TOWARDS CLASS-BALANCED TRANSDUCTIVE FEW-SHOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work, we present an observation of severe class-imbalanced predictions in few-shot learning and propose solving it by acquiring a more balanced marginal probability through Transductive Fine-tuning with Margin-based uncertainty weighting and Class-balanced normalization (TF-MC). Margin-based uncertainty weighting compresses the utilization of wrong predictions with lower loss weights to stabilize predicted marginal distribution. Class-balanced normalization adjusts the predicted probability for testing data to pursue class-balanced fine-tuning without directly regularizing the marginal testing distribution. TF-MC effectively improves the class balance in predictions with state-of-the-art performance on in-/ out-of-distribution evaluations of Meta-Dataset (Triantafillou et al., 2019) and surpasses previous transductive methods by a large margin.

1 INTRODUCTION

Deep Learning has gained vital development with various architecture designs, optimization techniques, data augmentation, learning strategies, etc. As deep learning techniques demonstrate the great potential to be applied to more practical applications, the lack of manual labeling force and the difficulty of data acquisition makes few-shot learning increasingly important. Few-Shot Learning (FSL) has been quite an active research field (Finn et al., 2017; Ravi & Larochelle, 2016; Vinyals et al., 2016; Snell et al., 2017; Gidaris et al., 2019; Chen et al., 2019; Triantafillou et al., 2019) and the recent development to benchmark over datasets from different scales and domains (Triantafillou et al., 2019) encourages practically efficient algorithms in few-shot learning.

In this work, we present an observation: *the predictions in few-shot learning are severely class-imbalanced.* As shown in Figure. 1, previous state-of-the-art methods (Li et al., 2022; Tao et al., 2022; Li et al., 2021; Dhillon et al., 2019) on Meta-Dataset without exception suffers from the issue of class-imbalanced predictions. A good classification model usually learns to abstract categorical and semantic information robust to variations, which might not be sufficiently present in only a few training samples. Consequently, the insufficient learning manifests as the class-imbalanced predictions in FSL. Classes with the least number of predictions would carry lower per-class accuracy, as the number of corrected predictions is at most the number of predictions. For testing scenarios in favor of these classes, the class-imbalanced prediction could cause a fatal failure. Solving the issue of class-imbalanced predictions would improve the robustness of algorithms to

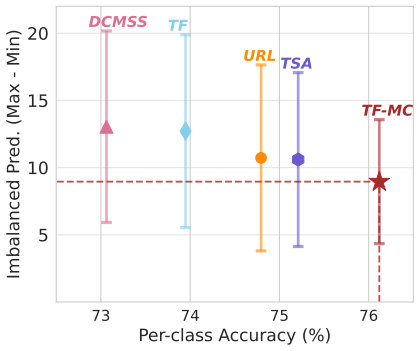

Figure 1: Class-Imbalanced Predictions in Few-Shot Learning. The difference between the maximum and minimum per-class predictions is used to quantify the level of class imbalance, and the average of imbalanced predictions with per-class accuracy is reported. Data are from ten datasets in Meta-Dataset (Triantafillou et al., 2019) with 100 episodes for each dataset and ten per-class testing samples. All previous methods suffer from the severe issue of class-imbalanced predictions; the difference between the number of per-class predictions could be more significant than ten. TF-MC successfully reduces the imbalanced predictions with the best per-class accuracy. Refer to Appendix. B for further discussions.

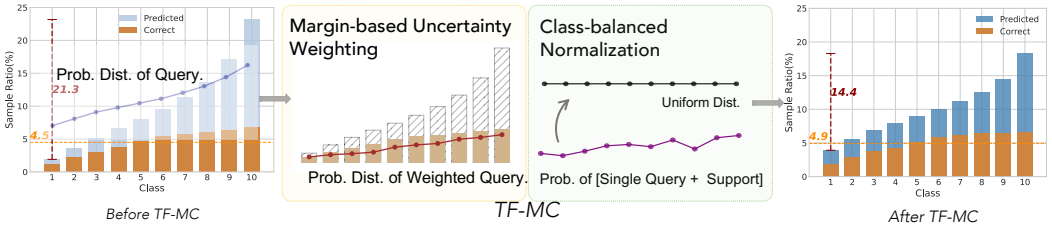

Figure 2: Illustration of TF-MC. TF-MC obtains a more balanced marginal probability by compressing the utilization of wrong predictions (Margin-Based Uncertainty Weighting) and normalizing probabilities for unlabeled data to encourage class-balanced fine-tuning (Class-balanced Normalization). By using TF-MC, the difference between per-class predictions reduces from 21.3% to 14.4% with per-class accuracy improved from 4.5% to 4.9% in 1-shot 10-way classification. Results are averaged over 100 episodes in Meta-Dataset (Triantafillou et al., 2019).

different testing scenarios, which is crucial for real applications where the testing environment is often largely non-uniform.

