Parameterless Transductive Feature Re-representation for Few-Shot Learning

Wentao Cui [1] **Yuhong Guo** [1 2]

Abstract

Recent literature in few-shot learning (FSL) has shown that transductive methods often outperform their inductive counterparts. However, most transductive solutions, particularly the meta-learning based ones, require inserting trainable parameters on top of some inductive baselines to facilitate transduction. In this paper, we propose a parameterless transductive feature re-representation framework that differs from all existing solutions from the following perspectives. (1) It is widely compatible with existing FSL methods, including meta-learning and fine tuning based models. (2) The framework is simple and introduces no extra training parameters when applied to any architecture. We conduct experiments on three benchmark datasets by applying the framework to both representative meta-learning baselines and state-of-the-art FSL methods. Our framework consistently improves performances in all experiments and refreshes the state-of-the-art FSL results.

1. Introduction

Deep learning has gained huge success across wide applications in recent years, including computer vision, natural language processing and reinforcement learning (LeCun et al., 2015). Sophisticated deep neural network architectures with carefully tuned hyperparameters can surpass human level performance (Silver et al., 2017). However, such success is typically conditioned on one key resource: sufficient annotated training data, which is hardly available in real world. Data annotation is often either time consuming or expensive in many real world domains. Therefore, a machine learning model has to train with limited annotated data, where only a few labeled instances for each category are available. Such practical problem is termed as few-shot learning (FSL) and has attracted enormous attention in the past few years.

[1]School of Computer Science, Carleton University, Canada [2]Canada CIFAR AI Chair, Amii. Correspondence to: Wentao Cui <wentaocui@cmail.carleton.ca>, Yuhong Guo <yuhong.guo@carleton.ca>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

FSL solutions are generally developed in two branches: meta-learning based methods and fine tuning based methods. In both branches, transductive settings have been reported to outperform their inductive counterparts (Kim et al., 2019; Liu et al., 2019; Hou et al., 2019; Dhillon et al., 2020; Boudiaf et al., 2020). In general, transductive algorithms can (1) perform information (feature and/or label) propagation between query and support sets (Kim et al., 2019; Liu et al., 2019; Hou et al., 2019), (2) incorporate query instances' loss terms into the training objectives (Dhillon et al., 2020; Boudiaf et al., 2020; Antoniou & Storkey, 2019), and (3) utilize query instances' pseudo labels for support set augmentation (Liu et al., 2020; Ziko et al., 2020). Although transductive FSL has exactly the same amount of training/test data as their inductive counterparts (Boudiaf et al., 2020), information of unlabelled instances can be quite beneficial when data annotation is scarce. For example, two recent works (Dhillon et al., 2020; Boudiaf et al., 2020) experimentally demonstrate that simply adding transductive inference terms in the training objective can significantly boost performance. This design requires re-training and is applicable to fine tuning based frameworks. A more widely explored approach is to fuse features of unlabelled instances into other receiving instances (including both unlabelled and labelled), as in (Kim et al., 2019; Yang et al., 2020; Liu et al., 2019). The intuition lies in one intrinsic nature of FSL: few-shot labelled instances can hardly represent the belonging classes. We term this issue as *sample bias*. Reducing sample bias by exploiting the unlabelled instances can significantly improve the model performance.

While transductive setting is beneficial, the aforementioned models generally lack one or both of two critical properties: (1) wide applicability and (2) lightweight model design. For example, graph neural network (GNN) based models form a unique type of solutions for FSL and have achieved impressive performances (Liu et al., 2019; Kim et al., 2019; Yang et al., 2020). While being powerful, the GNN acts as a key part of the classifier and introduces many extra parameters. The specific architecture design also makes its generalization to other applications not easy (Hou et al., 2019). In (Hou et al., 2019), although no heavyweight classifier is adopted as in GNN based models, a meta-learner is trained to generate kernels to achieve feature propagation between labelled and unlabelled instances. Transductive setting, par-

ticularly when designed with meta-learning based models, often requires introducing extra parameters.

Our proposed idea in this paper possesses both desired properties while alleviating sample bias in FSL. Specifically, we propose a transductive feature re-representation framework that enriches the features of each instance by merging information from the unlabelled instances. The proposed framework can work as a simple plug-in layer in the extracted feature space and is compatible with most existing FSL solutions, including meta-learning and fine tuning based ones. Unlike most existing transductive FSL models that insert extra parameters into an inductive baseline, our framework is free of training parameters, presented as a set of transformation formulas applied in the feature space. To properly fuse features of unlabelled instances into the receiving instances as a weighted sum, we utilize an attention mechanism by comparing the instances' distances. Moreover, we also include a self-supervised learning (SSL) loss to regularize the feature re-representation and facilitate representation learning when the framework is incorporated during meta-training or fine tuning. We apply our framework to three representative meta-learning baselines, Prototypical Network, Matching Network, and Relation Network (Snell et al., 2017; Vinyals et al., 2016; Sung et al., 2018) and verify its effectiveness on three FSL benchmark datasets: *mini*-ImageNet, *tiered*-ImageNet and CUB. The proposed transductive framework yields consistent notable improvements on all three meta-learning models. We also apply our framework to three recently developed strong non-meta learning FSL models (Wang et al., 2019; Boudiaf et al., 2020; Ziko et al., 2020) on the same benchmark datasets, which refreshes the state-of-the-art results. The improvements are particularly substantial in 1-shot scenarios, where sample bias is most significant.

