>
<td>89.6</td>
<td>66.2</td>
</tr>
<tr>
<td>Meta-B [6]</td>
<td>ResNet18</td>
<td>54.8</td>
<td>62.0</td>
<td>49.2</td>
<td>66.5</td>
<td>71.6</td>
<td>56.6</td>
<td>37.5</td>
<td>82.1</td>
<td>63.1</td>
<td>45.8</td>
</tr>
<tr>
<td>CNAPS [1]</td>
<td>DCM-S [29]</td>
<td>ResNet34</td>
<td>64.6</td>
<td>81.8</td>
<td>79.7</td>
<td>85.0</td>
<td>77.9</td>
<td>69.3</td>
<td>49.3</td>
<td>93.2</td>
<td>88.7</td>
</tr>
<tr>
<td>CTX [5]</td>
<td>ResNet34</td>
<td>62.7 ± 1.0</td>
<td>82.2 ± 1.0</td>
<td>79.5 ± 0.9</td>
<td>80.6 ± 0.9</td>
<td>75.6 ± 0.6</td>
<td>72.7 ± 0.8</td>
<td>51.6 ± 1.1</td>
<td>95.3 ± 0.4</td>
<td>82.6 ± 0.8</td>
<td>59.9 ± 1.0</td>
</tr>
<tr>
<td>TSA [17]</td>
<td>ResNet34</td>
<td>63.7 ± 1.0</td>
<td>82.6 ± 1.1</td>
<td>80.13 ± 1.0</td>
<td>83.4 ± 0.8</td>
<td>79.6 ± 0.7</td>
<td>71.0 ± 0.8</td>
<td>51.4 ± 1.2</td>
<td>94.1 ± 0.5</td>
<td>81.7 ± 1.0</td>
<td>61.7 ± 1.0</td>
</tr>
<tr>
<td>T-CNAPS [1]</td>
<td>ResNet18</td>
<td>54.1 ± 1.1</td>
<td>62.9 ± 1.3</td>
<td>48.4 ± 0.9</td>
<td>67.3 ± 0.9</td>
<td>72.5 ± 0.7</td>
<td>58.0 ± 1.0</td>
<td>37.7 ± 1.1</td>
<td>82.8 ± 0.8</td>
<td>61.8 ± 1.1</td>
<td>45.8 ± 1.0</td>
</tr>
<tr>
<td>TF [7]</td>
<td>WRN-28</td>
<td>60.5</td>
<td>82.0</td>
<td>72.4</td>
<td>82.1</td>
<td>80.5</td>
<td>57.4</td>
<td>47.7</td>
<td>92.0</td>
<td>64.4</td>
<td>42.9</td>
</tr>
<tr>
<td>TF-MP</td>
<td>ResNet18</td>
<td>62.2 ± 1.1</td>
<td>83.8 ± 1.1</td>
<td>70.9 ± 0.9</td>
<td>81.3 ± 0.8</td>
<td>79.2 ± 0.6</td>
<td>70.5 ± 0.6</td>
<td>51.2 ± 1.0</td>
<td>93.3 ± 0.4</td>
<td>78.2 ± 1.0</td>
<td>62.5 ± 0.9</td>
</tr>
<tr>
<td>TF-MP</td>
<td>ResNet34</td>
<td>66.4 ± 1.0</td>
<td>87.5 ± 0.8</td>
<td>80.0 ± 0.9</td>
<td>87.4 ± 0.6</td>
<td>81.9 ± 0.6</td>
<td>71.9 ± 0.4</td>
<td>54.9 ± 0.9</td>
<td>94.8 ± 0.4</td>
<td>89.2 ± 0.9</td>
<td>61.5 ± 0.9</td>
</tr>
</tbody>
</table>

4.2.2 Studies on Probability Regularization

Probability regularization (PR) generalizes on different testing sets. In Fig. 5, we verify the design of probability regularization and its robust performance over different settings of query set using ILSVRC-2012 validation, namely the uniform setting with an equal number of per-class testing samples and the stochastic setting with various numbers. The uniform setting uses an equal number of per-class testing samples, and the stochastic setting uses various numbers of per-class testing samples. Each class is randomly sampled from [0, 50]. The stochastic setting is more challenging than the uniform setting, which is affected by the lower general performance, as shown in Fig. 5. We ablation the design of PR: for the expected marginal distributions \( \hat{p}(y) \), we experiment on uniform distribution (Uni.) and prior distribution from estimating labeled data (Est.); for the estimated marginal distribution \( \hat{p}(y) \), using all query set (All Query.) and using one single query set with the support set (Single Query.) separately. Using Est. as \( p(y) \), 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 \( p(y) \) wins over Est as Uni., which indicates applying a stronger regularization like Uniform is beneficial to encourage class balance during fine-tuning. Note that Distribution Alignment (DA) in [2] is the same methodology as Est.+ All Query. And Uni.+ Single Query. describes PR. PR outperforms DA in both uniform and stochastic testing settings.