Recent works with (test-time) finetuning (Hu et al., 2022; Li et al., 2022; Tao et al., 2022) show great potential to fast and effectively adapt feature extractors on few-shot tasks with the state-of-the-art cross-domain performance. However, the optimization criterion in (test-time) finetuning is not designed to improve the class-balanced learning. During finetuning, parameters are optimized by minimizing the loss on a few training samples, known as empirical risk minimization (ERM), which assumes that training and testing datum follow the same distribution. However, the distribution estimated from a few training data is biased with its true distribution (Tao et al., 2022), which invalidates the assumption of ERM and potentially leads to class-imbalanced predictions on the testing datum. Transductive finetuning (Dhillon et al., 2019) proposes to optimize parameters together with training and testing datum, which to some degree dismisses the effect of biased distribution estimation from training datum. However, as with the other transductive methods in FSL, the predictions on testing datum are directly used without explicitly considering the class-imbalanced issue, which theoretically leads to sub-optimal solutions as we analyze in Sec. 2.2.

We propose Transductive Fine-tuning with Margin-based uncertainty weighting and Class-balanced normalization (TF-MC) to obtain a more balanced marginal probability during finetuning, which in turn solves the class-imbalanced predictions to some extent. Margin-based uncertainty weighting assigns per-sample loss weights according to the uncertainty scores computed from predicted probabilities. Specifically, we address the importance of utilizing top-two maximum probabilities (margin (Scheffer et al., 2001)) in entropy computation and demonstrate its supreme ability to compress the utilization of wrong predictions through experimental results. Class-balanced normalization aims to adjust the predicted probabilities of testing data to pursue class-balanced finetuning. The learned marginal probability is estimated by combining *each query sample with the full support set*, which is further aligned with the Uniform prior. In doing so, each testing sample's prediction is adjusted by a scale vector, which quantifies the difference between the marginal and uniform. The predictions of the testing datum are eventually balanced without directly regularizing on the marginal probability. TF-MC effectively alleviates the issue of class-imbalanced predictions: as shown in Fig. 1, compared with Transductive-Finetuning (TF), TF-MC largely reduces the imbalanced predictions by around 5 samples and further improves per-class accuracy with 2.1%. TF-MC shows robust cross-domain performance boosts on Meta-Dataset, demonstrating its potential in real applications.

Our contributions can be summarized as: **1).** We design an effective weighting strategy – margin-based uncertainty, to compress the utilization of wrong predictions in transductive fine-tuning. Margin-based uncertainty weighting achieves consistent performance boosts over all ten datasets across different domains in Meta-Dataset (Triantafillou et al., 2019). **2).** We propose Class-balanced normalization to individually adjust the predicted probability of testing data, encouraging the class-balanced fine-tuning while averting direct regularization on testing prior. **3).** We present an essential observation of class-imbalanced predictions in FSL and propose the simple yet efficient TF-MC to improve it. TF-MC shows robust performance boosts on Meta-Dataset in-/out-of-distribution evaluations, demonstrating its potential in real applications for different domains.

2 METHOD

For one episode in FSL, the training and testing set are referred as the support and query set, respectively. Let $(\mathbf{x}, \mathbf{y})$ denote the pair of an input $\mathbf{x}$ with its ground-truth one-hot label $\mathbf{y} \in \mathbb{R}^C$, where C is the number of classes. The support set is then represented as $\mathcal{D}_s = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_s}$, and the query set is denoted as $\mathcal{D}_q = \{(\mathbf{x}_i)\}_{i=1}^{N_q}$ where the ground-truth labels are unknown if used in the transductive manner; N_s and N_q are the total number of samples in support set and query set, respectively.

2.1 RE-VISIT TRANSDUCTIVE FINE-TUNING

We first introduce the transductive fine-tuning framework adopted in (Dhillon et al., 2019). A feature extractor f_θ 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 $\mathbf{p}_\theta(\mathbf{y}|\mathbf{x})$ as the categorical probabilities on C classes which is the output from the softmax layer in the model:

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

where $z_i = \langle \omega_i, f_\theta(\mathbf{x}) \rangle, i \in C$, the dot-product between ω_i and $f_\theta(\mathbf{x})$, is the logit for class i. As widely used in (Snell et al., 2017; Qi et al., 2018; Chen et al., 2020; Tao et al., 2022; Li et al., 2022), ω_i is the novel class prototype that is initialized as the mean feature from the support set $\mathcal{D}_s$. A model with parameter θ is learnt to classify $\mathcal{D}_s$ and $\mathcal{D}_q$ as measured by the following criterion:

$$\theta^*(\mathcal{D}_s, \mathcal{D}_q) = \text{argmin}_\theta \Big(\frac{1}{N_s} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_s} \mathcal{L}_s(\mathbf{x}, \mathbf{y}) + \frac{1}{N_q} \sum_{(\mathbf{x}) \in \mathcal{D}_q} \mathcal{L}_q(\mathbf{x})\Big). \tag{2}$$

The loss $\mathcal{L}_s(\mathbf{x}, \mathbf{y})$ for the labeled support set is the cross-entropy loss. And the loss $\mathcal{L}_q(\mathbf{x})$ for the unlabeled query set is constructed as entropy minimization:

$$\mathcal{L}_q(\mathbf{x}) = \lambda H(\mathbf{p}_\theta(\mathbf{y}|\mathbf{x}), \mathbf{p}_\theta(\mathbf{y}|\mathbf{x})) = \lambda \mathbf{p}_\theta(\mathbf{y}|\mathbf{x}) \log(\mathbf{p}_\theta(\mathbf{y}|\mathbf{x})), \tag{3}$$

where λ denotes the *per-sample* loss weight. The previous transductive finetuning (Dhillon et al., 2019) assigns equal weights to testing samples ($\lambda = 1$).

