Class-Specific Channel Attention for Few-Shot Learning

Ying-Yu Chen¹, Jun-Wei Hsieh², Ming-Ching Chang³

¹,²National Yang Ming Chiao Tung University
³University at Albany, SUNY

Abstract

Few-Shot Learning (FSL) has attracted growing attention in computer vision due to its capability in model training without the need for excessive data. FSL is challenging because the training and testing categories (the base vs. novel sets) can be largely diversified. Conventional transfer-based solutions that aim to transfer knowledge learned from large labeled training sets to target testing sets are limited, as critical adverse impacts of the shift in task distribution are not adequately addressed. In this paper, we extend the solution of transfer-based methods by incorporating the concept of metric-learning and channel attention. To better exploit the feature representations extracted by the feature backbone, we propose Class-Specific Channel Attention (CSCA) module, which learns to highlight the discriminative channels in each class by assigning each class one CSCA weight vector. Unlike general attention modules designed to learn global-class features, the CSCA module aims to learn local and class-specific features with very effective computation. We evaluated the performance of the CSCA module on standard benchmarks including miniImagenet, Tiered-ImageNet, CIFAR-FS, and CUB-200-2011. Experiments are performed in inductive and in/cross-domain settings. We achieve new state-of-the-art results. Our codes and models are available at https://github.com/YingYuChen97/CSCA.

Introduction

Deep Learning and Convolutional Neural Networks (CNNs) have achieved great progress in a wide range of tasks including image classification, objection detection/segmentation, and many others, where large and well-annotated datasets are available. However, supervised deep learning faces challenges in real-world usage scenarios, as it is expensive or sometimes even impractical to collect large enough data for sufficient model training. Optimizing deep CNNs built with parameters much larger than the training samples can lead to catastrophic overfitting. To this end, Few-Shot Learning (FSL) (Wang et al. 2020b) is a promising remedy under active development that can transfer the learned efforts across distinct domains using very few annotated data. FSL can be solved using meta-learning and pretraining following the fine-tuning paradigm. The FSL learner is trained on a set of general base categories that can be generalized to a few unseen but related test categories, after tuning with only a few examples. FSL methods can be divided into two types: (1) inductive few-shot, where each prediction is made independently, and (2) transductive few-shot, which performs the prediction taking into account the relationships among all labeled and unlabeled samples in a batch.

In recent years, many methods have been proposed to solve FSL, among which transfer-based methods (Shalam and Korman 2022; Hu, Pateux, and Gripon 2022; Hu, Gripon, and Pateux 2021; Bendou et al. 2022) have been shown to achieve state-of-the-art performance. To learn from a few samples, transfer-learning aims at transferring the knowledge learned from a large labeled dataset (base dataset) to the unlabeled target dataset (novel dataset). The above methods all claim that the main problem is the skewed distribution of the feature vectors extracted using the pretrained feature backbone. To better leverage the quality of features, they preprocess the feature vectors with the PCA-like transform and the Power transform to fit a particular distribution (i.e. Gaussian-like), then perform optimal trans-
port (Villani 2009) to solve the classification problem. Even if all of the above methods improve the quality of the extracted representations, they overlook the task distribution shift, which we consider to be a critical adverse impact in FSL.

The task distribution shift is an important problem related to training scalability that is often overlooked in FSL, where the training dataset can be completely disjoint or across different domains w.r.t. the target set at the time of test. The channel bias is a crucial problem in the learned image representations (i.e. features) regarding the task distribution shift. Specifically, in the penultimate layer after global average pooling in the feature backbone, different channels represent different features in the image. Each channel captures the importance of the features in the training data. However, when the model is transferred to an unseen test data in FSL, the feature backbone does not possess the knowledge of the importance to each channel (feature), which leads to an imprecise attention in the channels. For example, for categories having images with overlapped features such as trifle and school bus in a miniImagenet FSL task in Figure 1, these categories are assigned to the novel class, leading to the situation that the feature backbone may focus only on the shared indiscriminate feature person but disregard the discriminative information of each category.

To address the task distribution shift problem, (Luo, Xu, and Xu 2022) improve the performance by applying a channel-wise transformation equation to each channel. Despite the effectiveness of the transformation equation, we argue that using a single global transformation for all object categories is not ideal. We believe that class-specific channel attention has its own importance and can be leveraged in FSL. Therefore, we develop an efficient transformation module named Class-Specific Channel Attention (CSCA) depicted in Figure 2. The CSCA module assigns one channel attention weight vector to each class, which is trained to highlight important and discriminative channels for each class. Meanwhile, each channel attention weight vector consists of very few parameters and requires very few computations. We achieved significant experimental results while evaluating the CSCA module on various datasets with in-domain and cross-domain FSL settings.

