Self-supervised Knowledge Distillation for Few-shot Learning

Jathushan Rajasegaran[1]
brjathu@gmail.com
Salman Khan[2,3]
salman.khan@mbzuai.ac.ae
Munawar Hayat[4]
munawar.hayat@monash.edu
Fahad Shahbaz Khan[2,5]
fahad.khan@mbzuai.ac.ae
Mubarak Shah[6]
mshah@ucf.edu

[1] University of California, Berkeley, USA
[2] Mohamed Bin Zayed University of AI, UAE
[3] Australian National University, AU
[4] Monash Univeristy, AU
[5] CVL, Linköping University, Sweden
[6] University of Central Florida, USA

Abstract

Real-world contains an overwhelmingly large number of object classes, learning all of which at once is infeasible. Few-shot learning provides a promising learning paradigm due to its ability to quickly adapt to novel distributions with only a few samples. Recent works [8, 39] show that simply learning a good feature embedding can outperform more sophisticated meta-learning and metric learning algorithms for few-shot learning. This paper proposes a self supervised knowledge distillation approach, which learns a strong equivariant feature embedding for few-shot learning, by faithfully encoding inter-class relationships and preserving intra-class diversity. To this end, we follow a two-stage learning process: *first*, we train our model using a self-supervised auxiliary loss to maximize the entropy of the feature embedding, thus creating an optimal output manifold. In the *second* stage, we minimize the entropy on feature embedding by bringing self-supervised positive twins together, while constraining the learned manifold with student-teacher distillation. Our experiments show that, even in the first stage, features learnt by self-supervision can outperform current state-of-the-art methods, with further gains achieved by our second stage distillation process. Our codes are publicly available at: https://github.com/brjathu/SKD

1 Introduction

Modern deep learning algorithms generally require a large amount of annotated data which is expensive to acquire [1, 20]. Inspired by the fact that humans can learn from only a few examples, few-shot learning (FSL) offers a promising machine learning paradigm. FSL aims to develop models that can generalize to new concepts using only a few annotated samples (typically in 1-5 range). Data scarcity and limited supervision makes FSL a challenging task.

Existing works mainly approach FSL using meta-learning [2, 12, 18, 22, 23, 33, 35] to adapt the base learner for the new tasks, or by enforcing margin maximizing constraints

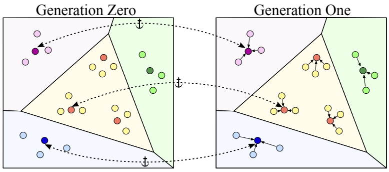

Figure 1: *Self-supervised Knowledge Distillation* (SKD) operates in two phases. In Gen-0, self-supervision is used to estimate the true prediction manifold, equivariant to input transformations. Specifically, we enforce the model to predict the amount of input image rotation using only the output logits (pretext task). In Gen-1, we force the original sample outputs to be the same as in Gen-0 (*dotted lines*), while reducing the distance with their augmented versions to enhance discriminability.

through metric learning [21, 36, 38, 41]. In doing so, these FSL methods ignore the importance of intra-class diversity while seeking to achieve inter-class discriminability. In this work, instead of learning representations which are invariant to within classes, we argue for an equivariant representation. Our main intuition is that major transformations in the input domain are desired to be reflected in their corresponding outputs to ensure output space diversity. By faithfully reflecting these changes in an equivariant manner, we seek to learn the true natural manifold of an object class samples.

We propose a two-stage self-supervised knowledge distillation (SKD) approach for FSL. Despite the availability of only few-shot labeled examples, we show that auxiliary self-supervised learning (SSL) signals can be mined from the limited data, and effectively leveraged to learn the true output-space manifold of each class. For this purpose, we take a direction in contrast to previous works which learn an invariant representation that maps augmented inputs to the same prediction [5]. With the goal to enhance generalizability of the learnt features, we first learn a *Generation-zero* (Gen-0) model whose output predictions are equivariant to the input transformations, thereby avoiding overfitting and ensuring heterogeneity in the prediction space.

Once the *Generation-zero* model has learned to estimate the optimal output manifold, we perform knowledge distillation by treating the learned model as a teacher network and training a student model with the teacher's outputs. Different from the first stage, we now enforce that the augmented samples and their original inputs result in similar predictions to enhance between-class discrimination. The knowledge distillation mechanism therefore guides the *Generation-one* (Gen-1) model to develop the following intuitive properties. *First,* the output class manifold is diverse enough to preserve major transformations in the input, thereby avoiding overfitting and improving generalization. *Second,* the learned relationships in the output space encode natural connections among classes e.g., two similar classes should have correlated predictions as opposed to totally independent and orthogonal projections considered in one-hot encoded ground-truths. Thus, by faithfully representing the output space via encoding inter-class relationships and preserving intra-class diversity, our approach learns improved representations for FSL.

The following are the main contributions of this work:

- Different to existing approaches that use SSL as an auxiliary task, we show the benefit of SSL towards enforcing diversity constraints in the output prediction space. Our simple design sequentially applies self-supervision after the final classification layer.
- A dual-stage training regime which first estimates the diverse output manifold by learning equivariant features, and then minimizes the positive-augmented pair distance while anchoring the original samples to preserve the learned manifold using distillation.

- Extensive evaluations on five datasets with consistent gains over recent FSL methods.