2.2 CLASS-BALANCED NORMALIZATION

We firstly illustrate that the current optimization under Eq. 3 does not necessarily improve class balance in predictions but to some degree further worsen it. We take the class with the most prediction c_{max} as a illustration. For samples wrongly predicted to c_{max}, $\langle \omega_{c_{max}}, f_\theta(\mathbf{x}) \rangle > \langle \omega_y, f_\theta(\mathbf{x}) \rangle$ directly leads to the wrong prediction. In other words, features are closer to the prototype of $\omega_{c_{max}}$ rather than the prototype of the ground-truth class.

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

$$\frac{\partial \mathcal{L}_q}{\partial f_\theta(\mathbf{x})} = \sum_{i}^{C} \frac{\partial \mathcal{L}_q}{\partial z_i} \omega_i \tag{4}$$

Where $\frac{\partial \mathcal{L}_q}{\partial z_i}$ serves as the scalar to control how much the gradient would update with ω_i. By summing over all samples, the gradient on $z_i, i \in C$ is:

$$\frac{\partial \mathcal{L}_q}{\partial z_i} = -\frac{1}{N_q} \sum_{j}^{N_q} p_{ij}(1 - p_{ij}) \tag{5}$$

Where $p_{ij} = p_\theta(i|j)$ to simplify notations. By applying gradient descent, the optimization through features are:

$$f_\theta(\mathbf{x}) = f_\theta(\mathbf{x}) - \frac{\partial \mathcal{L}_q}{\partial f_\theta(\mathbf{x})} = f_\theta(\mathbf{x}) + \sum_i^C (\frac{1}{N_q}\sum_j^{N_q} p_{ij}(1-p_{ij}))\omega_i \tag{6}$$

The overall stochastic gradient update on $f_\theta(\mathbf{x})$ leads the direction of $f_\theta(\mathbf{x})$ towards a direction that is weighted by the absolute value of $|\frac{\partial \mathcal{L}_q}{\partial z_i}|$ on ω_i. Meanwhile by Jensen's inequality:

$$|\frac{\partial \mathcal{L}_q}{\partial z_i}| = \frac{1}{N_q}\sum_j^{N_q}(p_{ij}(1-p_{ij})) \geq p(i)(1-p(i)) \tag{7}$$

And $p(i)$ is the marginal probability for class i involved during finetuning:

$$p(i) = \frac{1}{N_q}\sum_j^{N_q} p(i|j) \tag{8}$$

As shown in Eq. 7, $|\frac{\partial \mathcal{L}_q}{\partial z_i}|$ is lower bounded by the marginal probability. Thus class c_{max} owns the relatively larger gradient $\frac{\partial \mathcal{L}_q}{\partial z_{c_{max}}}$ than other classes. *As indicated in Eq. 4 the larger* $|\frac{\partial \mathcal{L}_q}{\partial z_{c_{max}}}|$ *deviates the optimization of* $f_\theta(x)$ *getting closer to the direction of* $\omega_{c_{max}}$*, which exacerbates the class-imbalanced prediction by further deflecting on* c_{max}. Detailed discussions are further provided in Appendix.E.

This addresses the necessity of pursuing a class-balanced marginal probability $p(\mathbf{y})$ during finetuning which we propose to solve by Margin-based uncertainty weighting and Class-balanced normalization. To balance marginal probability, we adjust the predicted probability $p(y|x)$ for each testing data by class-balanced normalization, which is designed as the following: for each $\mathbf{x} \in \mathcal{D}_q$, it is combined with the full support set as $x \cup \mathcal{D}_s$; and the current learned marginal probability is estimated using $x \cup \mathcal{D}_s$. By aligning the estimated marginal probability with a uniform prior, a unique scale vector is obtained for each testing sample, which is further used in probability normalization. Formally, for $\mathbf{q} = p_\theta(\mathbf{y}|\mathbf{x})$:

$$\tilde{\mathbf{q}} = \text{Normalize}(\mathbf{q}\frac{U}{\hat{E}_{x\cup\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]}) \tag{9}$$

Where $\text{Normalize}(x_i) = \frac{x_i}{\sum_j x_j}$. In doing so, each sample from the query set obtains a unique scale vector $\frac{U}{\hat{E}_{x\cup\mathcal{D}_s}[p_\theta(\mathbf{y}|\mathbf{x})]}$ which allows per-sample probability normalization. Meanwhile, aligning the estimated marginal probability of $x \cup \mathcal{D}_s$ to Uniform avoids direct regularization on the marginal probability of the whole query set, which allows the probability normalization theoretically effective when the actual testing set is not uniform. Detailed discussion is further provided in Appendix.A.

