Exploring Complementary Strengths of Invariant and Equivariant Representations for Few-Shot Learning

Mamshad Nayeem Rizve† Salman Khan† Fahad Shahbaz Khan† Mubarak Shah†
†Center for Research in Computer Vision, UCF, USA
‡Mohamed bin Zayed University of AI, UAE
nayeemrizve@knights.ucf.edu, {salman.khan, fahad.khan}@mbzuai.ac.ae, shah@crcv.ucf.edu

Abstract

In many real-world problems, collecting a large number of labeled samples is infeasible. Few-shot learning (FSL) is the dominant approach to address this issue, where the objective is to quickly adapt to novel categories in presence of a limited number of samples. FSL tasks have been predominantly solved by leveraging the ideas from gradient-based meta-learning and metric learning approaches. However, recent works have demonstrated the significance of powerful feature representations with a simple embedding network that can outperform existing sophisticated FSL algorithms. In this work, we build on this insight and propose a novel training mechanism that simultaneously enforces equivariance and invariance to a general set of geometric transformations. Equivariance or invariance has been employed standalone in the previous works; however, to the best of our knowledge, they have not been used jointly. Simultaneous optimization for both of these contrasting objectives allows the model to jointly learn features that are not only independent of the input transformation but also the features that encode the structure of geometric transformations. These complementary sets of features help generalize well to novel classes with only a few data samples. We achieve additional improvements by incorporating a novel self-supervised distillation objective. Our extensive experimentation shows that even without knowledge distillation our proposed method can outperform current state-of-the-art FSL methods on five popular benchmark datasets.

1. Introduction

In recent years, deep learning methods have made great strides on several challenging problems [29, 71, 28, 6, 7]. This success can be partially attributed to the availability of large-scale labeled datasets [14, 6, 81, 43]. However, acquiring large amounts of labeled data is infeasible in several real-world problems due to practical constraints such as the rarity of an event or the high cost of manual annotation. Few-shot learning (FSL) targets this problem by learning a model on a set of base classes and studies its adaptability to novel classes with only a few samples (typically 1-5) [19, 76, 65, 70]. Remarkably, this setting is different from transfer and self/semi-supervised learning that assumes the availability of pretrained models [63, 79, 36] or large-amounts of unlabeled data [17, 9, 3].

FSL has been predominantly solved using ideas from meta-learning. The two most dominant approaches are optimization-based meta-learning [19, 32, 61] and metric-learning based methods [65, 70, 1]. Both sets of approaches attempt to train a base learner which can be quickly adapted in the presence of a few novel class examples. However, recently it has been shown in [55] that the quick adaptation of the base learner crucially depends on feature reuse. Other recent works [72, 15, 10] have also shown that a baseline...
We enforce complimentary (see Fig. 1). Ance outperforms enforcing either of these objectives alone to demonstrate that enforcing both equivariance and invariance properties that help in learning the data structure of data can be modeled more efficiently. On the other hand, methods like [37] try to be robust against nuisance variations by learning transformation invariant features. However, such inductive biases do not provide optimal generalization on FSL tasks and the design of efficient inductive designs for FSL is relatively unexplored.

In this paper, we propose a novel feature learning approach by designing an effective set of inductive biases. We observe that the features required to achieve invariance against input transformations can provide better discrimination, but do not ensure optimal generalization. Similarly, features that focus on transformation discrimination are not optimal for class discrimination but learn equivariant properties that help in learning the data structure leading to better transferability. Therefore, we propose to combine the complementary strengths of both feature types through a multi-task objective that simultaneously seeks to retain both invariant and equivariant features. We argue that learning such generic features encourages the base feature extractor to be more general. We validate this claim by performing extensive experimentation on multiple benchmark datasets. We also conduct thorough ablation studies to demonstrate that enforcing both equivariance and invariance outperforms enforcing either of these objectives alone (see Fig. 1).

Our main contributions are:

- We enforce complimentary equivariance and invariance to a general set of geometric transformations to model the underlying structure of the data, while remaining discriminative, thereby improving generalization for FSL.
- Instead of extensive architectural changes, we propose a simple alternative by defining self-supervised tasks as auxiliary supervision. For equivariance, we introduce a transformation discrimination task, while an instance discrimination task is developed to learn transformation invariant features.
- We demonstrate additional gains with cross-task knowledge distillation that retains the variance properties.