2 Related work

Few-shot learning (FSL): There have been several efforts on FSL ranging from metric learning to meta-learning methods. Metric learning methods commonly learn a metric space, in which the support set can be easily matched with the query set. For example, Koch *et al.* [21] use a Siamese network to learn a similarity metric to classify unknown classes, with the aid of a support set. Sung *et al.* [38] use a relation module to learn the relationships between support set and the query image. Matching networks [41] employ attention and memory to learn a network that matches support set to the query image. In addition, [36] assigns the mean embedding as a prototype and minimizes the distance from it with rest of the samples in the query set. In contrast, we only use augmented pairs of an image to move their embeddings closer, while preserving their respective distances in the output space.
Self-supervised learning (SSL): SSL defines a pretext learning task that can enhance model's learning capability without requiring any additional annotation effort [19]. Generally, these surrogate tasks require a higher-level understanding, thereby forcing the learning agent to learn useful representations while solving the auxiliary tasks. The existing SSL techniques mostly differ in the way supervisory signal is obtained from the data. For example, [15] defines pretext supervisory signal in terms of the amount of rotation applied to an input image. Doersch *et al.* [9] train a CNN to predict the relative position of a pair of randomly sampled image patches. This idea is further extended to predict permutations of multiple image patches in [26]. Alternatively, image colorization and object counting were employed as pretext tasks to improve representation learning [27, 45]. Zhai *et al.* [44] propose an SSL approach in a semi-supervised setting where some labelled and many unlabelled examples were available. Different from these works, our approach uses self-supervision to enforce additional constraints in the classification space. Close to our work is a set of approaches that seek to learn representations that are invariant to image transformations and augmentations [3, 5, 10]. In contrast, our approach does the *exact opposite*: we seek to learn an equivariant representation, so that the true manifold of a class can be learned with only a few-examples.
Embeddings for FSL: Recent works have highlighted the significance of learning strong embeddings for FSL. For example, [8, 39] show that a simple baseline can achieve very competitive performance by learning a powerful feature embedding. Similarly, [31] attribute the success of meta-learning for FSL [12, 13, 23, 35] to its strong feature representation capability rather than meta-learning itself. Our work is an effort along the same direction, and proposes a novel self-supervised knowledge distillation approach that can learn effective feature representations for FSL with limited supervision.
Differences with close works: The closest to our work is Gidaris *et al.* [16], which uses self-supervision to boost FSL. However, [16] simply employs self-supervision as an auxiliary loss, while we use it to shape and constrain the learning manifold . Architecture wise, we use a sequential self-supervision layer, while [16] has a parallel design. Furthermore, [16] does not explore multiple generations. We propose a dual-stage learning process, where the second generation further improves the learned representations by constraining the embedding space using distillation and bringing embeddings of and original and augmented image pairs closer. Unlike existing FSL methods (e.g., MAML [12], ProtoNets [36]) which require setting-specific models (e.g., a 5-way 1-shot model cannot be used for 10-way 1-shot, and requires re-training from scratch), we train a single generic model, which is applicable to any

number of ways or shots. Further, while recent works require extra steps to remember the base classes, our method can directly predict base classes with high accuracy (see Sec. 4.2).

There is a line work which uses distillation to improve the model quality, [30] proposed using various augmentations to create soft/pseudo labels for the unlabeled classes, subsequently used to re-train a new model. However, they do not exploit the self-supervised learning based on the augmentations. Their model is only trained with soft labels, no self-supervision is used in their training. Additionally, our two-stage training enforces different FSL objectives at different stages, not just knowledge distillation. First we force the model to explore the embeddings space and learn rich features and then we engorge the model to be more discriminative. These objectives are necessary for the model to perform well on FSL, with unseen classes. Our learning objective is similar to the learning objective of [42], however our motivation behind these objectives are fundamentally different from theirs. We only use self-supervision during Gen-0 to encourage the model to explore the feature space and create more diverse representations. This is essential because our evaluation is on unseen classes (while [42] is evaluating on seen classes). Further, unlike [B] we do not train our Gen-1 model (equivalent to [42] student) on self-supervision loss. Because, using the second stage we focus on making the model to be more discriminative between **unseen classes**, therefore no cross-entropy loss on labeled data is used during Gen-1. Our Gen-1 training is only about grouping unseen images, irrespective of the class labels. While [42] uses both label loss and self-supervision to train the student.

3 Self-supervised Knowledge Distillation

The proposed SKD uses a two stage training pipeline: *Generation-zero* (Gen-0) and *Generation-one* (Gen-1). Gen-0 utilizes self-supervision to learn a diverse classification manifold, in which the learned embeddings are equivariant to rotation (or another transformation). Later for Gen-1, we employ the Gen-0 model as a teacher and use original (non-augmented) images as anchors to preserve the learned manifold, while rotated version of the images are used to reduce intra-class distances in the embedding space to learn discriminative features.