2.3 Margin-based Uncertainty Weighting

We propose to assign each unlabeled sample with a loss weight $\lambda(\mathbf{p})$ associated with its possibility of being a wrong prediction, which equally turns Eq. 8 into:

$$p(i) = \frac{1}{N_q}\sum_j^{N_q} \lambda(\mathbf{p})p(i|j) \tag{10}$$

By compressing the utilization of possibly wrong predictions using $\lambda(p(i|j))$, a more balanced marginal probability is obtained from weighted testing data.

The maximum of probabilities p_{max} referred as confidence (Guo et al., 2017) is used to assign the predicted class. In semi-supervised learning (Iscen et al., 2019), entropy-based per-sample loss weight is used as:

$$\lambda(\mathbf{p}) = 1 - e(\mathbf{p}) \tag{11}$$

And $e(\mathbf{p})$ refers to the normalized entropy:

$$e(\mathbf{p}) = -\frac{\sum_i^c (p_i \log p_i)}{\log c} \tag{12}$$

where $\sum_i^c p_i = 1, \mathbf{p} = [p_1, p_2, ...p_c]$ and c is the number of classes. $e(\mathbf{p})$ is normalized to $[0, 1]$ as the entropy $\sum_i^c (p_i \log p_i)$ is scaled by its maximum value $\log c$. Entropy on $\mathbf{p}(\mathbf{y}|\mathbf{x})$ quantifies the uncertainty of probabilities, and larger uncertainty generally refers to a lower confidence level the sample carries towards its class prediction. However, when diving into Eq. 12, we discover that *the uncertainty on the whole probability distribution* may not be ideal to distinguish 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 Δp can largely reflect how "uncertain" (low margin) or "certain" (large margin) an example is with its prediction (Scheffer et al., 2001).

When p_{max} is fixed, margin Δp is in the range of: $\min(\Delta p) = p_{max} - (1 - p_{max})$, $\max(\Delta p) = p_{max} - \frac{1-p_{max}}{c-1}$. For $\max(\Delta p)$, the entropy is:

$$e_{\max(\Delta p)} = e_{\min(\Delta p)} + \frac{(1 - p_{max}) \log(c - 1)}{\log c} \tag{13}$$

As $\frac{(1-p_{max}) \log(c-1)}{\log c}$ is non-negative, Eq. 13 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**.

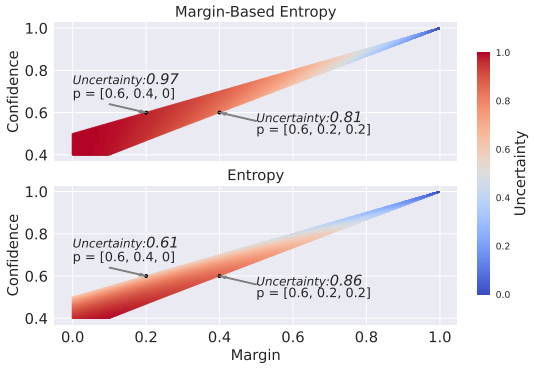

Figure 3: A 3-class Illustration on Uncertainty Scores computed by Margin-based entropy and entropy. We plot the values of uncertainty scores according to confidence and margin. Colors represent uncertainty scores referred as the colorbar attached. Entropy assigns lower uncertainty scores over the minimum margin area (lighter red) which is opposite with the margin information. Margin-based entropy assigns higher uncertainty scores (darker red) over the low confidence (0.4 - 0.5) and small margin areas. Margin-based entropy applies the uncertainty scores consistent with margin: increasing the uncertainty score of $p = [0.6, 0.4, 0]$ (0.2 margin) from 0.61 to 0.98 and decreasing the uncertainty score of $p = [0.6, 0.2, 0.2]$ (0.4 margin) from 0.86 to 0.81.

To solve this contradiction, we address the importance of only using top-2 probabilities in Eq. 12. The maximum and second maximum probabilities are first normalized to satisfy the requirement of $\sum_i^c p_i = 1$ in Eq. 12 and further used in Eq. 14. This simple modification can unify the information carried by both confidence, margin and entropy. The margin-based uncertainty is defined as:

$$\hat{e}(\mathbf{p}) = \frac{-1}{\log 2} \left[p_{max} \log p_{max} + (p_{max} - \Delta p) \log(p_{max} - \Delta p) \right] \tag{14}$$

When margin $\hat{\Delta} p$ is fixed, $\hat{e}(\mathbf{p})$ is non-decreasing with confidence p_c; when confidence p_c is fixed, margin $\hat{e}(\mathbf{p})$ is as well non-decreasing with $\hat{\Delta} p$. In doing so, the margin-based entropy score could consistently reflect the confidence level $max(\mathbf{p})$ as well as the margin $\Delta \mathbf{p}$, as shown in Fig. 3. By focusing on the uncertainty delivered by the margin in $\mathbf{p}$, it achieves stronger compression on utilization of wrong predictions compared with entropy-based loss weights, which is furthered illustrated in Appendix.D.