2. Related Works

Few-shot Learning: The FSL approaches generally belong to the meta-learning family, which either learn a generalizable metric space [65, 55, 77, 50] or apply gradient-based updates to obtain a good initialization. In the first class of methods, Siamese networks related a pair of images [35], matching networks applied an LSTM based context encoder to match query and support set images [77], and prototypical networks used the distance between the query and the prototype embedding for class assignment [65]. A task-dependent metric scaling approach to improve FSL was introduced in [50]. The second category use gradient-based meta-learning methods that include using a sequence model (e.g., LSTM) to learn generalizable optimization rules [57], Model-agnostic Meta-Learning (MAML) to find a good initialization that can be quickly adapted to new tasks with minimal supervision [19], and Latent Embedding Optimization (LEO) that applied MAML in the low dimensional space from which high-dimensional parameters can be generated. A few recent efforts, e.g., ProtoMAML [75], combined the complementary strengths of metric-learning and gradient-based meta-learning methods.

Inductive Biases in CNNs: Inductive biases reflect our prior knowledge regarding a particular problem. State of the art CNNs are based on such design choices which range from the convolutional operator (e.g., the weight sharing and translational equivariance) [39], pooling operator (e.g., local neighbourhood relevance) [11], regularization mechanisms (e.g., sparsity with $\ell_1$ regularizer) [33], and loss functions (e.g., max-margin boundaries) [27]. Similarly, recurrent architectures and attention mechanisms are biased towards preserving contextual information and being invariant to time translation [2]. A number of approaches have been designed to achieve invariance to nuisances such as natural perturbations [30, 74], viewpoint changes [45], and image transformations [13, 5]. On the other hand, equivariant representations have also been investigated to retain knowledge regarding group actions [12, 53, 62, 41], thereby maintaining meaningful structure in the representations. In this work, we advocate that the representations required to simultaneously achieve invariance and equivariance can be useful for generalization to new tasks with limited data.

Self-supervised Learning for FSL: Our self-supervised loss is inspired by the recent progress in self-supervised learning (SSL), where proxy tasks are defined to learn transferable representations without adding any manual annota-
tions [56]. The pretext tasks include colorization [38, 80], inpainting [51], relative patch location [17, 49], and amount of rotation applied [24]. Recently, the potential of SSL for FSL was explored in [23, 67]. In [23] a parallel branch with the rotation prediction task to help learn generalizable features was added. Su et al. [67] also used rotation and permutation of patches as auxiliary tasks and concluded that SSL is more effective in low-shot regimes and under significant domain shifts. A recent approach employed SimCLR [9] style contrastive learning with augmented pairs to learn improved representations in either unsupervised pretraining [44] or episodic training [18] for FSL.

In contrast to the existing SSL approaches for FSL, we propose to jointly optimize for a complimentary pair of pretext tasks that lead to better generalization. Our novel distillation objective acquires knowledge from the classification as well as proxy task heads and demonstrates further performance improvements. We present our approach next.

3. Our Approach

We first describe the problem setting and the baseline training approach and then present our proposed approach.

3.1. Problem Formulation

Few-shot learning (FSL) operates in two phases, first a model on a set of base classes is trained and then at inference a new set of few-shot classes are received. We define the base training set as $D_b = \{(x, y)\}$, where $x \in \mathcal{I} \subset \mathbb{R}^{h \times w \times 3}$ is an image, and the one-hot encoded label $y \in \mathcal{Y} \subset \mathbb{R}^{N_b}$ can belong to a total of $N_b$ base classes. At inference, a data set of few-shot classes $D_f = \{(x, y)\}$ is presented for learning such that the label $y$ belongs to one of the $N_f$ novel classes, each with a total of $K$ examples ($K$ typically ranges between 1-5). The evaluation setting for few-shot classes is denoted as $N_f$-way, $K$-shot. Importantly, the $N_b$ base and $N_f$ few-shot classes belong to totally disjoint sets.

For solving the FSL task, most meta-learning methods [19, 65, 76] have leveraged an episodic training scheme. An episode consists of a small train and test set $(D_{tr}^i, D_{ts}^i)$. The examples for the train and test set of an episode are sampled from the same distribution i.e. from the same subset of meta-training classes. Meta-learning methods try to optimize the parameters of the base learner by solving a collection of these episodes. The main motivation is that the evaluation conditions should be emulated in the base training stage. However, following recent works [72, 15, 10], we do not use an episodic training scheme which allows us to train a single generalizable model that can be efficiently used for any-way, any-shot setting without retraining. Specifically, we train our base learner on the whole base training set $D_b$ in a supervised manner.