The contribution of this paper includes the following:

- We propose a novel class-specific attention module that can point out each channel’s importance in each class with few computations.
- The performance of our method tested on numerous standard benchmarks and also with cross-domain settings, which almost hits the ceiling shows that a well-pretrained feature backbone can be well adapted to novel data with task distribution shift, without training on the target data.
- Our method achieves new state-of-the-art FSL results in miniImagenet, Tiered-ImageNet, CIFAR-FS, and CUB-200-2011 with inductive setting.

**Related Works**

**Metric Learning** (Xing et al. 2002; Kaya and Bilge 2019) aims at learning a feature backbone that maps input data to a high-dimensional feature space while preserving the similarity among data in the same classes. The similarity between two feature vectors is obtained by adopting distance metrics such as Cosine Distance or Mahalanobis Distance (Mahalanobis 1936). Metric learning is widely used in FSL (Vinyals et al. 2016; Sung et al. 2018). The Prototypical Network (ProtoNet) (Snell, Swersky, and Zemel 2017) is one of the well-known metric-based FSL methods. ProtoNet constructs prototypes (centroids) by averaging each channel in all high-dimensional feature vectors per class. The feature backbone is trained to center the feature vectors on the prototypes of each class. For evaluation, the feature backbone maps the query set to the high-dimensional feature space and performs nearest-centroid classification.

**Task Distribution Shift.** The category shift, domain shift, and granularity shift are the three main types of task distribution shifts between training and testing data. Category shift exists in datasets while the categories are disjoint between training and testing sets, such as miniImagenet (Vinyals et al. 2016) and CIFAR-FS (Bertinetto et al. 2018). Several recent benchmarks, including BSCD-FSL (Guo et al. 2020) and Meta-dataset (Triantafillou et al. 2019) based on domain shift, have attracted research attention in cross-domain few-shot learning. The granularity shift is also known as Coarse-to-Fine Few-Shot Learning (C2FS) (Luo et al. 2021; Bukchin et al. 2021; Yang, Yang, and Chen 2021), where models are trained on coarsely labeled images and tested on a few-shot task to differentiate fine-grained subclasses of training classes. In this paper, our aim is to solve the category and domain shift problems.

**Attention Mechanism.** Attention is a data processing method that is widely used in various types of machine learning tasks such as natural language processing (Galassi, Lippi, and Torroni 2020), image recognition (Guo et al. 2022), and speech recognition (Karmakar, Teng, and Lu 2021). Attention is essentially similar to the human observation mechanism for external things. Generally speaking, when people observe an unseen object, they first pay more attention to a certain discriminative local information; they then combine information from different areas to form an overall impression of the observed scene. By helping models assign different weights to each part of the input, the attention mechanism helps to extract more critical information, without much computation and model storage.

It is well known that attention is useful for strengthening convolutional neural networks in many works. These efforts can be organized into two trends: (1) feature aggregation and (2) channel/spatial attention. SE-Net (Hu, Shen, and Sun 2018) takes the lead in demonstrating the efficiency of channel attention. cSE, sSE, and scSE (Roy, Navab, and Wachinger 2018) modify SE-Net to further enhance meaningful features while suppressing other features. ECA (Wang et al. 2020a) is also based on SE-Net in learning effective channel attention with low model complexity. CBAM (Woo et al. 2018) aggregates features with the combination of max- and average-pooling. GSoP (Gao et al. 2019) brings in global second-order pooling in producing covariance matrices as image representation to achieve effective feature aggregation.
extract features from local spatial locations and “Excite” to restore the features to the original scale. This process is similar to the encoder-decoder module. The Non-Local (NL) neural network (Wang et al. 2018b) expands the receptive field to aggregate wider information. GCNet (Cao et al. 2019) combines SENet and the concept of NL to achieve an efficient and lightweight attention module. The Double Attention Networks of $A^2$-net (Chen et al. 2018) develop a generic function to collect global features and distribute them to the local space.

All attention methods surveyed above aim at developing sophisticated attention architectures that manage to improve the learning of important global information across all classes. In contrast, our CSCA module targets at the learning of important local information within each class. The CSCA module requires only few computations, but it can greatly boost the FSL performance.