Method	CN	MW	ILSVRC	Omni	Acraft	Birds	DTD	QDraw	Fungi	Flower	Sign	COCO
			58.57	68.13	53.32	76.55	74.38	77.68	44.34	89.11	49.61	56.4
F			59.96	78.7	**72.32**	78.30	76.96	86.04	47.51	91.95	76.39	57.32
TF			59.19	73.71	57.56	77.53	75.63	80.83	48.18	90.14	60.42	58.82
TF	✓		59.63	74.58	56.45	76.07	75.92	81.4	46.04	91.26	62.21	59.17
TF		✓	61.49	81.64	68.88	80.23	78.55	85.2	50.72	92.67	73.96	60.09
TF	✓	✓	**62.18**	**83.78**	70.9	**81.25**	**79.15**	**86.85**	**51.17**	**93.3**	**78.23**	**62.46**

Table 1: Ablation studies using ResNet18. Results are reported using average of 600 episodes. The first row corresponds to the performance of the Proto-classifier. fine-tuning (F) the backbone is firstly evaluated. Transductive Finetune (TF), Margin-based Uncertainty Weighting (MW) and Class-balanced Normalization (CN) separately or combined are verified. TF with MW and CN achieves the best results in the ablation study.

3 Experimental Validation

In this section, we first conduct comprehensive ablation experiments to verify the effectiveness of Margin-based Uncertainty Weighting and Class-balanced Normalization and address the essential role of transductive fine-tuning for extreme few-shot cases compared with purely fine-tuning. Further we evaluate and compare our results with the other latest techniques on Meta-Dataset (Triantafillou et al., 2019) *Imagenet-only* and All-datasets evaluations.

3.1 Implementation Details

A Briefing on Datasets. We evaluate our method on Meta-Dataset (Triantafillou et al., 2019), 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, also referred as *All-datasets* evaluation, allows using available training sets from 8 of 10 datasets and the out-of-distribution evaluation, referred as *Imagenet-only* evaluation, allows only using the training set from ILSVRC-2012 (Russakovsky et al., 2015b).

Pre-training the Backbone: Choice of the Network and Training Setting. For *Imagenet-only* evaluation, the ILSVRC-2012 (Russakovsky et al., 2015a) in Meta-Dataset is splitted 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 (Li et al., 2021; 2022) as the feature extractor for *All-Datasets* evaluation. For ResNet18, we follow the same protocol in Meta-Baseline (Chen et al., 2020), which is: the images are randomly resized cropped to 128x128, horizontal flipped and normalized. For ResNet34, we follow the same structure modification in (Doersch et al., 2020) 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 total 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 (Triantafillou et al., 2019), 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 (Dhillon et al., 2019; Tao et al., 2022): learning rate of 5e-5, Adam optimizer and 25 total epochs. We follow the same metrics in meta-Baseline (Chen et al., 2020): 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 (Li et al., 2022) and follow the same fine-tuning setting with adadelta but only extending 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 (Chen et al., 2020). *Finetune* refers to only using the support set to finetune the (pre-)trained feature extractor. *TF-MC* refers to our methods.

Method	Model	ILSVRC	Omni	Acraft	Birds	DTD	QDraw	Fungi	Flower	Sign	COCO
fo-P-M (Triantafillou et al., 2019)	–	49.5 ± 1.1	60.0 ± 1.4	53.1 ± 1.0	68.8 ± 1.0	66.6 ± 0.8	49.0 ± 1.1	39.7 ± 1.1	85.3 ± 0.8	47.1 ± 1.1	41.0 ± 1.1
BOHB (Saikia et al., 2020)	–	51.9 ± 1.1	67.6 ± 1.2	54.1 ± 0.9	70.7 ± 0.9	68.3 ± 0.8	50.3 ± 1.0	41.4 ± 1.1	87.3 ± 0.6	51.8 ± 1.0	48.0 ± 1.0
LR (Tian et al., 2020)	R-18	60.1	64.9	63.1	77.7	78.6	62.5	47.1	91.6	77.5	57.0
Meta-B (Chen et al., 2020)	R-18	59.2	69.1	54.1	77.3	76.0	57.3	45.4	89.6	66.2	55.7
CNAPS (Bateni et al., 2022)	R-18	54.8	62.0	49.2	66.5	71.6	56.6	37.5	82.1	63.1	45.8
DCM-SS (Tao et al., 2022)	R-34	64.6	81.8	79.7	85.0	77.9	87.1	49.3	93.2	88.7	57.7
CTX (Doersch et al., 2020)	R-34	62.7 ± 1.0	82.2 ± 1.0	79.5 ± 0.9	80.6 ± 0.9	75.6 ± 0.6	72.7 ± 0.8	51.6 ± 1.1	**95.3** ± 0.4	82.6 ± 0.8	59.9 ± 1.0
TSA (Li et al., 2022)	R-34	63.7 ± 1.0	82.6 ± 1.1	80.1 ± 1.0	83.4 ± 0.8	79.6 ± 0.7	71.0 ± 0.8	51.4 ± 1.2	94.1 ± 0.5	81.7 ± 1.0	61.7 ± 1.0
T-CNAPS (Bateni et al., 2022)	R-18	54.1 ± 1.1	62.9 ± 1.3	48.4 ± 0.9	67.3 ± 0.9	72.5 ± 0.7	58.0 ± 1.0	37.7 ± 1.1	82.8 ± 0.8	61.8 ± 1.1	45.8 ± 1.0
T-F (Dhillon et al., 2019)	WR-28	60.5	82.0	72.4	82.1	80.5	57.4	47.7	92.0	64.4	42.9
TF-MC	R-18	62.2 ± 1.1	83.8 ± 1.1	70.9 ± 0.9	81.3 ± 0.8	79.2 ± 0.6	86.9 ± 0.6	51.2 ± 1.0	93.3 ± 0.4	78.2 ± 1.0	**62.5** ± 0.9
TF-MC	R-34	**66.4** ± 1.0	**87.5** ± 0.8	**80.3** ± 0.9	**87.4** ± 0.6	**81.9** ± 0.6	**87.3** ± 0.4	**54.9** ± 0.9	94.8± 0.4	**89.2** ± 0.9	61.5 ± 0.9