3.1 Problem Formulation

Let's assume a neural network F contains feature embedding parameters Θ, and classification weights Φ. Any input image $\boldsymbol{x}$ can be mapped to a feature vector $\boldsymbol{v} \in \mathbb{R}^d$ by a function f_Θ: $\boldsymbol{x} \rightarrow \boldsymbol{v}$. Consequently, features $\boldsymbol{v}$ are mapped to logits $\boldsymbol{p} \in \mathbb{R}^c$ by another function f_Φ: $\boldsymbol{v} \rightarrow \boldsymbol{p}$, where c denotes the number of output classes. Hence, conventionally F is defined as a composition of these functions, $F{=}f_\Phi \circ f_\Theta$. In this work, we introduce another function f_Ψ, parameterized by Ψ, such that, f_Ψ: $\boldsymbol{p} \rightarrow \boldsymbol{q}$, which maps logits $\boldsymbol{p}$ to a secondary set of logits $\boldsymbol{q} \in \mathbb{R}^s$ for self-supervised task (e.g., rotation classification). For each input $\boldsymbol{x}$, we automatically obtain labels $r \in \{1, \dots, s\}$, where s is the number of self-supervised labels, e.g., $s = 4$ when predicting 4 rotation angles. Therefore, the complete network can be represented as $F_{\Theta,\Phi,\Psi} = f_\Psi \circ f_\Phi \circ f_\Theta$.

We consider a dataset $\mathcal{D}$ with m image-label pairs $\{\boldsymbol{x}_i, y_i\}_m$ where $y_i \in \{1, \dots, c\}$. During evaluation, we sample episodes as in classical FSL setting. An episode $\mathcal{D}_{eval}$ contains, $\mathcal{D}_{supp}$ and $\mathcal{D}_{query}$. In an n-way k-shot setting, $\mathcal{D}_{supp}$ has k samples for each of n classes.

3.2 Generation-Zero: Learning Data Manifold

During our first stage (called Gen-0), a minibatch $\mathcal{B}{=}\{\mathbf{x},\mathbf{y}\}$ is randomly sampled from the dataset $\mathcal{D}$, which has m number of image-label pairs such that $\mathbf{x} = \{\boldsymbol{x}_i\}_m, \mathbf{y} = \{y_i\}_m$. We first

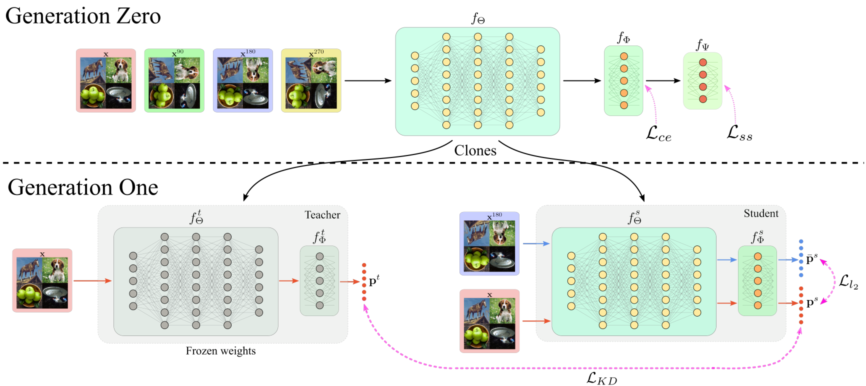

Figure 2: *Overall training process of SKD:* Gen-0 uses multiple rotated versions of the images to train the neural network to predict the class as well as the rotated angle. Then during Gen-1, we use original version of the images as anchor points to preserve the manifold while moving the logits for the rotated version closer, to increase the discriminative ability of the network.

take the images $\mathbf{x}$ and apply a transformation function $\mathcal{T}(\cdot)$ to create augmented copies of $\mathbf{x}$. For the sake of brevity, here we consider $\mathcal{T}(\cdot)$ as a rotation transformation, however, any other suitable transformation can also be considered as we show in our experiments (Sec. 4.2). Applying rotations of $90, 180$ and 270 degrees to $\mathbf{x}$, we create $\mathbf{x}^{90}$, $\mathbf{x}^{180}$ and $\mathbf{x}^{270}$, respectively. Then we combine all augmented versions of images into a single tensor $\widehat{\mathbf{x}} = \{\mathbf{x}, \mathbf{x}^{90}, \mathbf{x}^{180}, \mathbf{x}^{270}\}$ whose corresponding class labels are $\widehat{\mathbf{y}} \in \mathbb{R}^{4\times m}$. Additionally, one-hot encoded labels $\widehat{\mathbf{r}} = \{\boldsymbol{r}_i \in \mathbb{R}^s\}_{4\times m}$ for the rotation direction are also created, where $s = 4$ due to the four rotation angles in our self-supervised task.

First, we pass $\widehat{\mathbf{x}}$ through f_Θ, resulting in the features $\widehat{\mathbf{v}} \in \mathbb{R}^{d\times(4\times m)}$. Then, the features are passed through f_Φ to get the corresponding logits $\widehat{\mathbf{p}} \in \mathbb{R}^{c\times(4\times m)}$, and finally, the logits are passed through f_Ψ, to get the rotation logits $\widehat{\mathbf{q}} \in \mathbb{R}^{s\times(4\times m)}$,

$$f_\Theta(\widehat{\mathbf{x}}) = \widehat{\mathbf{v}}, \qquad f_\Phi(\widehat{\mathbf{v}}) = \widehat{\mathbf{p}}, \qquad f_\Psi(\widehat{\mathbf{p}}) = \widehat{\mathbf{q}}.$$