Large Margin Cosine Loss. CosFace (Wang et al. 2018a) is a typical solution for deep face recognition. The central task of face recognition involves face feature discrimination. Faces are harder to classify than general classification datasets because the variance of face features among persons is less obvious. Recent advanced solutions such as center loss (Wen et al. 2016), large margin softmax loss (Liu et al. 2016), and angular softmax loss (Liu et al. 2017) share the same core idea in maximizing inter-class variation and minimizing intra-class variation. CosFace reformulates the softmax loss as a cosine loss with a newly introduced cosine margin, $N$ is the number of training samples, $x_i$ is the $i$th feature vector corresponding to the ground-truth class of $y_i$, $W_j$ is the weight vector of the $j$th class, and $\theta_j$ is the angle between $W_j$ and $x_i$.

Class-Specific Channel Attention

The main contribution of this paper is to develop a Few-Shot Learning (FSL) method using channel attention to solve the (category and domain) task distribution shift problem, and achieve new state-of-the-art results. In this section, we first introduce the problem setup, followed by the training of the feature backbone, the main architecture of our method, and the loss function design. Finally, we present the experimental evaluation and results. Figure 2 depicts a summary workflow of the proposed Class-Specific Channel Attention (CSCA) module FSL approach.

Problem Setup. In transfer-based few-shot image classification, a well-labeled base dataset $D^b$ with a large number of data is first used to train a feature backbone $F$ parameterized by $\theta$. The learned $F_{\theta}$ will be evaluated later with a series of few-shot classification tasks constructed with a novel dataset $D^{novel}$. Note that $D^b$ and $D^{novel}$ are completely disjoint, such that $D^b \cap D^{novel} = \emptyset$, and often time $D^b$ and $D^{novel}$ are largely distinct object categories or domains.

Each few-shot classification task is built up with a support set $S_\tau$ and a query set $Q_\tau$ to form an $N$-way $K$-shot classification task $\tau$, where both $S_\tau$ and $Q_\tau$ consist of $N$ different classes from $D^{novel}$ and $K$ denotes the number of samples of each class in $S_\tau$. Let $Q$ be the number of samples of each class in $Q_\tau$. After training with $D^b$, we can either (1)
Figure 3: A single CSCA weight vector. Each CSCA weight vector consists of a $1 \times c$ trainable weight vector $W$ to record the importance of each channel in each class. Then, each feature vector $E$ is weighted by $W$ to get $E'$.

Further fine-tune the pre-trained feature backbone $F_0$ with $S$, to better suit the evaluation task, or (2) use $F_0$ to extract features from samples of $S$, and $Q$, for subsequent classification steps. Following the above training steps, the model will next perform inference on $Q$. We use the average prediction accuracy among each randomly sampled $Q$, as the evaluation metric for the few-shot classification task.

For the case of transductive few-shot learning, the prediction is performed by considering all $N \times (K + Q)$ samples together. On the contrary, for the case of inductive few-shot learning, the prediction is performed independently on each of the $N \times Q$ samples. Here the prediction does not rely on other query samples, which aligns better with real-world usage scenarios. In this paper, we evaluate our method with the inductive and the typical 5-way, 1 or 5-shot, 15-query classification setting, i.e., $N = 5$, $K = 1$ or $Q = 5$.

**Feature Backbone.** We train the proposed feature backbone $F_0$ using a conventional few-shot setup on $D^b$ in each few-shot dataset. This is different from many other works in the literature (Hu et al. 2022; Bateni et al. 2022; Chen et al. 2021a; Rodriguez et al. 2020; Li et al. 2019; Huang et al. 2021), where external datasets are used to train the feature backbone. Using feature backbones trained with an external dataset is thought to be inconsistent with FSL, while there are many similar categories between the large dataset used in the above methods and the novel dataset in the Few-Shot Learning tasks. To provide a fair comparison with other methods without using external datasets, we consider WRN-Net (Zagoruyko and Komodakis 2016) to be the feature backbone and S2M2R (Mangla et al. 2020) to be the training method using the same setting as (Shalam and Korman 2022; Hu, Gripon, and Pateux 2021; Hu, Pateux, and Gripon 2022). After training with $D^b$, we discard the last classification layer and fix the weights $\theta$ of the feature backbone $F_0$ before the penultimate layers for the following steps. Note that a ReLU activation function is applied in the penultimate layers of $F_0$, thus, all the values in the output feature vectors are non-negative.

**Feature Transformation.** For an $N$-way $K$-shot classification task and a pre-trained backbone $F_0$ with output feature vector $E$ with $c$ channels, the $n$th class $C^n$ will be assigned a CSCA weight vector $W^n$ which includes $c$ trainable parameters.