Let’s assume our base learner for the FSL task is a neural network, $f_{\Theta}$, parameterized with parameters $\Theta$. The role of this base learner is to extract good feature embeddings that can generalize for novel classes. The base learner $f_{\Theta}$ can project an input image $x$ into the embedding space $f_{\Theta} : x \rightarrow z$, such that $z \in \mathbb{R}^d$. Now, to optimize the parameters of the base learner $f_{\Theta}$ we need a classifier to project the extracted embeddings into the label space. To this end, we introduce a classifier function, $f_{\Phi}$, with parameters $\Phi$ that projects the embeddings $z$ into the label space $\mathcal{Y}$ i.e., $f_{\Phi} : z \rightarrow \tilde{y}$, such that $\tilde{y} \in \mathcal{Y}$.

We jointly optimize the parameters of both $f_{\Theta}$ and $f_{\Phi}$ by minimizing cross-entropy loss on the whole base-training set $D_b$. The classification loss is given by,

$$L_{ce} = -\log \frac{\exp(\tilde{y}_j \cdot z_i)}{\sum_i \exp(\tilde{y}_i)}, \ s.t., y \in \{0, 1\}^{N_b}, \tilde{y} = f_{\Theta, \Phi}(x).$$

To regularize the parameters of both of the sub-networks, we add a regularization term. Hence, the learning objective for our baseline training algorithm becomes:

$$L_{baseline} = \mathbb{E}_{(x,y) \sim D_b} [L_{ce}(f_{\Theta, \Phi}(x), y)] + \mathcal{R}(\Theta, \Phi). \quad (1)$$

Here, $\mathcal{R}(\Theta, \Phi)$ is an $L_2$ regularization term for the parameters $\Theta$ and $\Phi$. Next, we present our inductive objectives.

3.2. Injecting Inductive Biases through SSL

We propose to enforce equivariance and invariance to a general set of geometric transformations $\mathcal{T}$ by simply performing self-supervised learning (SSL). Self-supervision is particularly useful for learning general features without accessing semantic labels. For representation learning, self-supervised methods generally aim for either achieving equivariance to some input transformations or learn to discriminate instances by making the representations invariant. To the best of our knowledge, simultaneous equivariance and invariance to a general set of geometric transformations $\mathcal{T}$ has not been explored in the self-supervised literature. We are the first ones to do so.

The transformation set $\mathcal{T}$ can be obtained from a family of geometric transformations, $\mathcal{D}_T$; $\mathcal{T} \sim \mathcal{D}_T$. Here, $\mathcal{D}_T$ can be interpreted as a family of geometric transformations like Euclidean transformation, Similarity transformation, Affine transformation, and Projective transformation. All of these geometric transformations can be represented with a $\mathbb{R}^{3 \times 3}$ matrix with varying degrees of freedom. However, enforcing equivariance and invariance for a continuous space of geometric transformations, $\mathcal{T}$, is difficult and may even lead to suboptimal solutions. To overcome this issue, in this work, we quantize the complete space of affine transformations. We approximate $\mathcal{D}_T$ by dividing it into $M$ discrete set of transformations. Here, $M$ can be selected based on the nature of the data and computation budget.
For training, we generate $M$ transformed copies of an input image $x$ by applying all $M$ transformations. Then we combine all of these transformed images together into a single tensor, $x_{all} = \{x_0, x_1, \ldots, x_{M-1}\}$. Here, $x_i$ is the input image $x$ transformed through the $i^{th}$ transformation, $T_i$ (the subscript of $x_i$ is dropped in the subsequent discussion for clarity). We send this composite input to the network and optimize for both equivariance and invariance. The training is performed in a multi-task fashion. In addition to the classification head, which is needed for the baseline supervised training, two other heads are added on top of the base learner, as shown in Figure 2. One of these heads is used for enforcing equivariance, and the other is used for enforcing invariance. This multi-task training scheme ensures that the base learner retains both transformation equivariant and invariant features in the output embedding. We explain each component of our inductive loss below.

### 3.2.1 Enforcing Equivariance

As discussed above, equivariant features help us encode the inherent structure of data that improves generalization of features to new tasks. To enforce equivariance for the set $\mathcal{T}$ comprising of $M$ quantized transformations, we introduce an MLP $f_\theta$ with parameters $\Psi$. The role of $f_\theta$ is to project the output embeddings from the base learner $z$ into an equivariant space i.e., $f_\theta : z \rightarrow \tilde{u}$, where $\tilde{u} \in \mathcal{U} \subset \mathbb{R}^M$.