We employ, two loss functions to optimize the model in Gen-0: (a) categorical cross entropy loss $\mathcal{L}_{ce}$ between the predicted logits $\widehat{\mathbf{p}}$ and the true labels $\widehat{\mathbf{y}}$, and (b) a self-supervision loss $\mathcal{L}_{ss}$ between the rotation logits $\widehat{\mathbf{q}}$ and rotation labels $\widehat{\mathbf{r}}$. Note that, in this paper all our self-supervision tasks are simply the prediction of what data augmentation is used [15]. Therefore, our self-supervision loss is simply a cross entropy loss, using n-way classification of the data augmentations used in the model. However other kind of self-supervison can be also used without any modifications. These two loss terms are combined with a weighting coefficient α (tuned on a validation set) to get our final loss,

$$\mathcal{L}_{\text{Gen-0}} = \mathcal{L}_{ce} + \alpha \cdot \mathcal{L}_{ss}, \ \text{s.t.,} \ \mathcal{L}_{ce}(\boldsymbol{p}, y) = -\log\left(\frac{\exp(p_y)}{\sum_j \exp(p_j)}\right), \ \mathcal{L}_{ss}(\boldsymbol{q}, r) = -\log\left(\frac{\exp(q_r)}{\sum_j \exp(q_j)}\right).$$

The training process for Gen-0 model can be stated as the following optimization problem,

$$\min_{\Theta,\,\Phi,\,\Psi} \ \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{D}}\big[\mathcal{L}_{ce}(f_{\Phi,\Theta}(\widehat{\mathbf{x}}), \widehat{\mathbf{y}}) + \alpha \cdot \mathcal{L}_{ss}(f_{\Phi,\Theta,\Psi}(\widehat{\mathbf{x}}), \widehat{\mathbf{r}})\big]. \tag{1}$$

The above objective ensures that the output logits are representative enough to encapsulate information about the input transformation, thereby successfully predicting the amount of rotation applied to the input. This behaviour allows us to maintain diversity in the output space while faithfully estimating the natural data manifold of each object category.

3.3 Generation-One: Knowledge Distillation

Once the Gen-0 model is trained with cross entropy and self-supervision loss functions, we take two clones of the trained model: a teacher model F^t and a student model F^s. The weights of the teacher model are frozen and used only to guide students learning. Again, we sample a minibatch $\mathcal{B}$ from $\mathcal{D}$ and generate a twin $\bar{\mathbf{x}} \in \widehat{\mathbf{x}} \backslash \mathbf{x}$ from $\mathbf{x}$. In this case, a twin $\bar{\mathbf{x}}$ is simply a rotated version of $\mathbf{x}$ (e.g., $\mathbf{x}^{180}$). During Gen-1 training, $\mathbf{x}$ is used as an anchor point to constrain any changes to the classification manifold. This is enforced by a knowledge distillation [17] loss between teacher and student networks. Concurrently, an auxiliary ℓ_2 loss is employed to bring the embeddings of $\mathbf{x}$ and $\bar{\mathbf{x}}$ together to enhance feature discriminability while preserving the original output manifold. Note that, by using positive augmented pairs we only change embedding space within the class manifold.

Specifically, we first pass $\mathbf{x}$ through the teacher network $F^t = f^t_{\Phi,\Theta} \circ f^t_{\Psi}$ and its logits $\mathbf{p}^t$ are obtained. Then, $\mathbf{x}$, $\bar{\mathbf{x}}$ are passed through the F^s to get their corresponding logits $\mathbf{p}^s$, and $\bar{\mathbf{p}}^s$,

$$f^t_{\Phi,\Theta}(\mathbf{x}) = \mathbf{p}^t, f^s_{\Phi,\Theta}(\{\mathbf{x},\bar{\mathbf{x}}\}) = \{\mathbf{p}^s,\bar{\mathbf{p}}^s\} \text{ s.t., } f_{\Phi,\Theta}{=}f_{\Phi}{\circ}f_{\Theta}.$$

We use Kullback–Leibler (KL) divergence measure between $\mathbf{p}^t = \{\boldsymbol{p}^t_i\}$ and $\mathbf{p}^s = \{\boldsymbol{p}^s_i\}$ for knowledge distillation, and apply ℓ_2 loss between $\mathbf{p}^s$ and $\bar{\mathbf{p}}^s$ to achieve better discriminability,

$$\mathcal{L}_{\text{KD}}(\boldsymbol{p}^s,\boldsymbol{p}^t,T) = \text{KL}\left(\sigma(\frac{\boldsymbol{p}^s}{T}),\sigma(\frac{\boldsymbol{p}^t}{T})\right), \qquad \mathcal{L}_{\ell_2}(\boldsymbol{p}^s,\bar{\boldsymbol{p}}^s) = \|\boldsymbol{p}^s - \bar{\boldsymbol{p}}^s\|_2,$$

where, σ is a softmax function and T is a temperature parameter used to soften the output distribution. Finally, we combine these two loss terms by a coefficient β as follows,

$$\mathcal{L}_{\text{Gen-1}} = \mathcal{L}_{\text{KD}} + \beta \cdot \mathcal{L}_{\ell_2}. \tag{2}$$

The overall Gen-1 training process can be stated as the following optimization problem,

$$\min_{\Theta,\,\Phi} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{D}}\left[\mathcal{L}_{\text{KD}}(f^s_{\Phi,\Theta}(\mathbf{x}),f^t_{\Phi,\Theta}(\mathbf{x})) + \beta \cdot \mathcal{L}_{\ell_2}(f^s_{\Phi,\Theta}(\mathbf{x}),f^s_{\Phi,\Theta}(\bar{\mathbf{x}}))\right].$$

Note that, for our model, it is necessary to have the self-supervision branch (rotation classification head) sequentially added to the classification layer, which is unlike previous works [4, 16, 37] that connect rotation classification head directly after the feature embedding layer. This is because, during the Gen-0, we encourage the penultimate layer to encode information about both the image class and its rotation (thus preserving output space diversity). Later in Gen-1, we bring the logits of the rotated pairs closer (to improve discrimination). These objectives are not achievable if the rotation head is connected directly to the feature embedding layer, or if distillation is performed on the features as in previous works.