For $K > 1$ few-shot classification tasks, the feature backbone $F_0$ first maps each input sample $X^n_{S,k}$ from the support set to a $c$-dimensional feature vector $E^n_{S,k}$. In the case of $K = 1$, we first conduct data augmentation to produce multiple representations, where detailed steps are provided in the experiment section. Next, a channel-wise transformation is used to multiply $E^n_{S,k}$ with $W^n$ to create $E^n_{S,k}'$ as follows:

$$E^n_{S,k}' = W^n(E^n_{S,k}).$$

Detailed operations of the CSCA weight vector are shown in Figure 3.

**Loss Functions.** We consider the loss function with the concept of grouping the feature vectors in the same class and separating them with the others, in replacement of conventional classification loss such as Cross-Entropy loss. As shown in Figure 4, in order to calculate the intra-class distance among samples in the $n$th class $C^n$, we need to construct prototype $P^n$ for $C^n$ by channel-wise averaging all the weighted feature vectors $E^n_{S,k}$ as follows:

$$P^n = \text{avg}(E^n_{S,k}).$$

Note that $P^n$ is in the same dimensional space as each weighted feature vector $E^n_{S,k}$. Then, the intra-class distance $\xi^n_{\text{intra}}$ for the $n$th class $C^n$ can be defined as:

$$\xi^n_{\text{intra}} = \frac{1}{K} \sum_{k=1}^{K} (1 - \cos(P^n, E^n_{S,k})),$$

while $\cos$ is the Cosine Similarity function. $\xi^n_{\text{intra}}$ is the average distance among the $c$-dimensional feature vectors $E^n_{S,1\sim K}$ and the prototype $P^n$ in the $n$th class. In addition, we can calculate the inter-class distance $\xi^n_{\text{inter}}$ by averaging the Cosine Distance between $P^n$ and other prototypes $P^m$ ($m \neq n$) as follows:

$$\xi^n_{\text{inter}} = \frac{1}{N-1} \sum_{m=1, m \neq n}^{N} (1 - \cos(P^n, P^m)).$$
For each dataset, we all entries to be 1, but increasing W.

Algorithm 1: Train the CSCA module

Input: Support set in high-dimensional space \( E_{S,k} \), \( W \) with all entries to be 1, \( n \in N \), \( k \in K \).

Output: \( W \) for \( C^n \), \( n \in N \).

1: for \( n \) in \( N \) do
2: while not converged do
3: for \( m \) in \( N \), \( k \) in \( K \) do
4: \( E_{S,k}^m = W^n( E_{S,k}^m ) \)
5: end for
6: for \( m \) in \( N \) do
7: calculate \( P^n \) (Eq.(4)).
8: end for
9: calculate \( \xi_{\text{intra}}^n, \xi_{\text{inter}}^n \) (Eq.(5),Eq.(6)).
10: optimize \( W \) with \( \mathcal{L}( \xi_{\text{intra}}^n, \xi_{\text{inter}}^n ) \) (Eq.(7)).
11: end while
12: end for
13: return \( W^{1-N} \)

The goal of this paper is to optimize \( W \) by reducing \( \xi_{\text{intra}}^n \) but increasing \( \xi_{\text{inter}}^n \) for \( C^n \) with the loss function \( \mathcal{L}^n \):

\[
\mathcal{L}^n( \xi_{\text{intra}}^n, \xi_{\text{inter}}^n ) = \log \frac{ \xi_{\text{intra}}^n + \epsilon }{ \xi_{\text{inter}}^n + M },
\]

As shown in Figure 5, the margin \( M \) is introduced to widen the distance among classes, and \( \epsilon \) is a small value to prevent taking log to zero. Algorithm 1 describes the detailed training process. Note that \( W^{1-N} \) are trained in turns.

Figure 5: The margin \( M \) is introduced to further maximize the decision margin in the high-dimensional feature space.

Evaluation Steps. After the training steps, we obtain \( W^{1-N} \) which contain the knowledge of the channel attention of the feature vectors in \( C^{1-N} \) extracted from \( F_\theta \), and the prototypes \( P^{1-N} \) for \( C^{1-N} \). To make a prediction on a query image \( X_{\text{Query,q}}^n \) belonging to \( C^n \), we first use the pre-trained feature backbone \( F_\theta \) to map \( X_{\text{Query,q}}^n \) to a \( n \)-dimensional feature vector \( P_{\text{Query,q}}^n \). The subsequent classification steps are similar to the operation of ProtoNet (Snell, Swersky, and Zemel 2017). Prediction is made by selecting the \( s \)th class which maximizes the Cosine Similarity \( \text{CosScore}(q, z) \) between \( P_{z}^{1-N} \) and the weighted feature vector \( W^2( E_{\text{Query,q}}^n ) \). The prediction is accurate if \( z = n \). Algorithm 2 describes the detailed evaluation steps.