2. Related Work

2.1. Transductive Few-Shot Learning

Information (feature/label) propagation models. A large portion of transductive FSL models design specific architectures to facilitate information propagation between the labelled and unlabelled instances. These architectures mostly achieve an attention mechanism over the candidate instances, whose features or labels are propagated. One representative design is GNN based meta-learning models. Liu et al. (2019) and Kim et al. (2019) use GNN to calculate instance-wise similarity (and dissimilarity) for label propagation. Yang et al. (2020) develop a dual-GNN to model both instance features and label distributions. In these models, the edges of the graph are the effective attention (or weight). Other non-GNN meta-learning models use special metrics or even parametric attention modules to calculate the attention scores. For example, Qiao et al. (2019) use bi-directional similarity softmax scores to highlight good matchings between queries and classes. By comparing to the incoming query's features, Doersch et al. (2020) represent the class prototype as a weighted sum of all the belonging instances' spatial features at all locations. However, in 1-shot scenarios, this weighted sum is only along the spatial dimension and will suffer from the sample bias. Hou et al. (2019) propose a parametric meta-learner to generate the spatial attention map between the class prototype and the query instance. Unlike the abovementioned works that often face sample bias in 1-shot scenarios, the feature propagation in our framework always originates from query instances and therefore does not have such concerns.

Query based adaptation models. It is reported recently that, by simply adding query instance related conditional (and marginal) entropy terms in the fine tuning objective, Dhillon et al. (2020) and Boudiaf et al. (2020) achieve quite strong FSL performance. In principle, the entropy terms can also be incorporated into meta-learning based models for adaptive representation learning while introducing no extra parameters. Antoniou et al. (2019) propose a self-critique and adapt model that incorporates query instances into meta-learning by learning a label-free critic loss function through a neural network. In our work, we take a different path by modifying the features of each instance in a transductive way. We apply our framework on the model from (Boudiaf et al., 2020) and consistently improve its performance in both 1-shot and 5-shot scenarios.

Models with pseudo-labelled data. To alleviate sample bias, some models directly predict pseudo labels of unlabelled instances and select the top confident ones as the augmented training data. Liu et al. (2020) use the augmented labelled set to rectify class prototypes. Similarly, Ziko et al. (2020) formulate a graph clustering of the query set constrained by the labelled set supervision. Such labelled set supervision is partially contributed by the pseudo-labelled instances. Li et al. (2019) perform hard selection of the top confident predictions and meta-learn a soft weighting network to weight the augmented training data. Our proposed framework does not exploit pseudo labels but rather work in the feature representation space.

2.2. Parameter Efficient Few-Shot Learning

According to (Chen et al., 2019), FSL performance is proportional to the depth of feature extractor *to some extent* in a given architecture. Therefore the capacity of a novel FSL design should be separated from the feature extractor. For simplicity and computational benefit, it is preferable to have parameter efficient FSL architecture designs, which introduce none or fewer extra parameters. Below we briefly review FSL works that make effort in this direction.

Snell et al. (2017) and Finn et al. (2017) propose the

early influential models for their simple and effective designs. Although both models' only parameters belong to the feature extractor, with different designs in the training strategy and metric function, they have successfully pioneered the exploration of metric learning and optimization based meta learning. Dhillon et al. (2020) and Boudiaf et al. (2020) achieve very strong performance by simply adding the queries' entropy related terms in the objective without introducing any parameters. Sun et al. (2019) and Tseng et al. (2020) have proven the importance of feature transformation in FSL with strong performances at the cost of very limited parameters, due to the efficient channel-wise scaling and shifting. Bateni et al. (2020) improve upon the meta-learning method, CNAPS, from (Requeima et al., 2019) by introducing a class covariance based distance metric for a parameterless classifier design. This simpleCNAPS is not only more lightweight than CNAPS, but also performs better. Gidaris et al. (2019) explore applying self-supervised learning (SSL) to FSL, which justifies the benefit of flexible SSL manipulation in training data. Liu et al. (2020) train a cosine similarity based Prototypical network then apply prototype rectification. This parameterless formula effectively drives class prototype closer to the real prototype and alleviates sample bias. Similarly, LaplacianShot's whole parameters belong to feature extractor, while its optimization over the binary assignment function benefits from a closed form solution and efficiently outputs query label prediction (Ziko et al., 2020).

All these papers demonstrate good performance without relying on complicated architectures or (many) extra parameters. Our framework respects such a design philosophy and is categorized as a parameterless model. In all experiments, we simply apply our framework as a plug-in formula layer between the feature extractor and the classifier of the baselines. In principle, our framework is applicable to most existing FSL solutions, beyond the tested baselines.

3. Transductive Feature Re-representation

3.1. Problem Definition

In a N-way K-shot FSL classification task, we are given a support set $\mathbb{S} = \{(x_i, y_i)\}_{i=1}^{N\times K}$ and a query set $\mathbb{Q} = \{(x_i^*, y_i^*)\}_{i=1}^{N\times M}$. The support set includes K labelled instances from each of the N classes, typically referred to as novel classes, and is used to train the classification model. The query set includes M instances from each of the same N classes, whose labels are only used to test the classification model. In FSL, K is typically a small number between 1 and 5. Such small amount of training data is not sufficient to produce a good classifier, so we also assume to have access to an auxiliary base training set $\mathbb{D} = \{(x_i, y_i)\}_{i=1}^{N_D\times P}$, where each of the N_D base classes has sufficient number of labelled instances ($P \gg K$). This base training set has a disjoint label space from the target support and query sets. It is typically used to sample a set of N-way K-shot tasks for an episodic meta-training process, or conduct pre-training for fine tuning based FSL models.

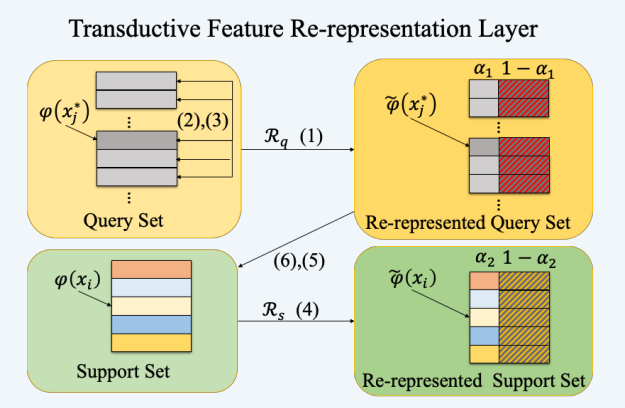

Figure 1. The schematic diagram of our proposed transductive feature re-representation layer, which can be deployed between the feature extractor and the classifier. (1)-(6) denote the corresponding equations in Section 3.2.1. Query set instances are first re-represented using Equation (2), (3) and (1). The re-represented query instances are then merged into support instances using Equation (6), (5) and (4). Hyperparameter α_1 and α_2 control the ratio between an instance's original features and the fused features.