3.4 SKD at Inference

During evaluation, a held out part of the dataset is used to sample tasks. This comprises of a support set and a query set $\{\mathcal{D}_{supp}, \mathcal{D}_{query}\}$. $\mathcal{D}_{supp}$ has image-label pairs $\{\mathbf{x}_{supp}, \mathbf{y}_{supp}\}$, while $\mathcal{D}_{query}$ is an image tensor $\mathbf{x}_{query}$. Both $\mathbf{x}_{supp}$ and $\mathbf{x}_{query}$ are fed to the final trained f^s_{Θ}

model to get the feature embeddings $\mathbf{v}_{supp}$ and $\mathbf{v}_{query}$, respectively. We use a simple logistic regression classifier [2, 39] to map the labels from support set to query set. The embeddings are ℓ_2 normalized onto a unit sphere [39]. We randomly sample 600 tasks, and report mean classification accuracy with 95% confidence interval. Note that unlike popular meta-learning algorithms (e.g., [12, 23]), a major strength of the proposed method is that it does not need to train multiple models for different values of n and k in n-way, k-shot classification. Since, the classification is disentangled from feature learning in our case, the same model can be used to evaluate for any value of n and k in FSL.

4 Experiments and Results

We comprehensively compare our method on five benchmark few-shot learning datasets that include miniImageNet [41], tieredImageNet [34], CIFAR-FS [2], FC100 [28] and Meta-dataset [40]. Additionally, we provide an extensive ablation study to investigate the individual contributions of different components in our framework (Sec. 4.2).

Method	Backbone	miniImageNet, 5-way		tieredImageNet, 5-way	
		1-shot	5-shot	1-shot	5-shot
MAML [12]	32-32-32-32	48.70 ± 1.84	63.11 ± 0.92	51.67 ± 1.81	70.30 ± 1.75
Prototypical Networks† [36]	64-64-64-64	49.42 ± 0.78	68.20 ± 0.66	53.31 ± 0.89	72.69 ± 0.74
Dynamic Few-shot [14]	64-64-128-128	56.20 ± 0.86	73.00 ± 0.64	-	-
Relation Networks [38]	64-96-128-256	50.44 ± 0.82	65.32 ± 0.70	54.48 ± 0.93	71.32 ± 0.78
R2D2 [2]	96-192-384-512	51.2 ± 0.6	68.8 ± 0.1	-	-
SNAIL [25]	ResNet-12	55.71 ± 0.99	68.88 ± 0.92	-	-
TADAM [28]	ResNet-12	58.50 ± 0.30	76.70 ± 0.30	-	-
MetaOptNet [22]	ResNet-12	62.64 ± 0.61	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53
Diversity w/ Cooperation [11]	ResNet-18	59.48 ± 0.65	75.62 ± 0.48	-	-
Boosting [16]	WRN-28-10	63.77 ± 0.45	80.70 ± 0.33	70.53 ± 0.51	84.98 ± 0.36
Fine-tuning [8]	WRN-28-10	57.73 ± 0.62	78.17 ± 0.49	66.58 ± 0.70	85.55 ± 0.48
LEO-trainval† [35]	WRN-28-10	61.76 ± 0.08	77.59 ± 0.12	66.33 ± 0.05	81.44 ± 0.09
FEAT [43]	ResNet-12	66.78 ± n/a	82.05 ± n/a	70.80 ± n/a	84.79 ± n/a
Meta-baseline [6]	ResNet-12	63.17 ± 0.23	79.26 ± 0.17	68.62 ± 0.27	83.29 ± 0.18
RFS-simple [39]	ResNet-12	62.02 ± 0.63	79.64 ± 0.44	69.74 ± 0.72	84.41 ± 0.55
RFS-distill [39]	ResNet-12	64.82 ± 0.60	82.14 ± 0.43	71.52 ± 0.69	86.03 ± 0.49
SKD-GEN0	ResNet-12	65.93 ± 0.81	83.15 ± 0.54	71.69 ± 0.91	**86.66 ± 0.60**
SKD-GEN1	ResNet-12	**67.04 ± 0.85**	**83.54 ± 0.54**	**72.03 ± 0.91**	86.50 ± 0.58

Table 1: FSL results on miniImageNet [41] and tieredImageNet [34] datasets, with mean accuracy and 95% confidence interval. †results obtained by training on train+val sets.

Datasets: We evaluate SKD on five widely used FSL benchmarks. These include two datasets which are subsets of the ImageNet i.e., miniImageNet [41] and tieredImageNet [34], the other two which are splits of CIFAR100 i.e., CIFAR-FS [2] and FC100 [28], and a very large-scale Meta-dataset [40] (composed of multiple datasets of diverse nature). For miniImageNet [41], we use the split proposed in [32], with 64, 16 and 20 classes for training, validation and testing, respectively. The tieredImageNet [34] contains 608 classes which are semantically grouped into 34 high-level classes, that are further divided into 20, 6 and 8 for training, validation, and test splits, thus ensuring diversity. CIFAR-FS [2] has a random split of 100 classes into 64, 16 and 20 for training, validation, and testing, while FC100 [28] uses splits similar to tieredImageNet, making them more diverse. FC100 has 60, 20, 20 classes for training, validation, and testing respectively. For Meta-dataset [40], the model is trained on ImageNet-training-split (1000 ImageNet classes grouped into 712 training, 158 validation and 130 test) and evaluated on three datasets within Meta-dataset including Describable Textures [7], MSCOCO [24] and ImageNet test split.