Experimental Results

Datasets. We evaluate our proposed method on four standard benchmarks: miniImagenet (Vinyals et al. 2016), Tiered-ImageNet (Ren et al. 2018), CIFAR-FS (Bertinetto et al. 2018), and CUB-200-2011 (Wah et al. 2011). miniImagenet contains 100 randomly chosen classes from ILSVRC-12, which are then split into 64 training, 16 validation, and 20 testing classes. Each class contains 600 images of size \( 84 \times 84 \). Tiered-ImageNet is a larger subset of ILSVRC-12 with images selected from 608 classes. These classes are re-assigned to 20 categories for training, 6 categories for validation, and 8 categories for testing. The size of each image is also \( 84 \times 84 \). CIFAR-FS (or CIFAR-100 Few-Shots) divides the original CIFAR-100 set into 64 training, 16 validation, and 20 testing classes, respectively. There are 600 images per class, with 60,000 images in total, all in size \( 32 \times 32 \). CUB-200-2011 includes a total of 11,788 images of 200 bird species. The dataset is divided into 100 training, 50 validation, and 50 test classes, respectively. Each image is uniformly cropped to size \( 84 \times 84 \). All of our experiments are conducted with the listed datasets, and no external datasets are used.

Implementation Details. To provide fair comparisons with the state-of-the-art methods listed on the benchmarks, we perform our experiments using WRN-Net as the feature backbone trained following S2M2. For each dataset, we train the feature backbone with base classes and evaluate our methods on novel classes. For an \( N \)-way \( K \)-shot task, \( N \) classes are randomly sampled from the novel classes. Among these classes, \( K \) labeled support set and \( Q \) unlabeled query set are randomly sampled to form \( D_{\text{novel}} \). The support set is used for training the CSCA weight vectors, and the query set is used to evaluate the trained vectors. We conduct 10,000 randomly sampled runs for each task with \( N=5, K=1 \) or 5, \( Q=15 \). The final score is the mean accuracy among all 10,000 results.

For the \( N \)-way 1-shot few-shot classification training, we apply data augmentation first to increase the feature vectors, so that we could construct the prototypes for later steps. For all experiments, we consider RandomRotation, CenterCrop, Resize, RandomHorizontalFlip, RandomResizedCrop, and ColorJitter as augmentation functions.

All entries in the CSCA weight vectors are initialized to 1 to prevent the features from being damaged by the randomly initialized weights. The margin \( M \) is set to 0.5 and \( \epsilon \) is set to 0.0001 in Eq. (7). We train each CSCA weight vector for 10 to 20 epochs in the in-domain settings and 15 to 25 epochs in the cross-domain settings. We use Adam optimizer with
Table 1: Comparison of the accuracy on miniImagenet, Tiered-ImageNet, CIFAR-FS, and CUB-200-2011 with typical inductive and state-of-the-art transductive algorithms.

| Algorithm     | miniImageNet 1-shot | miniImageNet 5-shot | Tiered-ImageNet 1-shot | Tiered-ImageNet 5-shot | CIFAR-FS 1-shot | CIFAR-FS 5-shot | CUB-200-2011 1-shot | CUB-200-2011 5-shot |
|---------------|---------------------|---------------------|------------------------|------------------------|----------------|----------------|---------------------|---------------------|
| Baseline++    | 51.87               | 75.68               | -                      | -                      | -              | -              | 69.55               | 85.17               |
| MAML          | 49.61               | 65.72               | 53.31                  | 72.69                  | 55.50          | 72.00          | 72.99               | 86.64               |
| Matching Networks | 64.03             | 76.32               | -                      | -                      | -              | -              | 73.49               | 84.45               |
| S2M2 R        | 64.93               | 83.18               | 73.71                  | 88.59                  | 74.81          | 87.48          | 80.68               | 90.85               |
| AmdimNet      | 76.82               | 90.98               | -                      | -                      | -              | -              | 77.09               | 89.18               |
| P>M>F         | 95.3               | 98.4                | -                      | -                      | 84.3           | 92.2           | -                   | -                   |
| HCTransformers | 74.74              | 89.19               | 79.67                  | 91.72                  | 79.89          | 89.18          | -                   | -                   |