Probability regularization effectively improves TF w/o MW. Meanwhile, we evaluate adding PR with TF w/o MW. As shown in Table. 1, comparing with TF, adding PR improves performance over 7 datasets from 0.29% on DTD to 1.79% on Traffic sign. Further, by adding PR on TF with MW, PR brings consistent performance improvements over 10 datasets from 0.45% on Fungi to 4.27% on the Traffic sign. Firstly down-weighting samples with possibly wrong predictions using MW would strengthen the effect of PR.

4.2.3 Studies on TF-MP

TF-MP improves transductive fine-tuning with a large margin. As we illustrate in Sec. 3.3, directly and equally optimizing the predicted probabilities of all query samples will further deteriorate the class imbalance issue. This is reflected in the performance drop on 8 datasets using TF compared with fine-tuning. TF with MW essentially makes up for the tremendous performance loss of direct TF, and further adding PR brings consistent performance improvement. Meanwhile, as shown in Fig. 4b, TF-MP can effectively reduce the imbalance in per-class predictions over datasets from different domains, which demonstrates that our methodology serves as a good solution to the issue of class-imbalanced predictions in FSL.

TF-MP overall boosts performance in few-shot cases, and its better performance over fine-tuning is high-
lighted under extreme few-shot cases. As shown in Table 1, fine-tuning the feature extractor with the support set retains good domain generalization and improves performance by a large margin over all ten datasets. TF-MP can further boost the performance over 9 out of 10 datasets from 0.81% on Quickdraw to 5.14% on MSCOCO. We also conduct experiments to compare performance under a different number of images for each class in the support set. In Fig. 4c, for a 1-shot case where fine-tuning drops performance by around 5%, TF-MP makes up for the extreme lack of training samples and boosts performance over 10% compared with fine-tuning. Moreover, TF-MP also shows more considerable performance improvement compared with purely fine-tuning. This demonstrates the effectiveness and practical importance of transductive learning in few-shot classification.

### 4.3. Comparing with State-Of-The-Art

We report our results with different backbones and provide a comparison over other methods on Imagenet-only evaluation (Table 2) and All-datasets evaluation (Table 2).

**Out-of-Distribution Evaluation.** We achieve the state-of-the-art performance on Meta-Dataset evaluation with an Imagenet-only setting. Results of TF-MP on ResNet18 and ResNet34 show that with a more powerful (pre-)trained feature extractor, the performance of transductive fine-tuning is expected to be boosted. Compared with other transductive methods [1, 7], the performance of TF-MP over all ten datasets gains consistent improvement by a large margin. TF-MP with ResNet18 surpasses [1] using the same backbone and gets better results over 7 datasets compared with [7] using a larger backbone. TF-MP also beats [8] with ResNet34, a well-designed meta-learning inductive method. The performance gain of TF-MP over the first proposed transductive fine-tuning [7] implies the importance of reducing the issue of class-imbalanced predictions when utilizing the testing set.