Method	Backbone	CIFAR-FS, 5-way		FC100, 5-way	
		1-shot	5-shot	1-shot	5-shot
MAML [12]	32-32-32-32	58.9 ± 1.9	71.5 ± 1.0	-	-
Prototypical Networks† [36]	64-64-64-64	55.5 ± 0.7	72.0 ± 0.6	35.3 ± 0.6	48.6 ± 0.6
Relation Networks [38]	64-96-128-256	55.0 ± 1.0	69.3 ± 0.8	-	-
R2D2 [2]	96-192-384-512	65.3 ± 0.2	79.4 ± 0.1	-	-
TADAM [28]	ResNet-12	-	-	40.1 ± 0.4	56.1 ± 0.4
Shot-Free [33]	ResNet-12	69.2 ± n/a	84.7 ± n/a	-	-
TEWAM [29]	ResNet-12	70.4 ± n/a	81.3 ± n/a	-	-
Prototypical Networks† [36]	ResNet-12	72.2 ± 0.7	83.5 ± 0.5	37.5 ± 0.6	52.5 ± 0.6
Boosting [16]	WRN-28-10	73.6 ± 0.3	86.0 ± 0.2	-	-
MetaOptNet [22]	ResNet-12	72.6 ± 0.7	84.3 ± 0.5	41.1 ± 0.6	55.5 ± 0.6
RFS-simple [39]	ResNet-12	71.5 ± 0.8	86.0 ± 0.5	42.6 ± 0.7	59.1 ± 0.6
RFS-distill [39]	ResNet-12	73.9 ± 0.8	86.9 ± 0.5	44.6 ± 0.7	60.9 ± 0.6
SKD-GEN0	ResNet-12	74.5 ± 0.9	88.0 ± 0.6	46.4 ± 0.8	63.3 ± 0.7
SKD-GEN1	ResNet-12	**76.9** ± **0.9**	**88.9** ± **0.6**	**47.3** ± **0.8**	**63.8** ± **0.7**

Table 2: FSL results on CIFAR-FS [2] and FC100 [28] datasets, with mean accuracy and 95% confidence interval. †results obtained by training on train+val sets.

4.1 Few-shot learning results

Our results shown in Table 1 (miniImageNet [41] & tieredImageNet [34] datasets) and Table 2 (CIFAR-FS [2] & FC100 [28] datasets) suggest that the proposed SKD consistently outperforms the existing methods across all datasets. Even, our Gen-0 alone performs better than the current state-of-the-art (SOTA) methods by a considerable margin. For example, SKD Gen-0 model surpasses SOTA performance on miniImageNet by ∼1% on both 5-way 1-shot and 5-way 5-shot tasks. The same can be observed on other datasets. Compared to feature embedding based RFS [39], SKD shows an improvement of 3.91% on 5-way 1-shot and 3.51% on 5-way 5-shot learning. A similar trend is observed across other evaluated datasets with consistent 2-3% gains over RFS [39]. This is due to the self-supervised learning strategy which enables SKD to learn diverse and generalizable embeddings.

Gen-1 incorporates knowledge distillation and proves even more effective compared with Gen-0. On miniImageNet, we achieve 67.04% and 83.54% on 5-way 1-shot and 5-way 5-shot learning tasks, respectively. These are gains of 2.22% and 1.4% on 5-way 1-shot and 5-way 5-shot tasks. Similar consistent gains of 2-3% over SOTA results can be observed across other evaluated datasets. Note that, RFS-distill [39] uses multiple iterations (up to 3-4 generations) for model distillation, while SKD only uses a single generation for the distillation. We attribute our gain to the way we use knowledge distillation to constrain changes in the embedding space, while minimizing the embedding distance between images and their augmented pairs, thus enhancing representation capabilities of the model.

4.2 Ablation Studies and Analysis

Choices of loss function: We study the impact of different contributions by progressively integrating them into our pipeline in Table 3. We first evaluate SKD with and without the self-supervision loss. If we train the Gen-0 with only cross entropy loss, which is same as RFS-simple [39], the model achieves $71.5 \pm 0.8\%$ and $62.02 \pm 0.63\%$ for 5-way 1-shot task on CIFAR-FS and miniImageNet, respectively. Then, if we train the Gen-0 with additional self supervision, the performance improves to $74.5 \pm 0.9\%$ and $65.93 \pm 0.81\%$. This shows an absolute gain of 3.0% and 3.91%, by incorporating our proposed self-supervision. Additionally, if we only keep knowledge distillation for Gen-1, we can see that self-supervision for Gen-0 has a clear impact on the next generation. As shown in Table 3, self-supervision at Gen-0 is responsible for 2% performance improvement on Gen-1. Further, during Gen-1, the