3.2. Feature Re-representation Framework

The key challenge of FSL lies in the scarcity of the labeled data, which creates the sample bias issue. Our proposed framework aims at achieving transductive feature propagation to alleviate sample bias in FSL without introducing any trainable parameters. To achieve this goal, the framework includes two components: (1) a feature re-representation layer that consists of only a set of formulas, and (2) a self-supervised learning loss appended to the default training loss of any baseline models. The formula layer performs a two-step feature re-representation which respectively enriches the query and support instances' features to overcome the sample bias problem. The self-supervised learning loss is added to further facilitate representation learning by regularizing the feature re-representation. The overall framework can be either added as a plug-in layer on existing FSL models without retraining or integrated into the feature extraction process through meta-training or fine-tuning without additional training parameters.

3.2.1. Feature Re-representation Formula

Modern FSL frameworks typically consist of two major components: *backbone feature extractor* φ and *classifier* ϑ. The feature extractor φ can encode both the support instances in $\mathbb{S}$ and the query instances in $\mathbb{Q}$ into the same

embedding space: $\varphi : \mathcal{X} \rightarrow \Psi$. However, due to the scarcity of the labeled data, the few-shot support set can be hardly representative to the unlabeled data even in the embedding space Ψ, which consequently hampers the induction of an accurate classifier ϑ. To overcome this problem, we propose to transductively propagate information between the unlabeled query instances and from the query instances to the support instances with a two-step feature re-representation layer, which is illustrated in Figure 1.

First, we re-represent each query instance $x_j^* \in \mathbb{Q}$ by fusing its own features with a weighted sum of all the other query instances' features through an attention mechanism:

$$\begin{aligned}\widetilde{\varphi}(x_j^*) &= \mathcal{R}_q(\varphi(\mathbb{Q}), \varphi(x_j^*)) \\ &= (1-\alpha_1)\,\varphi(x_j^*) + \alpha_1 \sum_{j' \neq j, j' \in \mathbb{Q}} a_{jj'}^* \varphi(x_{j'}^*) \end{aligned} \quad (1)$$

where α_1 is a hyperparameter that controls the degree of re-representation. By keeping some degree, $1-\alpha_1$, of the original feature representation, we expect to maintain the uniqueness and original information of each instance while fusing information from others. The weight $a_{jj'}^*$ is the attention score on each query instance $x_{j'}^*$, which is computed based on the distance between $x_{j'}^*$ and x_j^*. Specifically, we compute the squared Euclidean distance between each pair of query instances in the embedding space Ψ, such as

$$d_{jj'}^* = \|\varphi(x_j^*) - \varphi(x_{j'}^*)\|_2^2. \quad (2)$$

The attention score $a_{jj'}^*$ can then be computed using a soft-max operation:

$$a_{jj'}^* = \frac{\exp(-\tau d_{jj'}^*)}{\sum_{k \neq j, k \in \mathbb{Q}} \exp(-\tau d_{jk}^*)}, \quad (3)$$

where τ is a hyperparameter that controls the sharpness of the attention score. As the attention scores are normalized similarity scores between a query instance x_j^* and all the other query instances, such attention based re-representation can help shift similar query instances closer to each other.

Next, we re-represent each support instance $x_i \in \mathbb{S}$ using a similar attention mechanism but by propagating information from the re-represented query set to the support set:

$$\begin{aligned}\widetilde{\varphi}(x_i) &= \mathcal{R}_s(\widetilde{\varphi}(\mathbb{Q}), \varphi(x_i)) \\ &= (1-\alpha_2)\,\varphi(x_i) + \alpha_2 \sum\nolimits_{j \in \mathbb{Q}} a_{ij} \widetilde{\varphi}(x_j^*) \end{aligned} \quad (4)$$

where α_2 is the hyperparameter that controls the degree of re-representation using the query data. The attention weight a_{ij} associated with the query instance x_j^* for the re-representation of the support instance x_i is computed in a similar way as above with a soft-max operation normalized over all the query instances:

$$a_{ij} = \frac{\exp(-\tau d_{ij})}{\sum_{j' \in \mathbb{Q}} \exp(-\tau d_{ij'})} \quad (5)$$

The squared Euclidean distance d_{ij} however is computed between the embedding features of the support instance x_i and the re-represented features of the query instance x_j^*, such as

$$d_{ij} = \|\varphi(x_i) - \widetilde{\varphi}(x_j^*)\|_2^2. \quad (6)$$

The feature re-representation process above benefits few-shot learning from the representation perspective in twofold. First, by fusing information across the query set with the attention mechanism, similar instances will be driven closer to each other to induce better separated clusters. Second, by further propagating the information of unlabeled query instances into the labeled support instances via feature re-representation, the representation of the scarce support instances can be rectified to better represent the class prototypes, and hence effectively reduce sample bias. Therefore, a classifier induced by the re-represented support instances is expected to have much better generalization capacity. Moreover, it is worth noting the proposed re-representation scheme has wide applicability and a unique lightweight design. It can be incorporated into many existing FSL models simply as an additional layer between the feature extractor and the classifier, without re-training requisitions.

3.2.2. Auxiliary Self-Supervised Learning

The proposed re-representation framework is not limited to being used as a plug-in layer for one-time forward pass during inference. Instead it can be integrated into the meta-training or fine-tuning process as well. To facilitate model training with the re-representation layer, we propose to incorporate a self-supervised learning (SSL) loss into the training process to regularize the feature re-representation.

