

Charting the Right Manifold: Manifold Mixup for Few-shot Learning

Puneet Mangla^{*†2}

cs17btech11029@iith.ac.in

Mayank Singh^{*1}

msingh@adobe.com

Abhishek Sinha^{*1}

abhsinha@adobe.com

Nupur Kumari^{*1}

nupkumar@adobe.com

Vineeth N Balasubramanian²

vineethnb@iith.ac.in

Balaji Krishnamurthy¹

kbalaji@adobe.com

1. Media and Data Science Research lab, Adobe

2. IIT Hyderabad, India

Abstract

Few-shot learning algorithms aim to learn model parameters capable of adapting to unseen classes with the help of only a few labeled examples. A recent regularization technique - Manifold Mixup focuses on learning a general-purpose representation, robust to small changes in the data distribution. Since the goal of few-shot learning is closely linked to robust representation learning, we study Manifold Mixup in this problem setting. Self-supervised learning is another technique that learns semantically meaningful features, using only the inherent structure of the data. This work investigates the role of learning relevant feature manifold for few-shot tasks using self-supervision and regularization techniques. We observe that regularizing the feature manifold, enriched via self-supervised techniques, with Manifold Mixup significantly improves few-shot learning performance. We show that our proposed method S2M2 beats the current state-of-the-art accuracy on standard few-shot learning datasets like CIFAR-FS, CUB, mini-ImageNet and tiered-ImageNet by 3 – 8%. Through extensive experimentation, we show that the features learned using our approach generalize to complex few-shot evaluation tasks, cross-domain scenarios and are robust against slight changes to data distribution.

1. Introduction

Deep convolutional networks (CNN's) have become a regular ingredient for numerous contemporary computer vision tasks. They have been applied to tasks such as object recognition, semantic segmentation, object detection [23, 66, 21, 24, 35] to achieve state-of-the-art performance. However, the at par performance of deep neural networks

requires huge amount of supervisory examples for training. Generally, labeled data is scarcely available and data collection is expensive for several problem statements. Hence, a major research effort is being dedicated to fields such as transfer learning, domain adaptation, semi-supervised and unsupervised learning [15, 29, 46] to alleviate this requirement of enormous amount of examples for training.

A related problem which operates in the low data regime is few-shot classification. In few-shot classification, the model is trained on a set of classes (base classes) with abundant examples in a fashion that promotes the model to classify unseen classes (novel classes) using few labeled instances. The motivation for this stems from the hypothesis that an appropriate prior should enable the learning algorithm to solve consequent tasks more easily. Biologically speaking, humans have a high capacity to generalize and extend the prior knowledge to solve new tasks using only small amount of supervision. One of the promising approach to few-shot learning utilizes meta-learning framework to optimize for such an initialization of model parameters such that adaptation to the optimal weights of classifier for novel classes can be reached with few gradient updates [50, 14, 54, 40]. Some of the work also includes leveraging the information of similarity between images [63, 58, 60, 3, 16] and augmenting the training data by hallucinating additional examples [20, 65, 56]. Another class of algorithms [49, 17] learns to directly predict the weights of the classifier for novel classes.

Few-shot learning methods are evaluated using N -way K -shot classification framework where N classes are sampled from a set of novel classes (not seen during training) with K examples for each class. Usually, the few-shot classification algorithm has two separate learning phases. In the first phase, the training is performed on base classes to develop robust and general-purpose representation aimed to be useful for classifying novel classes. The second phase of training exploits the learning from previous phase in the

^{*} Authors contributed equally

[†] Work done during Adobe MDSR internship

form of a prior to perform classification over novel classes. The transfer learning approach serves as the baseline which involves training a classifier for base classes and then subsequently learning a linear classifier on the penultimate layer of the previous network to classify the novel classes [7].

Learning feature representations that generalize to novel classes is an essential aspect of few-shot learning problem. This involves learning a feature manifold that is relevant for novel classes. Regularization techniques enables the models to generalize to unseen test data that is disjoint from training data. It is frequently used as a supplementary technique alongside standard learning algorithms [30, 27, 5, 62, 28]. In particular for classification problems, Manifold Mixup [62] regularization leverages interpolations in deep hidden layer to improve hidden representations and decision boundaries at multiple layers.

In Manifold Mixup [62], the authors show improvement in classification task over standard image deformations and augmentations. Also, some work in self-supervision [18, 68, 11] explores to predict the type of augmentation applied and enforces feature representation to become invariant to image augmentations to learn robust visual features. Inspired by this link, we propose to unify the training of few-shot classification with self-supervision techniques and Manifold Mixup [62]. The proposed technique employs self-supervision loss over the given labeled data unlike in semi-supervised setting that uses additional unlabeled data and hence our approach doesn't require any extra data for training.