**INDUCTIVE**

**TRANSDUCTIVE**

| Backbone     | miniImageNet 1-shot | miniImageNet 5-shot | Tiered-ImageNet 1-shot | Tiered-ImageNet 5-shot | CIFAR-FS 1-shot | CIFAR-FS 5-shot | CUB-200-2011 1-shot | CUB-200-2011 5-shot |
|--------------|---------------------|---------------------|------------------------|------------------------|----------------|----------------|---------------------|---------------------|
| CNAPS+TETI   | 79.9                | 91.5                | 73.8                   | 87.7                   | -              | -              | -                   | -                   |
| PT+MAP       | 82.92               | 88.82               | 85.41                  | 90.44                  | 87.69          | 90.68          | 91.55               | 93.99               |
| s+f          | 84.81               | 90.62               | 84.29                  | 89.76                  | 87.16          | 88.38          | 90.56               | 93.5                |
| EASY 3xResNet12 | 85.42            | 93.5                | 86.67                  | 91.09                  | 89.44          | 91.86          | 94.78               | 96.43               |
| SOT          | 85.91               | 91.34               | -                      | -                      | 89.94          | 92.83          | 95.80               | 97.12               |

**OURS**

(+1.38) (+1.56) (+10.51) (+7.65) (+8.91) (+6.99) (+1.63) (+1.97)

Table 2: Accuracy of the proposed method in the cross-domain settings.

| Backbone | CUB-200-2011 1-shot | CUB-200-2011 5-shot |
|----------|---------------------|---------------------|
| CIFAR-FS | 99.91               | 99.97               |
| CUB-200-2011 | 95.85          | 96.51               |

| Dataset |
|---------|
| miniImageNet 1-shot | miniImageNet 5-shot | Tiered-ImageNet 1-shot | Tiered-ImageNet 5-shot | CIFAR-FS 1-shot | CIFAR-FS 5-shot | CUB-200-2011 1-shot | CUB-200-2011 5-shot |
|---------|
| mini    | 92.73               | 97.92               | 99.04                  | 99.43                   | -              | -              | -                   | -                   |
| CIFAR-FS | 95.85               | 96.51               | 84.79                  | 93.93                   | -              | -              | -                   | -                   |

Table 2: Accuracy of the proposed method in the cross-domain settings.

A learning rate of 0.1 per task. Due to the tiny weight size and that the model does not require large-scale training, we obtained an average execution time of 0.44 seconds per run on the Nvidia GeForce RTX 2080 GPU. This execution time includes the training of the CSCA weight vectors and the classification task execution.

**Comparison with State-of-the-Art (SoTA) methods.** We evaluate the proposed method on several benchmarks and compare the results with SoTA solutions. Table 1 summarizes the results. Our CSCA module outperforms all listed SoTA algorithms in all benchmarks with an inductive setting. Table 2 shows the evaluation results of our method in cross-domain settings. We observe stable results in the cross-domain evaluation. Some of the cross-domain accuracy values even outperform the in-domain evaluation values.

**Analysis.** To better understand how the CSCA module impacts the distribution of feature vectors, we visualize the changes in channels for the 5-way, 5-shot miniImageNet and the CIFAR-FS classification task in Figure 6. Oracle prototypes are constructed with the mean values in each channel of all feature vectors extracted from pre-trained WRN-Net in the novel data per class. These oracle prototypes can be viewed as the standard channel distribution for each class. The closely distributed dots in blue and red represent the oracle prototypes before and after being transformed by the CSCA module. Each dotted line in light blue and orange shows the mean value of the fifty furthest outlier feature vectors from the oracle prototypes per channel before and after being transformed by the CSCA module, respectively. Observe in these plots that the intra-class variance between the oracle prototypes and the transformed feature vectors in each class is decreased. Figure 7 visualizes the distribution of the feature vectors with t-SNE (Van der Maaten and Hinton 2008). Observe that the categories are better separated with larger inter-class variance and smaller intra-class variance. Better cluster homogeneity and heterogeneity are observed this way as well.

**Ablation Study.** To investigate the contribution of the proposed method, we conduct ablation experiments on standard miniImageNet, Tiered-ImageNet, CIFAR-FS, and CUB-200-2011 with typical inductive and state-of-the-art transductive algorithms. In ablation study (1), we conduct the classification with the raw feature vectors obtained from the pre-trained feature backbone \( F_0 \) without additional transformation (Figure 8a). In this case, this structure can be seen as ProtoNet with backbone trained with the procedure of S2M2 R in replacement of the meta-training method. From the results we speculate that the task distribution shift may be harmful to the quality of the feature vectors. Thus, it is indispensable to fine-tune the feature backbone or the extracted feature vectors with the target data. In (2), we compare the performance between the
Figure 6: **A closer look at the distribution of channels in two 5-way, 5-shot FSL tasks.** In each plot, a dot represents a channel with a value corresponding to the $y$-axis. The WRN-Net backbone outputs 640-channel feature vectors along the $x$-axis.