The idea is that the re-representation of each instance is effectively a transformation of its original features, which resembles the application of a transformation function on instances in standard self-supervised learning (Chen et al., 2020). By letting the model recognize the pairs of original and the re-represented instances, self-supervised learning can work together with the re-representation layer to help learn a good feature extractor. Specifically, the re-represented features $\widetilde{\varphi}(x_i)$ and the original features $\varphi(x_i)$ can be treated as two views of an instance x_i in the same feature space. We then adopt a contrastive learning scheme to enforce the similarity between different views of the same instance under a cross-view contrastive loss.

We in particular focus on the support instances as they have completed the proposed two-step re-representation and fused information from all the query instances. For each support instance x_i, we calculate the squared Euclidean distance between its re-represented view $\widetilde{\varphi}(x_i)$ and the original view $\varphi(x_{i'})$ of any support instance $x_{i'}$, including itself:

$$d_{ii'}^{ssl} = \|\widetilde{\varphi}(x_i) - \varphi(x_{i'})\|_2^2. \quad (7)$$

Table 1. A general categorization of many existing FSL models and the potential deployment of our transductive re-representation formulas (RR) and SSL formulation on them. We use the following abbreviations. (1) Classifier: classifier property. (N)P: (non-)parametric. M: metric-based. C: multi-way classifier. (2) BT: base set training. SQT: target support and query set training. φ: feature extractor. ϑ: parametric classifier. ϑ_d: classifier only for pre-training on $\mathbb{D}$. The last two columns summarize the applied components of our proposed framework on base class set, $\mathbb{D}$, and target support and query sets, $\mathbb{S}$ and $\mathbb{Q}$. References for the these FSL models are given in Section 2 and Table 2 & Table 3 in Section 4.

Categorization	Classifier	BT Object	SQT Object	Example	$\mathbb{D}$	$\mathbb{S}$, $\mathbb{Q}$
Meta-learning	P,M	φ,ϑ	-	**Relation/Matching/**TPN	RR,SSL	RR
	NP,M	φ	-	**ProtoNet/**ProtoRectify		
	P,C	φ,ϑ	ϑ	LEO/CNAPS/MTL		
	NP,C	φ	-	simpleCNAPS		
non-Meta-learning	P,C	φ,ϑ_d	ϑ	**TIM/**baseline/baseline++	-	RR
	P,C	φ,ϑ_d	φ,ϑ	Ent-min	-	RR,SSL
	NP,M	φ,ϑ_d	-	**SimpleShot/LaplacianShot**	-	RR

We assume under an effective feature extraction, the re-representation of an instance should still be most similar to itself in the original view than all the other instances; that is, the following equality should hold:

$$d^{ssl}_{ii} = \underset{i' \in \mathbb{S}}{\operatorname{argmin}}\, d^{ssl}_{ii'}. \tag{8}$$

This can be encoded using the following contrastive loss under a cross-view soft-max operation on the support set:

$$L^{ssl} = -\mathbb{E}_{i \in \mathcal{S}} \left[\log \frac{\exp(-d^{ssl}_{ii})}{\sum_{i' \in \mathbb{S}} \exp(-d^{ssl}_{ii'})} \right]. \tag{9}$$

This self-supervised learning loss can be added into the default training loss L^0 of the FSL model as an auxiliary regularization term:

$$L^{tot} = L^0 + L^{ssl}. \tag{10}$$

We expect such a self-supervised learning augmented loss can facilitate learning a more effective feature extractor.

3.3. Applicability

The proposed feature re-representation framework can be applied on many existing FSL models, either by applying the re-representation formulas (RR) during inference or being deployed during training together with the self-supervised learning loss (SSL). To illustrate the wide applicability of the proposed framework, in Table 1 we summarize the potential deployment of the proposed framework on many existing FSL works in two general categories: meta-learning models and non-meta-learning models. The former category performs episodic meta-training on base class set $\mathbb{D}$ and the latter typically uses $\mathbb{D}$ to pre-train the feature extractor. The models in each category are further grouped by: (1) the classifier properties (parametric vs non-parametric, metric based generalizable classifier vs task specific N-way classifier), (2) the objects trained on base class set $\mathbb{D}$ (BT Object), and (3) the objects trained (or generated) on the target support set $\mathbb{S}$ and/or query set $\mathbb{Q}$ (SQT Object). The last two columns in Table 1 summarize the deployment of our framework on base class set $\mathbb{D}$, the target support and query sets $\mathbb{S}$ and $\mathbb{Q}$. The meta-learning models can use re-representation (RR) and SSL together during meta-training. Since their feature extractors are not adapted during meta-testing, only RR is applied **once** for inference. For the non-meta-leaning models, we only apply our model components on the target support and query sets. If the feature extractor φ is not fine tuned on $\mathbb{S}$ and $\mathbb{Q}$, we only apply RR once for inference. If φ is fine tuned as in (Dhillon et al., 2020), both RR and SSL can be involved in each update step of the feature extractor.

4. Experiments

We conducted extensive experiments by applying the proposed framework to different FSL models. In this section, we report our experimental setup and results.

4.1. Experiment Setup

Datasets. We conducted experiments on three FSL benchmark datasets: *mini*-ImageNet (Ravi & Larochelle, 2016), *tiered*-ImageNet (Ren et al., 2018) and CUB (Welinder et al., 2010). We follow the train/validation/test split configuration in (Ravi & Larochelle, 2016; Ren et al., 2018; Chen et al., 2019) for the three datasets respectively and report the average test results over multiple runs.

Comparison setup. We apply our framework to three well-known meta-learning models, *Protypical network*, *Matching network* and *Relation network*, and three recently developed state-of-the-art non-meta learning FSL models, *SimpleShot* (Wang et al., 2019), *LaplacianShot* (Ziko et al., 2020) and *transductive information maximization (TIM)* (Boudiaf et al., 2020). For the meta-learning models, our proposed framework is experimented in two scenarios. First,

Table 2. FSL testing accuracies on *mini*-ImageNet by applying transductive setting on three meta-learning baselines: ProtoNet (Snell et al., 2017), Matching (Vinyals et al., 2016) and Relation (Sung et al., 2018). RR: re-representation. MT: meta-learning with re-representation layer applied. T: Transductive. The best performances are highlighted in **bold**.