Generation	Loss Function	CIFAR-FS, 5-way 1-shot	CIFAR-FS, 5-way 5-shot	miniImageNet, 5-way 1-shot	miniImageNet, 5-way 5-shot
GEN-0	$\mathcal{L}_{CE}$	71.5 ± 0.8	86.0 ± 0.5	62.02 ± 0.63	79.64 ± 0.44
	$\mathcal{L}_{CE} + \alpha\mathcal{L}_{SS}$	74.5 ± 0.9	88.0 ± 0.6	65.93 ± 0.81	83.15 ± 0.54
GEN-1	$\mathcal{L}_{CE} \rightarrow \mathcal{L}_{KD}$	73.9 ± 0.8	86.9 ± 0.5	64.82 ± 0.60	82.14 ± 0.43
	$\mathcal{L}_{CE} \rightarrow \mathcal{L}_{KD} + \beta\mathcal{L}_{\ell_2}$	74.9 ± 1.0	87.6 ± 0.6	64.76 ± 0.84	81.84 ± 0.54
	$\mathcal{L}_{CE} + \alpha\mathcal{L}_{SS} \rightarrow \mathcal{L}_{KD}$	75.6 ± 0.9	88.7 ± 0.6	66.48 ± 0.84	$\mathbf{83.64 \pm 0.53}$
	$\mathcal{L}_{CE} + \alpha\mathcal{L}_{SS} \rightarrow \mathcal{L}_{KD} + \beta\mathcal{L}_{\ell_2}$	$\mathbf{76.9 \pm 0.9}$	$\mathbf{88.9 \pm 0.6}$	$\mathbf{67.04 \pm 0.85}$	83.54 ± 0.54

Table 3: FSL results on CIFAR-FS [2] and FC100 [28], with different combinations of loss functions for Gen-0 and Gen-1. For Gen-1, the loss functions on the left side of the arrow were used to train the Gen-0 model.

advantage of using the $\mathcal{L}_{\ell_2}$ loss to bring logits of rotated augmentations closer, is demonstrated in Table 3. We can see that, for the Gen-0 models trained either on $\mathcal{L}_{ce}$ or $\mathcal{L}_{ce} + \alpha\mathcal{L}_{ss}$, addition of $\mathcal{L}_{\ell_2}$ loss during Gen-1 gives about $\sim 1\%$ gain compared with using knowledge distillation only. We can also see that, in both 1-shot and 5-shot cases having the $\mathcal{L}_{\ell_2}$ loss term during Gen-1 helps to improve the performance. In CIFAR-FS, only using distillation loss at GEN-1 gives 73.9 ± 0.8 and 86.9 ± 0.5 for 1-shot and 5-shot respectively. However, having additional $L2$ loss term during the GEN-1 optimization helps to improve the performance to 74.9 ± 1.0 (1-shot) and 87.6 ± 0.6 (5-shot). Finally note that, our Gen-1 training does not use any class labels during this stage of the training. During this stage we are only interested in making the model more discriminative towards *unseen* classes, not on seen classes. These empirical evaluations clearly establish individual importance of different components (self-supervision, knowledge distillation and ensuring equivariance representations in the output space) of our proposed two stage approach.

Self-supervision Type	Generation 0, 5-way 1-shot	Generation 0, 5-way 5-shot	Generation 1, 5-way 1-shot	Generation 1, 5-way 5-shot
None	71.5 ± 0.8	86.0 ± 0.5	73.9 ± 0.8	86.9 ± 0.5
Rotation	$\mathbf{74.5 \pm 0.9}$	$\mathbf{88.0 \pm 0.6}$	$\mathbf{76.9 \pm 0.9}$	$\mathbf{88.9 \pm 0.6}$
Location	74.1 ± 0.9	$\mathbf{88.0 \pm 0.6}$	76.2 ± 0.9	87.8 ± 0.6
Jigsaw puzzle	72.7 ± 0.8	87.1 ± 0.6	75.0 ± 0.8	88.0 ± 0.5
Gaussian Blur	68.8 ± 0.9	84.9 ± 0.6	71.6 ± 0.9	85.7 ± 0.6

Figure 3: Performance of SKD on CIFAR-FS [2] dataset for different self-supervision tasks.

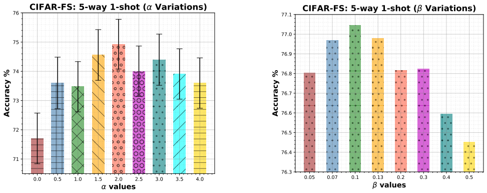

Figure 4: Ablation study on the sensitivity of the loss coefficient hyper-parameters α and β.

Choices of self-supervision: We further investigate different choices of self-supervision. **(a)** Instead of rotations based self-supervision, we use a 2×2 crop of an image, and train the final classifier to predict the correct crop quadrant [37]. **(b)** We apply Gaussian blur with varying strengths and define the pretext task in terms of predicting the level of degradation (in discrete manner). **(c)** The unordered patches are input to the model and the network is trained to solve the proxy task of re-arranging the patches in their correct order (jigsaw puzzle [26]). The results in Table 3 show that the crop-based location prediction and jigsaw puzzle self-supervision methods perform favorably well compared with the state-of-the-art FSL methods, though they perform slightly lower than the rotations based self-supervision. We find this trend to be consistent with self-supervised learning literature [19], where changing global information while preserving local information generally helps more. Thus rotation based proxy task provides an edge over localized transformations.