In order to train the network, we create proxy labels without any manual supervision. For a specific transformation, a $M$ dimensional one-hot encoded vector $u \in \{0, 1\}^M$ (such that $\sum_i u_i = 1$) is used to represent the label for $f_\theta$. Once proxy labels are assigned, training is performed in a supervised manner with the cross-entropy loss, as follows:

$$L_{eq} = -\log \frac{\exp(\tilde{u}^T u)}{\sum_j \exp(\tilde{u}_j^T u_j)} \text{ s.t. } \tilde{u} = f_\theta(x). \quad (2)$$

This supervised training with proxy labels in the equivariant space $\mathcal{U}$ ensures that the output embedding $z$ retains transformation equivariant features.

### 3.2.2 Enforcing Invariance

While equivariant representations are important to encode the structure in data, they may not be optimal for class discrimination. This is because the transformations we consider are nuisance variations that do not change the image class, therefore a good feature extractor should also encode representations that are independent of these input variations. To enforce invariance to the set $\mathcal{T}$ consisting of $M$ quantized transformations, we introduce another MLP $f_\Omega$ with parameters $\Omega$. The role of $f_\Omega$ is to project the output embeddings from the base learner $z$ into an invariant space i.e., $f_\Omega : z \rightarrow \tilde{v}$ where $\tilde{v} \in \mathcal{V} \subset \mathbb{R}^D$ and $D$ is the dimension of the invariant embedding.

To optimize for invariance we leverage a contrastive loss [26] for instance discrimination. We enforce invariance by maximizing the similarity between an embedding $v^m$ corresponding to a transformed image (after undergoing $m^{th}$ transformation $T_m$), and the reference embedding $v^0$ (embedding from the original image without applying any transformation $\mathcal{T}$). Importantly, we note that selecting negatives within a batch is not sufficient to obtain discriminant representations [78, 47]. We employ a memory bank in our contrastive loss to sample more negative samples without arbitrarily increasing the batch size. Further, the memory bank allows a stable convergence behavior [47]. Our learning objective is as follows:

$$L_{in} = -\frac{1}{M} \sum_{m=0}^{M-1} \log (h(v^m, \hat{v}^m)) \left\{ \begin{array}{ll} m \neq 0 \rightarrow \hat{v}^m = v^0 \\ m = 0 \rightarrow \hat{v}^m = \hat{v}^0 \end{array} \right. \quad (3)$$
Here, $m$ denotes the transformation index, $\Psi^0$ represents a previous copy of the reference $v^0$ held in the memory and the function $h(\cdot)$ is defined as,

$$h(v^\tau, v^m) = \frac{\exp\left(\frac{s(v^\tau, v^m)}{\tau}\right)}{\sum_{v' \in D_n} \exp\left(\frac{s(v^\tau, v^m)}{\tau}\right)}$$

Here, $s(\cdot)$ is a similarity function, $\tau$ is the temperature, and $D_n$ is the set of negative samples drawn from the memory bank for a particular minibatch. Note that we also maximize the similarity between the reference embedding $v^0$ and its past representation $\Psi^0$ which helps stabilize the learning.

### 3.2.3 Multi-head Distillation

Once the invariant and equivariant representations are learned by our model, we use self-distillation to train a new model using outputs from the previous model as anchor points (Fig. 2). Note that in typical knowledge distillation, information is exchanged from a larger model to a smaller one (student) by matching their softened outputs. In contrast, the outputs from the same models are matched in the self-distillation [21] where the smooth predictions encode inter-label dependencies, thereby helping the model to learn better representations.

In our case, a simple knowledge distillation by pairing the logits [72] would not ensure the transfer of invariant and equivariant representations learned by the previous model version. Therefore, we extend the idea of logit-based knowledge distillation and employ it to our invariant and equivariant embedding embeddings. Specifically, in parallel to minimizing the Kullback Leibler (KL) divergence for the soft output of supervised classifier head $f_\Phi$, we also minimize the KL divergence between the outputs of the equivariant head $f_\Psi$. Since the output of our invariant head $f_\Omega$ is not a probability distribution, we minimize a $L_2$ loss for distilling the knowledge at this head. The overall learning objective for knowledge distillation is as follows:

$$L_{kd} = KL(f^t_{\Theta, \Phi}(x), f_{\Theta, \psi}(x)) + KL(f^t_{\Theta, \psi}(x), f_{\Theta, \psi}(x))$$

Here, $f^t(\cdot)$ and $f^s(\cdot)$ are the teacher and student networks for distillation, respectively.