Method	T	Backbone	1-shot	5-shot
ProtoNet	×	ResNet-18	54.76	72.82
ProtoNet+RR(ours)	√	ResNet-18	59.02	74.76
ProtoNet+MT(ours)	√	ResNet-18	**59.06**	**74.92**
Matching	×	ResNet-18	53.67	68.85
Matching+RR(ours)	√	ResNet-18	55.57	70.78
Matching+MT(ours)	√	ResNet-18	**56.68**	**71.55**
Relation	×	ResNet-18	52.87	68.01
Relation+RR(ours)	√	ResNet-18	55.38	69.34
Relation+MT(ours)	√	ResNet-18	**55.70**	**70.29**

re-representation is applied once only at the inference time to the meta-trained feature extractor without retraining. We denote this scenario as **RR**. Second, re-representation is involved in meta-training together with the SSL loss. We denote this scenario as **MT**. For the three non-meta-learning FSL models, since the feature extractor is fixed after the base class training, we apply the re-representation formulas once on the extracted features.

We choose ResNet-18 (He et al., 2016) and WRN28-10 (Zagoruyko & Komodakis, 2016) as the feature extractors. For meta-learning based experiments under the second scenario, we use the trained base model as the starting model for training with the re-representation layer. This can make the transductive training more efficient.

Hyperparameters. We keep the default hyperparameter setup of all the reference models when applying the proposed framework, with one exception in the meta-learning baselines: the learning rate. We find that 5×10^{-5} generally works best for all baseline model training. When the re-representation layer is enabled, we fix the learning rate to 10^{-5}. The key hyperparameters in our proposed framework are α_1, α_2 and τ. Their values are selected using the validation split of the corresponding datasets.

4.2. Comparison Results

Comparison results with meta-learning base models. For the three well-known meta-learning base models, Protonetwork, Matching network, and Relation network, we deployed our proposed network in two ways, **RR** and **MT**. The comparison results on mini-ImageNet are reported in Table 2. We can see that even by only applying the re-representation formulas during inference (**RR**), the proposed framework consistently produces $2\%\sim4\%$ accuracy gain on all models in the most challenging 1-shot learning cases. In the cases of 5-shot learning, the performance gain is also in the range of $1.3\%\sim1.9\%$. We can credit such improvements to a rectified feature distribution: by aggregating information from query to support set, all labelled and unlabelled instances are driven closer to their better represented class prototypes. This is most important in 1-shot scenarios where the severe sample bias can be significantly reduced. By further incorporating our framework into meta-training (**MT**) through the self-supervised loss, slight but consistent improvements are gained over the **RR** version.

Comparison results with state-of-the-art FSL methods. In Table 3, we have included the 1-shot and 5-shot results reported by various transductive and inductive state-of-the-art FSL works. By deploying our proposed re-representation framework on three of these FSL models, we obtain three new methods, *SimpleShot+RR*, *LaplacianShot+RR*, and *TIM+RR*. We compared them with their baseline models and all the other methods in Table 3 on the three benchmark datasets. There are a few remarkable observations. First, our proposed framework consistently improves each of its corresponding base models across all test cases. In particular, SimpleShot+RR gains approximately $7\% \sim 9\%$ accuracy improvements in 1-shot tests with both backbones across the three datasets. LaplacianShot+RR yields more than 2% performance gains over its base model in 1-shot tests, and outperforms TIM, which provides the best results among the other comparison methods from the literature, in most 1-shot tests. Meanwhile, TIM+RR further yields more than 2% performance gains over TIM in all 1-shot tests. These strong performance gains again verified the effectiveness of the proposed simple re-representation framework in reducing sample bias in 1-shot learning. In contrast, the improvements on 5-shot tests are only marginal. This is within expectation: as more labelled training data is available, model training is less prone to sample bias and requires less feature propagation from the query instances. Second, when applied to TIM, our lightweight re-representation framework, TIM+RR, achieves the state-of-the-art best results across all tests but one — in 5-shot test on CUB with ResNet-18, DPGN performs slightly better. All these results validated the great capacity of the proposed transductive feature re-representation framework for FSL.

For completeness, we also conducted higher way FSL tests on *mini*-ImageNet using SimpleShot+RR, LaplacianShot+RR and TIM+RR. The results are reported in Table 4, and similarly demonstrate the proposed framework's effectiveness in alleviating sample bias when there are fewer labeled support instances, especially in the 1-shot cases.

4.3. Ablation Study

To verify the functionality of each design component, we conduct an ablation study to compare the full frame-

Table 3. Comparison results of the proposed re-representation framework (RR) with the state-of-the-art and related works on three benchmark FSL datasets. The best performances in each group are highlighted in **bold**.