| Algorithm      | miniImagenet 1 shot | Tiered-ImageNet 1 shot | CIFAR-FS 1 shot | CUB-200-2011 1 shot |
|----------------|----------------------|------------------------|-----------------|---------------------|
| Ablation (1)   | 53.12 76.09          | 49.30 82.31            | 60.92 82.46     | 58.25 81.58         |
| Ablation (2)   | 58.93 83.17          | 55.68 86.47            | 64.47 87.57     | 68.6 91.57          |
| OURS           | 96.68 99.96          | 96.58 99.37            | 98.85 99.82     | 97.43 99.09         |

Table 3: **Ablation study results in terms of model accuracy.**

Figure 7: **Visualization of the improvement with the distribution of the feature vectors using t-SNE.** These 2 tasks are randomly sampled from miniImagenet with a 5-way, 5-shot training setting. Each dot represents one feature representation in the query set with $Q=15$.

global-class attention (Figure 8b) and local-class attention (Figure 8c). For the global-class attention, the feature vectors are transformed by the attention vector that is trained to learn the global-class information. The architecture of the attention vector in (Figure 8b) is the same as (Figure 3) but is used to learn global-class information. Even if the performance has improved with additional global-class attention, our method outperforms the above experiments.

Figure 8: **The architectures for a 5-way classification task in the ablation studies.** The classification is made regard with feature vectors (a) extracted by the feature backbone without any transforms, (b) transformed by a global-class attention vector, (c) transformed by local-class attention vectors in each class.

**Conclusion**

This paper discusses the channel bias problem caused by the shift in task distribution in few-shot classification tasks. We propose a channel-wise transformation called the **Class-Dependent Channel Attention** (CSCA) module to alleviate this problem. By learning the local-class features, the CSCA module highlights the important channels and downplays the opposite for each class, thus improves subsequent classification methods. We evaluate the CSCA module on 4 different standard benchmarks with 5-way 1/5-shot and also in cross-domain settings. All of the above experiments achieve outstanding results with a large margin of improvement compared to most state-of-the-art solutions. The CSCA module requires very few computational resources and does not require high-end GPU equipment to run with.

**Future Works.** In the future, we plan to further evaluate the CSCA module on granularity shift tasks and apply our method to other computer vision tasks including object detection in a few-shot setting.
References

Agarwal, M.; Yurochkin, M.; and Sun, Y. 2021. On sensitivity of meta-learning to support data. Advances in Neural Information Processing Systems, 34: 20447–20460.

Bateni, P.; Barber, J.; van de Meent, J.-W.; and Wood, F. 2022. Enhancing few-shot image classification with unlabelled examples. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2796–2805.

Bendou, Y.; Hu, Y.; Lafargue, R.; Lioi, G.; Pasdeloup, B.; Pateux, S.; and Gripon, V. 2022. EASY: Ensemble Augmented-Shot Y-shaped Learning: State-Of-The-Art Few-Shot Classification with Simple Ingredients. arXiv preprint arXiv:2201.09699.

Bertinetto, L.; Henriques, J. F.; Torr, P. H.; and Vedaldi, A. 2018. Meta-learning with differentiable closed-form solvers. arXiv preprint arXiv:1805.08136.

Bertinetto, L.; Henriques, J. F.; Valmadre, J.; Torr, P.; and Vedaldi, A. 2016. Learning feed-forward one-shot learners. Advances in neural information processing systems, 29.

Boudiaf, M.; Ziko, I.; Rony, J.; Dolz, J.; Piantanida, P.; and Ben Ayed, I. 2020. Information maximization for few-shot learning. Advances in Neural Information Processing Systems, 33: 2445–2457.

Bukchin, G.; Schwartz, E.; Saenko, K.; Shahar, O.; Feris, R.; Giryes, R.; and Karlinisky, L. 2021. Fine-grained angular contrastive learning with coarse labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8730–8740.

Cao, Y.; Xu, J.; Lin, S.; Wei, F.; and Hu, H. 2019. Genet: Non-local networks meet squeeze-excitation networks and beyond. In Proceedings of the IEEE/CVF international conference on computer vision workshops, 0–0.

Chen, D.; Chen, Y.; Li, Y.; Mao, F.; He, Y.; and Xue, H. 2021a. Self-supervised learning for few-shot image classification. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1745–1749. IEEE.

Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C. F.; and Huang, J.-B. 2019. A closer look at few-shot classification. arXiv:1904.04232.