### 3.2.4 Overall Objective

Finally, we obtain the resultant loss for injecting the desired inductive biases by combining both equivariant $L_{eq}$, invariant $L_{in}$, and multi-head distillation $L_{kd}$ losses:

$$L_{inductive} = \mathbb{E}_{x \sim D_n, v' \sim D_n} \left[ L_{eq}(f_{\Theta, \psi}(x), u) + L_{in}(f_{\Theta, \Omega}(x), v') + L_{kd}(f^t_{\Theta, \Phi}(x), f^t_{\Theta, \psi}(x), f^t_{\Theta, \Omega}(x)) \right]$$

The overall loss is simply a combination of inductive and baseline objectives,

$$L = L_{base} + L_{inductive}. \quad (5)$$

### 3.3 Few-Shot Evaluation

For evaluation, we test our base learner $f_\Theta$ by sampling FSL tasks from a held-out test set comprising of images from novel classes. Each FSL task contains a support set and a corresponding query set $\{D_{supp}, D_{query}\}$; both contain images from the same subset of test classes. Using $f_\Theta$, we obtain embeddings for the images of both $D_{supp}$ and $D_{query}$. Following [72], we train a simple logistic regression classifier based on the image embeddings and the corresponding labels from the $D_{supp}$. We use that linear classifier to infer the labels of the query embeddings.

### 4. Experimental Evaluation

#### Datasets:

We evaluate our method on five popular benchmark FSL datasets. Two of these datasets are subset of the CIFAR100 dataset: CIFAR-FS [4] and FC100 [50]. Another two are derivatives of the ImageNet [14] dataset: minImageNet [76] and tieredImageNet [60]. The CIFAR-FS dataset is constructed by randomly splitting the 100 classes of the CIFAR-100 dataset into 64, 16, and 20 train, validation, and test splits. FC100 dataset makes the FSL task more challenging by making the splits more diverse; the FC100 train, validation, and test splits contain 60, 20, and 20 classes. Following [58], we use 64, 16, and 20 classes of the miniImageNet dataset for training, validation, and testing. The tieredImageNet dataset contains 608 ImageNet classes that are grouped into 34 high-level categories, and we use 20/351, 6/97, and 8/160 categories/classes for training, validation, and testing. We also evaluate our method on the newly proposed Meta-Dataset [75], which contains 10 diverse datasets to make the FSL task more challenging and closer to realistic classification problems.

#### Implementation Details:

Following [72, 46, 50, 40], we use a ResNet-12 network as our base learner to conduct experiments on CIFAR-FS, FC100, minImageNet, tieredImageNet datasets. Following [72, 40], we also apply Dropblock [22] regularizer to our Resnet-12 base learner. For Meta-Dataset experiments we use a Resnet-18 [29] network as our base learner to be consistent with [72]. We instantiate both of our equivariant and invariant embedding learners ($f_\Psi$, $f_\Theta$) with an MLP consisting of a single hidden layer. The classifier, $f_\Phi$, is instantiated with a single linear layer.

We use SGD optimizer with a momentum of 0.9 in all experiments. For CIFAR-FS, FC100, minImageNet, tieredImageNet datasets, we set the initial learning rate to 0.05 and use a weight decay of $5e-4$. For experiments on CIFAR-FS, FC100, minImageNet datasets, we train for 65 epochs; the learning rate is decayed by a factor of 10 after the first
Methods | Backbone | 1-Shot | 5-Shot |
---|---|---|---|
MAML [19] | 32-32-32-32 | 58.90 ± 1.90 | 71.50 ± 1.00 |
Proto-Net [65] | 64-64-64-64 | 55.50 ± 0.70 | 72.00 ± 0.60 |
Relation Net [70] | 64-96-128-256 | 55.00 ± 1.00 | 69.30 ± 0.80 |
R2D2 [5] | 96-192-384-512 | 65.30 ± 0.20 | 79.40 ± 0.10 |
Shot-Free [59] | ResNet-12 | 69.20 | 84.70 |
TEWAM [54] | ResNet-12 | 70.40 | 81.30 |
Proto-Net [65] | ResNet-12 | 72.20 ± 0.70 | 83.50 ± 0.50 |
MetaOptNet [40] | ResNet-12 | 72.60 ± 0.70 | 84.30 ± 0.50 |
Boosting [23] | WRN-28-10 | 73.60 ± 0.30 | 86.60 ± 0.20 |
Fine-tuning [15] | WRN-28-10 | 76.50 ± 0.68 | 85.79 ± 0.50 |
DSN-MR [64] | ResNet-12 | 75.60 ± 0.90 | 86.20 ± 0.60 |
MABAS [34] | ResNet-12 | 73.51 ± 0.92 | 85.49 ± 0.68 |
RFS-Simple [72] | ResNet-12 | 71.50 ± 0.80 | 86.00 ± 0.50 |
RFS-Distill [72] | ResNet-12 | 73.90 ± 0.80 | 86.90 ± 0.50 |
Ours | ResNet-12 | 76.83 ± 0.82 | 89.26 ± 0.58 |
Ours-Distill | ResNet-12 | 77.87 ± 0.85 | 89.74 ± 0.57 |