Method	Transductive	Backbone	*mini*-ImageNet		*tiered*-ImageNet		CUB	
			1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MAML (Finn et al., 2017)	×	ResNet-18	49.61	65.72	-	-	68.42	83.47
baseline (Chen et al., 2019)	×	ResNet-18	51.75	74.27	-	-	65.51	82.85
baseline++ (Chen et al., 2019)	×	ResNet-18	51.87	75.68	-	-	67.02	83.58
TPN (Liu et al., 2019)	√	ResNet-12	59.46	75.65	-	-	-	-
MTL (Sun et al., 2019)	×	ResNet-12	61.2	75.5	-	-	-	-
Ent-min (Dhillon et al., 2020)	√	ResNet-12	62.35	74.53	68.41	83.41	-	-
CAN+T (Hou et al., 2019)	√	ResNet-12	67.19	80.64	73.21	84.93	-	-
DPGN (Yang et al., 2020)	√	ResNet-12	67.77	84.60	72.45	87.24	75.71	**91.48**
SimpleShot (Wang et al., 2019)	×	ResNet-18	63.07	80.00	69.32	84.81	70.22	86.44
SimpleShot+RR(ours)	√	ResNet-18	70.25	81.92	77.11	86.30	79.60	88.38
LaplacianShot (Ziko et al., 2020)	√	ResNet-18	72.29	82.38	78.95	86.34	80.74	88.71
LaplacianShot+RR(ours)	√	ResNet-18	75.04	82.71	81.43	86.73	83.55	89.00
TIM (Boudiaf et al., 2020)	√	ResNet-18	73.92	85.04	79.94	88.53	82.19	90.79
TIM+RR(ours)	√	ResNet-18	**76.54**	**85.20**	**82.58**	**88.68**	**85.36**	90.99
LEO (Rusu et al., 2019)	×	WRN28-10	61.76	77.59	66.33	81.44	-	-
Ent-min (Dhillon et al., 2020)	√	WRN28-10	65.73	78.40	73.34	85.50	-	-
ProtoRectify (Liu et al., 2020)	√	WRN28-10	70.31	81.89	78.74	86.92	-	-
SimpleShot (Wang et al., 2019)	×	WRN28-10	63.32	80.28	69.98	85.45	74.47	89.74
SimpleShot+RR(ours)	√	WRN28-10	70.23	81.90	78.30	87.13	84.33	91.43
LaplacianShot (Ziko et al., 2020)	√	WRN28-10	74.90	84.07	80.22	87.49	84.94	91.71
LaplacianShot+RR(ours)	√	WRN28-10	77.63	84.27	82.95	87.94	87.62	91.92
TIM (Boudiaf et al., 2020)	√	WRN28-10	77.80	87.39	82.08	89.85	86.98	93.70
TIM+RR(ours)	√	WRN28-10	**80.04**	**87.64**	**84.30**	**90.01**	**89.78**	**93.93**

Table 4. Test results with higher number of classes on *mini*-ImageNet with ResNet-18 as backbone. Column header (N,K) denotes N way K shot.

Method	(10,1)	(10,5)	(20,1)	(20,5)
SimpleShot	47.40	67.86	34.02	55.17
SimpleShot+RR	52.28	69.86	36.58	55.59
LaplacianShot	55.56	69.73	41.22	55.93
LaplacianShot+RR	**58.42**	70.00	**42.47**	55.66
TIM	55.83	73.18	39.08	59.37
TIM+RR	57.72	**73.50**	39.76	**59.66**

Table 5. Ablation study of individual components of our framework by disabling α_1, α_2 and SSL respectively on meta-learning baselines. Column header (N,K) denotes N way K shot. Tests are on *mini*-ImageNet with ResNet-18 as backbone.

Method	ProtoNet+MT		Matching+MT		Relation+MT	
	(5,1)	(5,5)	(5,1)	(5,5)	(5,1)	(5,5)
All	**59.06**	**74.92**	**56.68**	**71.55**	**55.70**	**70.29**
No α_1	57.86	74.27	55.04	70.01	54.30	69.35
No α_2	55.09	73.33	54.13	69.72	53.89	68.78
No SSL	58.85	74.84	55.56	70.26	55.70	69.74
Baseline	54.76	72.82	53.67	68.85	52.87	68.01

work with three variants produced by dropping the two re-representation steps and SSL respectively. By setting $\alpha_1 = 0$, the information propagation between query instances is dropped; similarly the information propagation from query to support instances is dropped by setting $\alpha_2 = 0$. The ablation study is conducted with both the three meta-learning baselines and the three non-meta learning FSL models used in previous experiments. The two sets of results are presented in Table 5 and Table 6 respectively.

We can see that when deploying the framework on the three meta-learning baselines and SimpleShot, dropping either α_1 (via $\alpha_1 = 0$) or α_2 (via $\alpha_2 = 0$) will significantly degrade the performance. This suggests that when the classifier is not fine tuned or adapted, re-representation is critical for both the query and support instances. Meanwhile, we also notice that dropping α_2 hurts the performance more than dropping α_1, which suggests it is more important to tackle the sample bias problem by aggregating unlabeled data into the support set for inductive base models. The impact of SSL is marginal but consistently positive as discussed previously.

The ablation results on the transductive base models LaplacianShot and TIM are somehow different. Although α_2 is still helpful, particularly in 1-shot tests, its impact is less remarkable than for meta-learning baselines and SimpleShot. On TIM+RR, the removal of α_1 is more detrimental than

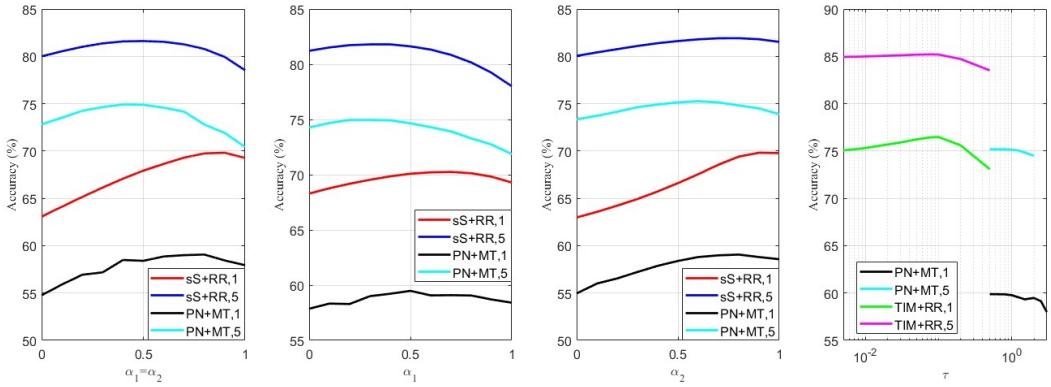

Figure 2. Test accuracy as a function of the key hyperparameters. From left to right, we vary the following hyperparameters while fixing the rest: (1) $\alpha_1 = \alpha_2$, (2) α_1, (3) α_2, and (4) τ. sS: SimpleShot. PN: ProtoNet. Both 1-shot and 5-shots are illustrated in each figure.