Table 2: Results on *Imagenet-only* evaluation of Meta-Dataset. We provide the statistical results with 95% confidence interval over 600 episodes. We provide the statistical results of over 600 episodes. TF-MC brings consistent performance improvements over all ten datasets compared with recent works.

3.2 Ablation Studies

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

Margin-based uncertainty Weighting (MW) out-performs the entropy-based weights by a large margin and shows domain-agnostic performance boosts over all datasets. We first conduct ablations to compare MW with entropy-based weights. As shown in Fig. 4, using MW offers an absolute advantage of around 6% performance gain over equally weighting unlabeled samples (i.e., without using MW). Moreover, reducing the number of top-k probabilities used in entropy computation improves the performance comparing with entropy-based weights (All), where MW (top-2) achieves the most improvement. The results further support the importance of addressing top-2 probabilities to distinguish wrong and correct predictions as illustrated in Sec. 2.3. In Table. 1, adding MW with TF helps to make up for the performance loss using TF. Comparing with only TF, adding MW brings performance boosts from 1.27% on MSCOCO to 13.54% on Traffic sign. Meanwhile, TF with MW surpasses fine-tuning over 7 out of 10 datasets with performance margins from 0.72% on VGG-flower to 3.21% on Fungi. This demonstrates the importance of down weighting samples with wrong prediction during transductive fine-tuning. The consistent performance gains demonstrate the robust domain generalization of transductive fine-tuning with MW.

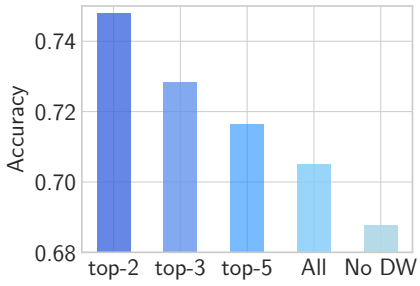

Figure 4: Weights with Top-k Prob. MW (Top-2) outperforms entropy-based weights (All) with a large margin. Results are averaged accuracy among datasets with 600 episodes for each dataset.

Class-balanced Normalization (CN) effectively improves Transductive finetuning w/o MW. We evaluate adding CN with TF w/o MW. As shown in Table. 1, comparing with TF, adding CN improves performance over 7 datasets from 0.29% on DTD to 1.79% on Traffic sign with an average improvement of 0.77%. And further by adding CN on TF with MW, CN brings consistent performance improvements over 10 datasets from 0.45% on Fungi to 4.27% on Traffic sign. By firstly down weighting samples with possibly wrong predictions using MW would manifest the effect of CN further. We also provide thorough ablations to verify that CN generalizes well under different testing scenarios in Appendix.

TF-MC improves transductive fine-tuning with a large margin. As we illustrate in Sec. 2.2, directly and equally optimizing the predicted probabilities of all query samples will further deteriorate the class imbalance issue, and this is reflected as the performance drop on 8 datasets using TF