Table 1. Average 5-way few-shot classification accuracy with 95% confidence intervals on CIFAR-FS dataset; †trained on train and validation sets. Top two results are shown in red and blue.

Methods | Backbone | 1-Shot | 5-Shot |
---|---|---|---|
MAML [19] | 32-32-32-32 | 48.70 ± 1.84 | 63.11 ± 0.92 |
Matching Net [76] | 64-64-64-64 | 43.56 ± 0.84 | 55.31 ± 0.73 |
Proto-Net [65] | 64-64-64-64 | 49.42 ± 0.78 | 68.20 ± 0.66 |
Relation Net [70] | 64-96-128-256 | 50.44 ± 0.82 | 65.32 ± 0.70 |
R2D2 [4] | 96-192-384-512 | 51.20 ± 0.60 | 68.80 ± 0.10 |
SNALL [46] | ResNet-12 | 55.71 ± 0.99 | 68.88 ± 0.92 |
AdaResNet [48] | ResNet-12 | 56.88 ± 0.62 | 71.94 ± 0.57 |
TADAM [50] | ResNet-12 | 58.50 ± 0.30 | 76.70 ± 0.30 |
Shot-Free [39] | ResNet-12 | 59.04 | 77.64 |
TEWAM [54] | ResNet-12 | 60.07 | 75.90 |
MTL [68] | ResNet-12 | 61.20 ± 1.80 | 75.50 ± 0.80 |
MetaOptNet [40] | ResNet-12 | 62.64 ± 0.61 | 78.63 ± 0.46 |
Boosting [23] | WRN-28-10 | 63.77 ± 0.45 | 80.70 ± 0.33 |
Fine-tuning [15] | WRN-28-10 | 57.73 ± 0.62 | 78.17 ± 0.49 |
LEO-train [61] | WRN-28-10 | 61.76 ± 0.08 | 77.59 ± 0.12 |
Deep DTN [8] | ResNet-12 | 63.45 ± 0.86 | 79.91 ± 0.62 |
AFHN [42] | ResNet-18 | 62.38 ± 0.72 | 78.16 ± 0.56 |
AWGIM [25] | WRN-28-10 | 63.12 ± 0.08 | 78.40 ± 0.11 |
DSN-MR [64] | ResNet-12 | 64.60 ± 0.72 | 79.51 ± 0.50 |
MABAS [34] | ResNet-12 | 65.08 ± 0.86 | 82.70 ± 0.54 |
RFS-Simple [72] | ResNet-12 | 62.02 ± 0.63 | 79.64 ± 0.44 |
RFS-Distill [72] | ResNet-12 | 64.82 ± 0.60 | 82.14 ± 0.43 |
Ours | ResNet-12 | 66.82 ± 0.80 | 84.35 ± 0.51 |
Ours-Distill | ResNet-12 | 67.28 ± 0.80 | 84.78 ± 0.52 |

Table 2. Average 5-way few-shot classification accuracy with 95% confidence intervals on FC100 dataset; †trained on train and validation sets. Top two results are shown in red and blue.

60 epochs. We train for 60 epochs for experiments on the tieredImageNet dataset; the learning rate is decayed by a factor of 10 for 3 times after the first 30 epochs. For Meta-Dataset experiments, we set the initial learning rate to 0.1 and use a weight decay of 1e−4. We train our method for 90 epochs and decay the learning rate by a factor of 10 every 30 epochs. We use a batch size of 64 in all of our experiments except on Meta-Dataset where the batch size is set to 256 following [72]. For Meta-dataset experiments, we use standard data augmentation which includes random horizontal flip and random resized crop. For all the other dataset experiments we use random crop, color jittering and random horizontal flip for data augmentation following [72, 40]. Consistent with [72], we use a temperature coefficient of 4.0 for our knowledge distillation experiments. For all datasets, we perform one stage of distillation. We sample 600 FSL tasks to report our scores on all datasets except Meta-Dataset.