Table 6. Ablation study of individual components of our framework by disabling α_1, α_2 on non-meta-learning baselines. Column header (N,K) denotes N way K shot. The three subgroups belong to tests on *mini*-ImageNet, *tiered*-ImageNet and CUB respectively with ResNet-18 as backbone.

Method	SimpleShot+RR		LaplacianShot+RR		TIM+RR	
	(5,1)	(5,5)	(5,1)	(5,5)	(5,1)	(5,5)
All	**70.25**	**81.92**	**75.04**	**82.71**	**76.54**	**85.20**
No α_1	68.29	81.15	74.26	82.67	74.93	84.87
No α_2	64.14	80.05	72.85	82.41	76.00	85.20
Baseline	63.07	80.00	72.29	82.38	73.92	85.04
All	**77.11**	**86.30**	**81.43**	**86.73**	**82.58**	**88.68**
No α_1	75.41	85.69	80.56	86.65	81.06	88.48
No α_2	69.44	84.42	79.94	86.62	82.20	88.63
Baseline	69.32	84.81	78.95	86.34	79.94	88.53
All	**79.60**	**88.38**	**83.55**	**89.00**	**85.36**	**90.99**
No α_1	76.76	87.34	82.41	88.94	83.74	90.76
No α_2	73.35	87.07	82.22	88.84	84.80	90.95
Baseline	70.22	86.44	80.74	88.71	82.19	90.79

omitting α_2. This is due to the fact that TIM is a transductive model with fine-tuning, which can exploit the query data in its own way. Nevertheless, the proposed framework overall still improves their performance.

4.4. Impact of Hyperparameters

We have also conducted experiments to investigate the impact of the three hyperparameters of the proposed re-representation framework: α_1, α_2, and τ. Specifically, we test how varying each hyperparameter affects the model performance while the other hyperparameters are fixed at their chosen values. The four subfigures in Figure 2 show the test results when the varying hyperparameters are (1) $\alpha_1 = \alpha_2$, (2) α_1, (3) α_2, and (4) τ, respectively. We have the following observations from the first three subfigures. (1) With ProtoNet and SimpleShot as base inductive models, 1-shot tests are more sensitive to α_2 than α_1. (2) It is typically beneficial to receive more query information in 1-shot tests by increasing α_1 and/or α_2 than in 5-shot tests. (3) It is important to keep a portion of the original features by keeping $\alpha_1 < 1$ and $\alpha_2 < 1$. In the fourth subfigure, we vary the value of τ to illustrate the role of sharpness control in computing the attention scores in Eq.(3) and Eq.(5). A very small τ value induces towards average attention scores, while a sufficient large τ will focus the attention only on a small number of closest neighbours. Such usage is similar to the temperature in a Gumbel-Softmax function. With a large range of values, the sensitivity curve of τ tends to have a single modal peak such as the green "TIM+RR,1" curve.

4.5. Results with Varying Support and Query Shots

To study the sensitivity of the proposed framework with respect to the amount of instances in the support set and query set, we conduct 5-way FSL experiments by applying RR on two baseline methods, TIM and SimpleShot, in the following two settings: (1) vary the support shot – the number of instances per class in the support set, while fixing the query set size; and (2) vary the query shot – the number of instances per class in the query set, while fixing the support set size. For the varying support shot experiments, the query shot is fixed at 15, and support shots are varied from 1 through 10. The average test results with different baselines, TIM and SimpleShot, on the three datasets are reported in Figure 3. We can see that RR again consistently improves the baseline performance across various support shots. However, such improvements begin to saturate with higher support shots, where sample bias is greatly reduced. This suggests RR brings most performance gains when sample bias is significant in few-shot tests.

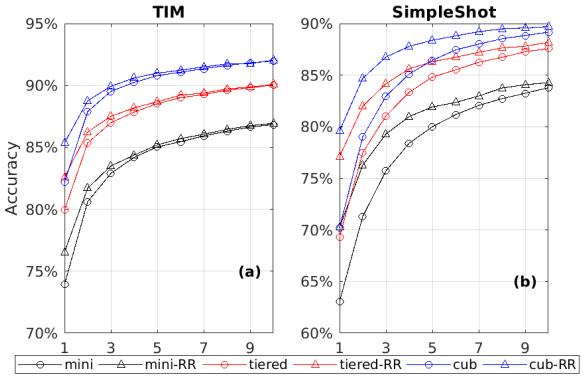

Figure 3. 5-way test accuracies with varying support shots. mini: *mini*-ImageNet. tiered: *tiered*-ImageNet.

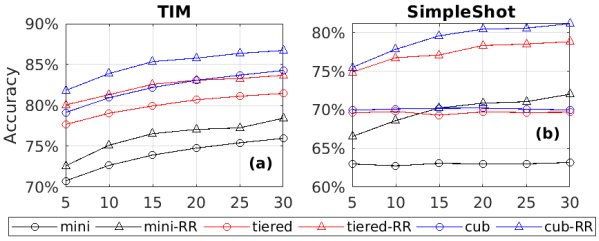

Figure 4. 5-way 1-shot test accuracies with varying query shots. mini: *mini*-ImageNet. tiered: *tiered*-ImageNet.

For the varying query shot experiments, we fix the support shot at 1 and vary query shots from 5 to 30. The corresponding results are presented in Figure 4. We can see that the transductive baseline, TIM, benefits more from higher query shots than the inductive baseline, SimpleShot. But when equipped with RR, both TIM+RR and SimpleShot+RR achieve better performances across all datasets and query shots over the corresponding baselines, while yielding better test accuracies with higher query shots. This indicates that it is beneficial to propagate more valid and available query information. Overall, these two sets of experiments validated that the proposed framework works well in few-shot scenarios where sample bias is significant, and performs positively related to the query set size.