---

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SUR ± 1.1</th>
<th>URT ± 1.1</th>
<th>FLUTE ± 1.1</th>
<th>URL ± 1.1</th>
<th>TrnM ± 1.1</th>
<th>S-CNAPS ± 1.1</th>
<th>TSA ± 1.1</th>
<th>T-CNAPS ± 1.1</th>
<th>TF-MP ± 1.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILSVRC</td>
<td>56.1 ± 1.1</td>
<td>55.7 ± 1.0</td>
<td>51.8 ± 1.1</td>
<td>57.5 ± 1.1</td>
<td>58.8 ± 1.0</td>
<td>56.5 ± 1.1</td>
<td>57.4 ± 1.1</td>
<td>57.9 ± 1.1</td>
<td>59.2 ± 1.0</td>
</tr>
<tr>
<td>Omni</td>
<td>93.1 ± 0.5</td>
<td>94.4 ± 0.4</td>
<td>93.2 ± 0.5</td>
<td>94.5 ± 0.4</td>
<td>92.0 ± 0.6</td>
<td>91.9 ± 0.6</td>
<td>94.9 ± 0.4</td>
<td>94.3 ± 0.4</td>
<td>95.8 ± 0.3</td>
</tr>
<tr>
<td>Acraft</td>
<td>84.6 ± 0.7</td>
<td>85.8 ± 0.6</td>
<td>87.2 ± 0.5</td>
<td>88.6 ± 0.5</td>
<td>82.8 ± 0.7</td>
<td>83.8 ± 0.6</td>
<td>89.3 ± 0.4</td>
<td>84.7 ± 0.5</td>
<td>89.7 ± 0.5</td>
</tr>
<tr>
<td>Birds</td>
<td>70.6 ± 1.0</td>
<td>76.3 ± 0.8</td>
<td>79.2 ± 0.8</td>
<td>80.5 ± 0.7</td>
<td>75.3 ± 0.8</td>
<td>76.1 ± 0.9</td>
<td>81.4 ± 0.7</td>
<td>78.4 ± 0.7</td>
<td>81.8 ± 0.7</td>
</tr>
<tr>
<td>DTD</td>
<td>71.0 ± 0.8</td>
<td>71.8 ± 0.7</td>
<td>68.8 ± 0.8</td>
<td>76.2 ± 0.7</td>
<td>71.2 ± 0.8</td>
<td>70.0 ± 0.8</td>
<td>78.6 ± 0.7</td>
<td>66.2 ± 0.8</td>
<td>77.0 ± 0.7</td>
</tr>
<tr>
<td>QDraw</td>
<td>81.3 ± 0.6</td>
<td>82.5 ± 0.6</td>
<td>79.5 ± 0.7</td>
<td>81.9 ± 0.6</td>
<td>77.3 ± 0.7</td>
<td>78.3 ± 0.7</td>
<td>82.0 ± 0.6</td>
<td>77.9 ± 0.6</td>
<td>82.7 ± 0.6</td>
</tr>
<tr>
<td>Fungi</td>
<td>64.2 ± 1.1</td>
<td>63.5 ± 1.0</td>
<td>58.1 ± 1.1</td>
<td>68.1 ± 1.0</td>
<td>48.5 ± 1.0</td>
<td>49.1 ± 1.2</td>
<td>67.4 ± 1.0</td>
<td>48.9 ± 1.2</td>
<td>67.9 ± 1.0</td>
</tr>
<tr>
<td>Flower</td>
<td>82.8 ± 0.8</td>
<td>88.2 ± 0.6</td>
<td>91.6 ± 0.6</td>
<td>92.1 ± 0.5</td>
<td>90.5 ± 0.5</td>
<td>91.3 ± 0.6</td>
<td>92.2 ± 0.5</td>
<td>92.5 ± 0.4</td>
<td>93.9 ± 0.4</td>
</tr>
</tbody>
</table>

Table 3. Results on All-datasets evaluation of Meta-Dataset. We provide the statistical results with a 95% confidence interval over 600 episodes. TF-MP achieves state-of-the-art performance on 9 out of 10 datasets.

---

**In-Distribution Evaluation.** Meanwhile, to further evaluate the potential of TF-MP and have a border comparison, we also benchmark our method on All-datasets evaluation by simply using TF-MP with TSA [17]. TSA [17] focuses on designing the network architecture named domain-specific adapters, which is an orthogonal direction with our method. Applying our method upon TSA [17] is to show that our method can work well with FSL methods in other directions. Compared with TSA [17], TF-MP improves performance over all 8 in-distribution datasets (0.82% average margin), 4 out of 5 out-of-distribution datasets, which demonstrates that TF-MP could be built on the latest technique of domain-specific adapters [17] in FSL. Meanwhile, TF-MP outperforms the other transductive method [1] with a large margin and achieves state-of-the-art on 7 in-distribution and 4 out-of-distribution datasets.

**Further Discussion.** Comparing TF-MP in Table 2 and Table 3, using ResNet34 trained on Imagenet-only surpasses the performance of a ResNet18 trained with all datasets on 7 out of 10 datasets. With a domain-generalized method like TF-MP, obtaining a more powerful backbone could potentially improve performance compared with extending the training datasets. We hope this discussion could be beneficial for TF-MP in real applications.

### 5. Conclusion

By solving the issue of class-imbalanced predictions in few-shot learning, we design the simple yet effective TF-MP, which is promising to enhance real-world few-shot applications. The margin-based uncertainty weighting provides a better measurement of the uncertainty in predictions with theoretical and empirical analysis. We hope the simplicity and effectiveness of margin-based uncertainty will inspire its application in other fields such as active learning and uncertainty evaluation.
References