For our geometric transformations, we sample from a complete space of similarity transformations and use four rotation transformations: {0°, 90°, 180°, 270°}, two scaling transformations: {0.67, 1.0} and three aspect ration transformations: {0.67, 1.0, 1.33}. These geometric transformations can be applied in any combination. For all of our experiments, we set the total number of applied transformations to 16. Additional details and experiments with more geometric transformations are included in the supplementary materials. For the contrastive loss, we use a memory bank that stores 64-dimensional embedding of instances; we sample 6400 negative samples from the memory bank for each mini-batch and set the value of τ to 1.0.

4.1. Results

We present our results on five popular benchmark FSL datasets in Table 1-5 which demonstrates that even without multi-head distillation our proposed method consistently outperforms the current state-of-the-art (SOTA) FSL methods on both 5-way 1-shot and 5-way 5-shot tasks. By virtue of our novel representation learning approach which retains both the transformation invariant and equivariant features in the learned embeddings, our proposed method improves over the baseline RFS-Simple [72] method across all datasets by 2-5% for both 1-shot and 5-shot tasks. To be more specific, our method outperforms the current best results on CIFAR-FS dataset (Table 1) by 1.3% in the 1-
4.2. Ablations

To study the contribution of different components of our method, we do a thorough ablation study on three benchmark FSL datasets: miniImageNet, CIFAR-FS, and FC100 (Table 6). On these three datasets, our baseline supervised training achieves 62.02%, 71.50%, and 42.60% average accuracy on 5-way 1-shot task respectively; which is the same as RFS-Simple [72]. By enforcing invariance we obtain 2.62%, 2%, and 3.5% improvements respectively. Likewise, enforcing equivariance gives 4.07%, 4.87%, and 4.13% improvements over the baseline respectively. On the other hand, we get even bigger improvements by simultaneously optimizing for both equivariance and invariance; achieving 4.8%, 5.33%, and 4.78% improvements on top of the baseline supervised training. Besides, joint training gives 1.3%-3.3% improvement over only invariance training and 0.5%-0.7% improvement in comparison to only equivariance training. We observe similar trends for 5-way 5-shot task. This consistent improvement across all datasets for both tasks empirically validates our claim that joint optimization for both equivariance and invariance is beneficial for FSL tasks. Our ablation study also shows that the multi-head distillation improves the performance over the standard logit-level supervised distillation across all datasets.

Effect of the number of Transformations: To investigate the effect of the total number of applied transformations, we perform an ablation study on the CIFAR-FS validation set by varying the number of transformations, $M$. We present the results in Table 7, which demonstrates that initially, the performance of our method improves with the increasing $M$. However, the performance starts to saturate beyond a particular point. We hypothesize that the performance for an increasing number of transformations decreases since discriminating a higher number of transformations is more difficult and the model spends more representation capability for solving this harder task. A similar trend is observed in [24], where increasing the number of recognizable rotations does not lead to better performance. Based on Table 7 results, we set the value of $M$ to 16 for all of our experiments and do not tune the $M$ value from dataset to dataset.

4.3. Analysis

We do a t-SNE visualization of the output embeddings from $f_0$ for the test images of miniImageNet to demonstrate the effectiveness of our method (see Fig. 3). We observe that the base learner trained in a supervised manner can retain good class discrimination even for unseen test classes. However, as evident in Fig. 3, the class boundaries are not precise and compact. Enforcing invariance on top of the base learner leads to more compact class boundaries; however, the sample embeddings of different classes are still relatively closer to one another. On the other hand, enforcing equivariance leads to class representations that are well spread out since it retains the transformation equivariant information in the embedding space. Finally, our proposed method takes advantage of both of these complementary

<table>
<thead>
<tr>
<th>Dataset</th>
<th>1-Shot</th>
<th>5-Shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILSVRC</td>
<td>49.53</td>
<td>40.14</td>
</tr>
<tr>
<td>Omniglot</td>
<td>63.37</td>
<td>64.92</td>
</tr>
<tr>
<td>Aircraft</td>
<td>55.55</td>
<td>63.12</td>
</tr>
<tr>
<td>Birds</td>
<td>68.66</td>
<td>72.69</td>
</tr>
<tr>
<td>Textures</td>
<td>66.49</td>
<td>78.59</td>
</tr>
<tr>
<td>Quick Draw</td>
<td>51.52</td>
<td>62.48</td>
</tr>
<tr>
<td>Fungi</td>
<td>39.96</td>
<td>47.12</td>
</tr>
<tr>
<td>VGG Flower</td>
<td>87.15</td>
<td>91.60</td>
</tr>
<tr>
<td>Traffic Signs</td>
<td>48.83</td>
<td>77.51</td>
</tr>
<tr>
<td>MSCOCO</td>
<td>43.71</td>
<td>57.00</td>
</tr>
</tbody>
</table>

Table 4. Average 5-way few-shot classification accuracy with 95% confidence intervals on tieredImageNet dataset; *trained on train and validation sets. Top two results are shown in red and blue.