4.6. Results with Varying RR Iterations

In the experiments above we only apply one iteration of RR during inference. To investigate whether better performance can be achieved via multiple RR iterations, we conduct experiments with various RR iterations during inference. Specifically, starting from iteration 2, each RR iteration uses the re-represented (query and support) features from the previous iteration as the original instance features and re-apply the feature re-representation formulas. The experiments are conducted with 5-way 1-shot and 5-shot tests on *mini*-ImageNet, *tiered*-ImageNet and CUB, by using TIM and SimpleShot as the baseline methods. We vary the number of RR iterations from 0 to 4, where 0 means no RR. The results based on the two baselines are reported in the two subfigures of Figure 5 respectively. We can see that the best performances are always achieved when just deploying one RR iteration. Our hypothesis is that although propagating query information is beneficial, it is also important to keep a portion of the original features. Running multiple RR iterations may dilute the original features to an extent that transduction starts to harm the performance.

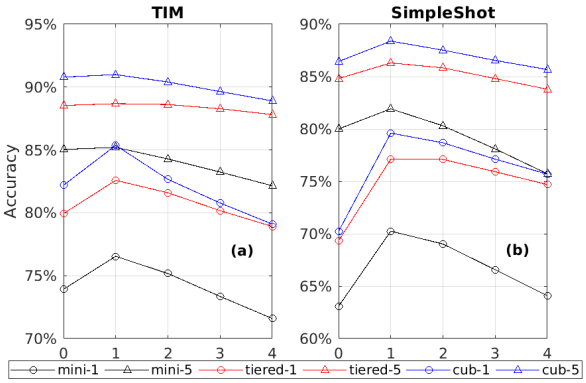

Figure 5. Test accuracies of applying RR various iterations during inference. mini: *mini*-ImageNet. tiered: *tiered*-ImageNet.

5. Conclusions

In this paper, we propose a lightweight two-step transductive feature re-representation framework to alleviate sample bias in FSL. In the first step, each query instance is re-represented by merging a weighted sum of other query instances with itself. In the second step, each support instance is re-represented similarly by aggregating information from the re-represented query instances. The framework has wide applicability, and can be deployed as a simple plug-in layer between the feature extractor and classifier on most existing FSL models. It can also be integrated into the training process of the base models with an auxiliary SSL loss. We conducted extensive experiments to validate the proposed framework. The empirical results show that the proposed framework consistently improves both meta-learning and non-meta-learning FSL models, and refreshes the state-of-the-art performance on benchmark FSL datasets, especially in the most challenging 1-shot scenarios.

Acknowledgements

This research was supported in part by the NSERC Discovery Grant, the Canada Research Chairs Program, and the Canada CIFAR AI Chairs Program.

References

Antoniou, A. and Storkey, A. J. Learning to learn by self-critique. In *NeurIPS*, 2019.

Bateni, P., Goyal, R., Masrani, V., Wood, F., and Sigal, L. Improved few-shot visual classification. In *CVPR*, 2020.

Boudiaf, M., Masud, Z. I., Rony, J., Dolz, J., Piantanida, P., and Ayed, I. B. Transductive information maximization for few-shot learning. In *NeurIPS*, 2020.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., and Huang, J.-B. A closer look at few-shot classification. In *ICLR*, 2019.

Dhillon, G. S., Chaudhari, P., Ravichandran, A., and Soatto, S. A baseline for few-shot image classification. In *ICLR*, 2020.

Doersch, C., Gupta, A., and Zisserman, A. Crosstransformers: spatially-aware few-shot transfer. In *NeurIPS*, 2020.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

Gidaris, S., Bursuc, A., Komodakis, N., Perez, P., and Cord, M. Boosting few-shot visual learning with self-supervision. In *ICCV*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hou, R., Chang, H., Ma, B., Shan, S., and Chen, X. Cross attention network for few-shot classification. In *NeurIPS*, 2019.

Kim, J., Kim, T., Kim, S., and Yoo, C. D. Edge-labeling graph neural network for few-shot learning. In *CVPR*, 2019.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015.

Li, X., Sun, Q., Liu, Y., Zhou, Q., Zheng, S., Chua, T.-S., and Schiele, B. Learning to self-train for semi-supervised few-shot classification. In *NeurIPS*, 2019.

Liu, J., Song, L., and Qin, Y. Prototype rectification for few-shot learning. In *ECCV*, 2020.

Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S., and Yang, Y. Learning to propagate labels: Transductive propagation network for few-shot learning. In *ICLR*, 2019.

Qiao, L., Shi, Y., Li, J., Wang, Y., Huang, T., and Tian, Y. Transductive episodic-wise adaptive metric for few-shot learning. In *ICCV*, 2019.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *ICLR*, 2016.

Ren, M., Ravi, S., Triantafillou, E., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., and Zemel, R. S. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018.

Requeima, J., Gordon, J., Bronskill, J., Nowozin, S., and Turner, R. E. Fast and flexible multi-task classification using conditional neural adaptive processes. In *NeurIPS*, 2019.

Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., and Hadsell, R. Meta-learning with latent embedding optimization. In *ICLR*, 2019.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

Sun, Q., Liu, Y., Chua, T., and Schiele, B. Meta-transfer learning for few-shot learning. In *CVPR*, 2019.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

Tseng, H.-Y., Lee, H.-Y., Huang, J.-B., and Yang, M.-H. Cross-domain few-shot classification via learned feature-wise transformation. In *ICLR*, 2020.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *NeurIPS*, 2016.

Wang, Y., Chao, W.-L., Weinberger, K. Q., and van der Maaten, L. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Yang, L., Li, L., Zhang, Z., Zhou, X., Zhou, E., and Liu, Y. Dpgn: Distribution propagation graph network for few-shot learning. In *CVPR*, 2020.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Ziko, I., Dolz, J., Granger, E., and Ayed, I. B. Laplacian regularized few-shot learning. In *ICML*, 2020.