Sequential vs Parallel heads: Sequential design is important to SKD. We ran SKD with parallel heads (as in [16]) on mini-ImageNet and achieve an accuracy of 64.29±0.80% and 80.58±0.53% (1 & 5 shots respectively). These results suggest that, under the same settings, our proposed sequential design performs favorably well against the parallel heads by achieving 65.93±0.81% and 83.15±0.54% (1 & 5 shots). We believe that with a parallel head design, the network can find a simple linearly separable solution like $[A^T, B^T]^T$, where A only capture the class distribution while B only capture the distribution of data augmentations independent of the class. However, our motivation is to capture both properties in the output logit space via a cascaded design, thereby helping learn the true data manifold for FSL settings.

Base Class Performance: While fine-tuning for novel classes, FSL methods can forget base-class information, which results in a performance drop over the original set of base classes. Since in practical settings, we require models to retain old knowledge while learning new classes, it is interesting to study the base class performance of fine-tuned models. Our experiments show that SKD can predict base classes with high accuracy i.e., miniImageNet: 81.9% and tieredImageNet: 73.6%. These strong results suggest that SKD retains base class information and generalizes equally well to novel and old classes.

Time Complexity SKD has a time complexity of $\mathcal{O}(2 \times T)$, where T is the time required to train one generation. In comparison, RFS [39] has time complexity of $\mathcal{O}(n \times T)$, where n is the number of generations (usually 3-4). Using a single Tesla V100 GPU on CIFAR-FS, for the first generation, both RFS and SKD take approx. the same time, i.e., $T = 88$ minutes. The complete training time on CIFAR-FS of RFS is $\sim$ 4 hours, while SKD only takes $\sim$ 2 hours.

Visualizing SKD Behaviour: To illustrate that a meaningful feature embedding is learned in our proposed two-stage self-supervised knowledge distillation scheme, we show a tSNE visualization of the learned features in Fig. 5. The scatter plots are obtained for feature embeddings of 10 classes of CIFAR-FS test-set (never seen during training). We clearly notice weak class boundaries for the model trained without self-supervision, while our proposed self-supervised models (both Gen-0 & 1) have better class separation. Further, Gen-0 & 1 models faithfully preserve the class structure in the feature space (distance from class centers in both cases are almost the same), thanks to distillation loss.

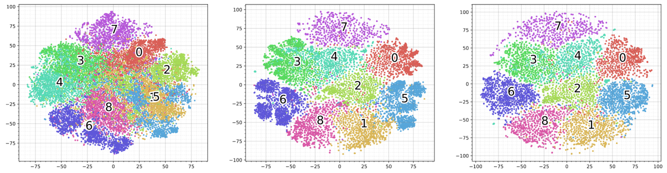

Figure 5: *Left to right*: A tSNE visualization of feature embeddings from the models trained with no-self-supervision, self-supervision (Gen-0) and self-supervision (Gen-1).

5 Conclusion

Deep learning models can easily overfit on the scarce data available in FSL settings. To enhance generalizability, existing approaches regularize the model to preserve margins or encode high-level learning behaviour via meta-learning. In this work, we take a different approach and propose to learn the true output embedding space via self-supervised learning. Our approach operates in two phases: first, the model learns to classify inputs such that the diversity in the outputs is not lost, thereby avoiding overfitting and modeling the natural output manifold structure. Once this structure is learned, our approach trains a student model that preserves the original output manifold structure while jointly maximizing the discriminability of learned representations. Our results on five popular benchmarks show the benefit of our approach where it establishes a new state-of-the-art for FSL.

References

[1] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*, volume 1. MIT press, 2017.

[2] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019.

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018.

[4] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

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

[7] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[8] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *International Conference on Learning Representations*, 2020.

[9] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *IEEE International Conference on Computer Vision*, 2015.

[10] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, 2014.

[11] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *IEEE International Conference on Computer Vision*, 2019.

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

[13] Sebastian Flennerhag, Andrei A. Rusu, Razvan Pascanu, Francesco Visin, Hujun Yin, and Raia Hadsell. Meta-learning with warped gradient descent. In *International Conference on Learning Representations*, 2020.

[14] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[15] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

[16] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *IEEE International Conference on Computer Vision*, 2019.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[18] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[19] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. doi: 10.1109/TPAMI.2020.2992393.

[20] Salman Khan, Hossein Rahmani, Syed Afaq Ali Shah, and Mohammed Bennamoun. A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision*, 8(1):1–207, 2018.

[21] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, 2015.

[22] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[23] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[25] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

[26] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, 2016.

[27] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *IEEE International Conference on Computer Vision*, 2017.

[28] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, 2018.

[29] Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *IEEE International Conference on Computer Vision*, 2019.

[30] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. in 2018 ieee. In *CVF Conference on Computer Vision and Pattern Recognition*, pages 4119–4128, 2017.

[31] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

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

[33] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *IEEE International Conference on Computer Vision*, 2019.

[34] Mengye Ren, Sachin Ravi, Eleni Triantafillou, Jake Snell, Kevin Swersky, Josh B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018.

[35] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019.

[36] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, 2017.

[37] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.

[38] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

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

[40] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2019.

[41] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, 2016.

[42] Guodong Xu, Ziwei Liu, Xiaoxiao Li, and Chen Change Loy. Knowledge distillation meets self-supervision. In *European Conference on Computer Vision*, pages 588–604. Springer, 2020.

[43] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Learning embedding adaptation for few-shot learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[44] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *IEEE international conference on computer vision*, 2019.

[45] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, 2016.