Dataset	SUR	URT	FLUTE	URL	TriM	S-CNAPS	TSA	T-CNAPS	TF-MC
ILSVRC	56.1 ± 1.1	55.7±1.0	51.8±1.1	57.5 ± 1.1	58.6 ± 1.0	56.5 ± 1.1	57.4 ± 1.1	57.9 ± 1.1	**59.2** ± 1.0
Omni	93.1 ± 0.5	94.4±0.4	93.2±0.5	94.5 ± 0.4	92.0 ± 0.6	91.9 ± 0.6	94.9 ± 0.4	94.3 ± 0.4	**95.8** ± 0.3
Acraft	84.6 ± 0.7	85.8±0.6	87.2±0.5	88.6 ± 0.5	82.8 ± 0.7	83.8 ± 0.6	89.3 ± 0.4	84.7 ± 0.5	**89.7** ± 0.5
Birds	70.6 ± 1.0	76.3±0.8	79.2±0.8	80.5 ± 0.7	75.3± 0.8	76.1 ± 0.9	81.4 ± 0.7	78.8 ± 0.7	**81.8** ± 0.7
DTD	71.0 ± 0.8	71.8±0.7	68.8±0.8	76.2 ± 0.7	71.2± 0.8	70.0 ± 0.8	76.8 ± 0.7	66.2 ± 0.8	**77.0** ± 0.7
QDraw	81.3 ± 0.6	82.5±0.6	79.5±0.7	81.9 ± 0.6	77.3± 0.7	78.3 ±0.7	82.0 ± 0.6	77.9 ± 0.6	**82.7** ± 0.6
Fungi	64.2 ± 1.1	63.5 ±1.0	58.1±1.1	**68.1** ± 1.0	48.5±1.0	49.1 ±1.2	67.4±1.0	48.9 ± 1.2	67.9 ± 1.0
Flower	82.8 ± 0.8	88.2 ± 0.6	91.6±0.6	92.1 ± 0.5	90.5 ± 0.5	91.3 ±0.6	92.2±0.5	92.3 ± 0.4	**93.9** ± 0.4
Sign	51.0 ± 1.1	48.2±1.1	58.4±1.1	63.3 ± 1.1	58.4 ± 1.1	63.0 ± 1.0	82.8 ±1.0	59.7 ± 1.1	**84.5** ± 1.0
COCO	50.1 ± 1.0	52.2±1.1	50.0 ±1.0	54.0 ± 1.0	52.8 ± 1.1	42.4 ± 1.1	55.8±1.1	42.5 ± 1.1	**56.2** ± 1.1
MNIST	94.3 ± 0.4	90.6 ± 0.5	96.2 ± 0.3	94.7 ± 0.4	95.6 ± 0.5	94.6 ± 0.4	96.7 ± 0.4	94.7 ± 0.3	**96.8** ± 0.2
CIFAR10	66.5 ± 0.9	67.0 ± 0.8	75.4 ± 0.8	72.4 ± 0.8	78.6 ± 0.7	74.9 ± 0.7	**82.9** ± 0.7	73.6 ± 0.7	82.6 ± 0.8
CIFAR100	56.9 ± 1.1	57.3 ±1.0	62.0 ± 1.0	63.5 ± 1.0	67.1 ± 1.0	61.3 ± 1.1	70.4 ± 0.9	61.8 ± 1.0	**71.6** ± 0.9

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

compared with fine-tuning. TF with MW essentially makes up for the tremendous performance loss of direct TF, and further adding CN brings consistent performance improvement.

Compared with fine-tuning, TF-MC overall boosts performance in few-shot cases, and its win over fine-tuning is addressed 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-MC 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. 5, for a 1-shot case where fine-tuning drops performance by around 5%, TF-MC makes up for the extreme lack of training samples and boosts performance over 10% compared with fine-tuning. Moreover, TF-MC 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.

3.3 Comparing with State-of-the-art

We report our results under different backbone models and provide a comparison over other popular methods in Table. 2 for *Imagenet-only* evaluation and Table. 3 for *All-datasets* evaluation. We achieve the state-of-the-art performance on Meta-Dataset evaluation with an *Imagenet-only* setting. Results of TF-MC 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 (Bateni et al., 2022; Dhillon et al., 2019), the performance of TF-MC over all ten datasets gains consistent improvement by a large margin. TF-MC with ResNet18 surpasses (Bateni et al., 2022) using the same backbone and gets better results over 7 datasets compared with (Dhillon et al., 2019) using a larger backbone. TF-MC also beats (Doersch et al., 2020) with ResNet34, a well designed meta-learning inductive method. The performance gain of TF-MC over the first proposed transductive fine-tuning(Dhillon et al., 2019) implies the importance of reducing the issue of class-imbalanced predictions when utilizing the testing set.

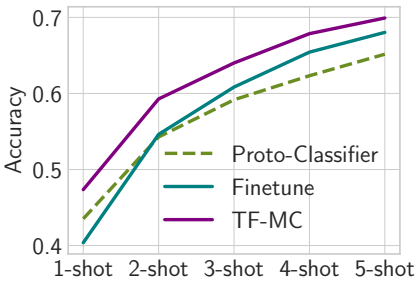

Figure 5: N-shot Analysis. Results are averaged accuracy over datasets with 600 episodes. TF-MC boosts performance where fine-tuning leads to significant performance drop especially for 1-shot case.

Meanwhile, to further evaluate the potential of TF-MC and have a border comparison, we also benchmark our method on *All-datasets* evaluation by simply using TF-MC with TSA Li et al. (2022). Comparing with TSA Li et al. (2022), TF-MC improves performance over all 8 in-distribution datasets (0.82% average margin), 4 out of 5 out-of-distribution datasets, which demonstrates that TF-MC could be built on the latest technique of domain-specific adapters (Li et al., 2022)in FSL.

Meanwhile, TF-MC outperforms the other transductive method (Bateni et al., 2022) with a large margin and achieves state-of-the-art on 7 in-distribution datasets and 4 out-of-distribution datasets.

With TF-MC, using ResNet34 trained on Imagenet-only surpasses the performance of a ResNet18 trained with training sets from all datasets on 7 out of 10 datasets. With a domain-generalized method like TF-MC, obtaining a more powerful backbone could potentially improve performance compared with extending the training datasets. We hope this discussion could be beneficial for TF-MC in real applications.