Chen, X.; and Wang, G. 2021. Few-shot learning by integrating spatial and frequency representation. In 2021 18th Conference on Robots and Vision (CRV), 49–56. IEEE.

Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; and Feng, J. 2018. A 2-nets: Double attention networks. Advances in neural information processing systems, 31.

Chen, Y.; Liu, Z.; Xu, H.; Darrell, T.; and Wang, X. 2021b. Meta-baseline: Exploring simple meta-learning for few-shot learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 9062–9071.

Dhillon, G. S.; Chaudhari, P.; Ravichandran, A.; and Soatto, S. 2019. A baseline for few-shot image classification. arXiv preprint arXiv:1909.02729.

Doersch, C.; Gupta, A.; and Zisserman, A. 2020. Cross-transformers: spatially-aware few-shot transfer. Advances in Neural Information Processing Systems, 33: 21981–21993.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In International conference on machine learning, 1126–1135. PMLR.

Galassi, A.; Lippi, M.; and Torroni, P. 2020. Attention in natural language processing. IEEE Transactions on Neural Networks and Learning Systems, 32(10): 4291–4308.

Gao, Z.; Xie, J.; Wang, Q.; and Li, P. 2019. Global second-order pooling convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3024–3033.

Guo, M.-H.; Xu, T.-X.; Liu, J.-J.; Liu, Z.-N.; Jiang, P.-T.; Mu, T.-J.; Zhang, S.-H.; Martin, R. R.; Cheng, M.-M.; and Hu, S.-M. 2022. Attention mechanisms in computer vision: A survey. Computational Visual Media, 1–38.

Guo, Y.; Codella, N. C.; Karlinsky, L.; Codella, J. V.; Smith, J. R.; Saenko, K.; Rosing, T.; and Feris, R. 2020. A broader study of cross-domain few-shot learning. In European conference on computer vision, 124–141. Springer.

Hospedales, T.; Antoniou, A.; Miccaiili, P.; and Storkey, A. 2020. Meta-learning in neural networks: A survey. arXiv preprint arXiv:2004.05439.

Hu, J.; Shen, L.; Albanie, S.; Sun, G.; and Vedaldi, A. 2018. Gather-excite: Exploiting feature context in convolutional neural networks. Advances in neural information processing systems, 31.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 7132–7141.

Hu, S. X.; Li, D.; Stühmer, J.; Kim, M.; and Hospedales, T. M. 2022. 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, 9068–9077.

Hu, Y.; Gripon, V.; and Pateux, S. 2021. Leveraging the feature distribution in transfer-based few-shot learning. In International Conference on Artificial Neural Networks, 487–499. Springer.

Hu, Y.; Pateux, S.; and Gripon, V. 2022. Squeezing backbone feature distributions to the max for efficient few-shot learning. Algorithms, 15(5): 147.

Huang, K.; Geng, J.; Jiang, W.; Deng, X.; and Xu, Z. 2021. Pseudo-loss confidence metric for semi-supervised few-shot learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 8671–8680.

Kang, D.; Kwon, H.; Min, J.; and Cho, M. 2021. Relational embedding for few-shot classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 8822–8833.

Karmakar, P.; Teng, S. W.; and Lu, G. 2021. Thank you for attention: a survey on attention-based artificial neural networks for automatic speech recognition. arXiv preprint arXiv:2102.07259.
Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018b. Non-local neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 7794–7803.

Wang, Y.; Yao, Q.; Kwok, J. T.; and Ni, L. M. 2020b. Generalizing from a few examples: A survey on few-shot learning. ACM computing surveys, 53(3): 1–34.

Weiss, K.; Khoshgoftaar, T. M.; and Wang, D. 2016. A survey of transfer learning. Journal of Big data, 3(1): 1–40.

Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In European conference on computer vision, 499–515. Springer.

Woo, S.; Park, J.; Lee, J.-Y.; and Kweon, I. S. 2018. Cham: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV), 3–19.

Xing, E.; Jordan, M.; Russell, S. J.; and Ng, A. 2002. Distance metric learning with application to clustering with side-information. Advances in neural information processing systems, 15.

Yang, J.; Yang, H.; and Chen, L. 2021. Towards cross-granularity few-shot learning: coarse-to-fine pseudo-labeling with visual-semantic meta-embedding. In Proceedings of the 29th ACM International Conference on Multimedia, 3005–3014.

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

Zhang, H.; Cao, Z.; Yan, Z.; and Zhang, C. 2021. Sill-net: Feature augmentation with separated illumination representation. arXiv preprint arXiv:2102.03539.