Table 5. Results on Meta-Dataset. Average accuracy (%) is reported with variable number of ways and shots, following the setup in [75]. 1000 tasks are sampled for evaluation. Top two results are shown in red and blue.

Method | BackBone | 1-Shot | 5-Shot |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML</td>
<td>32-32-32</td>
<td>51.67 ± 1.81</td>
<td>70.30 ± 1.75</td>
</tr>
<tr>
<td>Proto-Net [65]</td>
<td>64-64-64-64</td>
<td>53.31 ± 0.89</td>
<td>72.69 ± 0.74</td>
</tr>
<tr>
<td>Relation Net[70]</td>
<td>64-96-128-256</td>
<td>54.48 ± 0.93</td>
<td>71.32 ± 0.78</td>
</tr>
<tr>
<td>Shot-Free[59]</td>
<td>ResNet-12</td>
<td>63.52</td>
<td>82.59</td>
</tr>
<tr>
<td>MetaOptNet[40]</td>
<td>ResNet-12</td>
<td>65.99 ± 0.72</td>
<td>81.56 ± 0.53</td>
</tr>
<tr>
<td>Boosting[23]</td>
<td>WRN-28-10</td>
<td>70.53 ± 0.51</td>
<td>84.98 ± 0.36</td>
</tr>
<tr>
<td>Fine-tuning[15]</td>
<td>WRN-28-10</td>
<td>66.58 ± 0.70</td>
<td>85.55 ± 0.48</td>
</tr>
<tr>
<td>LEO-trainval*</td>
<td>WRN-28-10</td>
<td>66.33 ± 0.05</td>
<td>81.44 ± 0.09</td>
</tr>
<tr>
<td>AWGIM[25]</td>
<td>WRN-28-10</td>
<td>67.69 ± 0.11</td>
<td>82.82 ± 0.13</td>
</tr>
<tr>
<td>DSN-MR[64]</td>
<td>ResNet-12</td>
<td>67.39 ± 0.82</td>
<td>82.85 ± 0.56</td>
</tr>
<tr>
<td>RFS-Simple[72]</td>
<td>ResNet-12</td>
<td>69.74 ± 0.72</td>
<td>84.41 ± 0.55</td>
</tr>
<tr>
<td>RFS-Distill[72]</td>
<td>ResNet-12</td>
<td>71.52 ± 0.69</td>
<td>86.03 ± 0.49</td>
</tr>
<tr>
<td>Ours</td>
<td>ResNet-12</td>
<td>71.87 ± 0.89</td>
<td>86.82 ± 0.58</td>
</tr>
<tr>
<td>Ours-Distill</td>
<td>ResNet-12</td>
<td>72.21 ± 0.90</td>
<td>87.08 ± 0.58</td>
</tr>
</tbody>
</table>

Table 6. Average accuracy (%) is reported with variable number of ways and shots, following the setup in [75]. 1000 tasks are sampled for evaluation. Top two results are shown in red and blue.
information outperforms all of these SSL tasks by a good margin. We found that our proposed method which aims to learn representations that retain both transformation invariant and equivariant information outperforms all of these SSL tasks by a good margin. Besides, we have noticed that the patch-based SSL tasks generally underperform in comparison to SSL tasks that rely on changing the global statistics of the image while maintaining the local statistics; this conclusion is in line with the experimental results from [23].

5. Conclusion

In this work, we explored a set of inductive biases that help us learn highly discriminative and transferable representations for FSL. Specifically, we showed that simultaneously learning equivariant and invariant representations to a set of generic transformations results in retaining a complimentary set of features that work well for novel classes. We also designed a novel multi-head knowledge distillation objective which delivers additional gains. We conducted extensive ablation to empirically validate our claim that joint optimization for invariance and equivariance leads to more generic and transferable features. We obtained new state-of-the-art results on four popular benchmark FSL datasets as well as on the newly proposed challenging Meta-Dataset.

**Acknowledgements** This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA RD Contract No. D17PC00345. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.
References


Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2017. 2


Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 1