4 Related Work

Transductive Few-Shot Learning: Transductive few-shot learning uses the unlabeled query set (testing images) along the support set (training images) to make up for the lack of training data. (Nichol et al., 2018) made the attempt of updating parameters of batch normalization layers using unlabelled query samples. (Liu et al., 2018) proposes to propagate labels for unseen classes through episodic meta-learning and (Bateni et al., 2022) presents the label refinement with a Mahalanobis-distance based classifier. (Boudiaf et al., 2020) designs a loss to encourage the marginal distribution of query set to be uniform and pseudo-labels are directly used without compressing the possibly wrong predictions. (Hu et al., 2021) uses Optimal Transport Algorithm for pseudo label mapping with entropy minimization of OTA-based mapping, which implicitly forces the testing marginal probability to be uniform. (Lichtenstein et al., 2020) computes a linear projection space on features for each task when utilizing the query set, which focuses on different directions with TF-MC. (Boudiaf et al., 2020; Hu et al., 2021) enforce the testing distribution to be uniform and don't propose to compress the utilization of possibly wrong predictions. In (Dhillon et al., 2019), a transductive framework is firstly proposed to involve the testing images (query set) during fine-tuning. (Dhillon et al., 2019) builds the classification upon predicted logits other than directly on features. In contrast, we benchmark the result of transductive fine-tuning directly on features and further propose methods that focus on reducing the class imbalance in predictions. It is worth noticing that previous works on transductive few-shot learning directly use the class-imbalanced predictions and ignore compressing the utilization of possibly wrong predictions.

Semi-supervised Learning: Semi-supervised learning (SSL) is designed to introduce extra unlabelled data into training set, which is different from the transductive methods that utilize the unlabeled testing samples. Suppressing the influence of possible wrong predictions is as well an important task in SSL. There are methods like assigning per-sample loss weights using entropy-measured probability uncertainty (Iscen et al., 2019) and selecting samples with a strictly high confidence threshold (Sohn et al., 2020). We compare our margin-based uncertainty weighting with entropy-based weighting thoroughly in Sec. 2. And as shown in Appendix, the average confidence of correct predictions is only 0.4 empirically in FSL. Few-shot learning limits testing samples to have a very high confidence and the performance for different datasets varies which makes the handcraft high-confidence threshold inapplicable in this case.

Distribution Alignment and Confidence Calibration: Confidence calibration proposed in (Guo et al., 2017) targets at post-processing to calibrate the overall confidence distribution matching with the true correctness likelihood. Our work mainly address the class-wise imbalance issue on predictions and propose Class-balanced Normalization to adjust probabilities for each testing sample on the fly during fine-tuning. Distribution Alignment (DA) in (Berthelot et al., 2019) is designed to match the predicted marginal distribution of unlabeled data with the marginal distribution of labeled groundtruth. We provide thorough comparsion with DA in Appendix. Class alternating normalization (Jia et al., 2021) proposes the alternation normalization to normalize the probability for un-confident samples with the prior distribution of confident samples through multiple iterations in the post-training stage.

5 Conclusion

In this work, we design the simple yet effective TF-MC by focusing on solving the issue of class-imbalanced predictions in few-shot classification and we hope the strong performance of TF-MC would encourage more practical usages of testing-time finetuning in real-world few-shot applications.

REFERENCES

Peyman Bateni, Jarred Barber, Raghav Goyal, Vaden Masrani, Jan-Willem van de Meent, Leonid Sigal, and Frank Wood. Beyond simple meta-learning: Multi-purpose models for multi-domain, active and continual few-shot learning. *arXiv preprint arXiv:2201.05151*, 2022.

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.

Malik Boudiaf, Imtiaz Ziko, Jérôme Rony, José Dolz, Pablo Piantanida, and Ismail Ben Ayed. Information maximization for few-shot learning. *Advances in Neural Information Processing Systems*, 33:2445–2457, 2020.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification, 2019.

Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. A new meta-baseline for few-shot learning. *arXiv preprint arXiv:2003.04390*, 2020.

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

Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. *arXiv preprint arXiv:2007.11498*, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. *CoRR*, abs/1906.05186, 2019. URL `http://arxiv.org/abs/1906.05186`.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Shell Xu Hu, Da Li, Jan Stühmer, Minyoung Kim, and Timothy M Hospedales. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9068–9077, 2022.

Yuqing Hu, Vincent Gripon, and Stéphane Pateux. Leveraging the feature distribution in transfer-based few-shot learning. In *International Conference on Artificial Neural Networks*, pp. 487–499. Springer, 2021.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5070–5079, 2019.

Menglin Jia, Austin Reiter, Ser-Nam Lim, Yoav Artzi, and Claire Cardie. When in doubt: Improving classification performance with alternating normalization. *arXiv preprint arXiv:2109.13449*, 2021.

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.

Moshe Lichtenstein, Prasanna Sattigeri, Rogerio Feris, Raja Giryes, and Leonid Karlinsky. Tafssl: Task-adaptive feature sub-space learning for few-shot classification. In *European Conference on Computer Vision*, pp. 522–539. Springer, 2020.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. *arXiv preprint arXiv:1805.10002*, 2018.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5822–5830, 2018.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015a. doi: 10.1007/s11263-015-0816-y.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015b.

Tonmoy Saikia, Thomas Brox, and Cordelia Schmid. Optimized generic feature learning for few-shot classification across domains. *arXiv preprint arXiv:2001.07926*, 2020.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*, pp. 309–318. Springer, 2001.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

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.

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.

Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*, 2020.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pp. 3630–3638, 2016.