Many of the recent advances in few-shot learning exploit the meta-learning framework, which simulates the training phase as that of the evaluation phase in the few-shot setting. However, in a recent study [7], it was shown that learning a cosine classifier on features extracted from deeper networks also performs quite well on few-shot tasks. Motivated by this observation, we focus on utilizing self-supervision techniques augmented with Manifold Mixup in the domain of few-shot tasks using cosine classifiers.

Our main contributions in this paper are the following:

- We find that the regularization technique of Manifold Mixup [62] being robust to small changes in data distribution enhances the performance of few-shot tasks.
- We show that adding self-supervision loss to the training procedure, enables robust semantic feature learning that leads to a significant improvement in few-shot classification. We use rotation [18] and exemplar [11] as the self-supervision tasks.
- We observe that applying Manifold Mixup regularization over the feature manifold enriched via the self-supervision tasks further improves the performance of few-shot tasks. The proposed methodology

(S2M2) outperforms the state-of-the-art methods by 3-8% over the CIFAR-FS, CUB, *mini*-ImageNet and *tiered*-ImageNet datasets.

- We conduct extensive ablation studies to verify the efficacy of the proposed method. We find that the improvements made by our methodology become more pronounced with increasing N in the N -way K -shot evaluation and also in the cross-domain evaluation.

2. Related Work

Our work is associated with various recent development made in learning robust general-purpose visual representations, specifically few-shot learning, self-supervised learning and generalization boosting techniques.

Few-shot learning: Few-shot learning involves building a model using available training data of base classes that can classify unseen novel classes using only few examples. Few-shot learning approaches can be broadly divided into three categories - gradient based methods, distance metric based methods and hallucination based methods.

Some gradient based methods [50, 1] aim to use gradient descent to quickly adapt the model parameters suitable for classifying the novel task. The initialization based methods [14, 54, 40] specifically advocate to learn a suitable initialization of the model parameters, such that adapting from those parameters can be achieved in a few gradient steps. Distance metric based methods leverage the information about similarity between images to classify novel classes with few examples. The distance metric can either be cosine similarity [63], euclidean distance [58], CNN based distance module [60], ridge regression [3] or graph neural network [16]. Hallucination based methods [20, 65, 56] augment the limited training data for a new task by generating or hallucinating new data points.

Recently, [7] introduced a modification for the simple transfer learning approach, where they learn a cosine classifier [49, 17] instead of a linear classifier on top of feature extraction layers. The authors show that this simple approach is competitive with several proposed few-shot learning approaches if a deep backbone network is used to extract the feature representation of input data.

Self-supervised learning: This is a general learning framework which aims to extract supervisory signals by defining surrogate tasks using only the structural information present in the data. In the context of images, a pretext task is designed such that optimizing it leads to more semantic image features that can be useful for other vision tasks. Self-supervision techniques have been successfully applied to diverse set of domains, ranging from robotics to computer vision [31, 12, 57, 55, 45]. In the context of visual

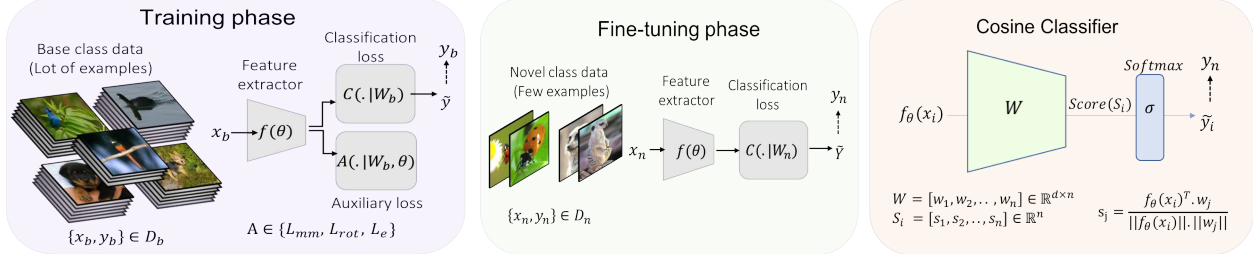


Figure 1. Flowchart for our proposed approach (S2M2) for few-shot learning. The auxiliary loss is derived from Manifold Mixup regularization and self-supervision tasks of rotation and exemplar.

data, the surrogate loss functions can be derived by leveraging the invariants in the structure of the image. In this paper, we focus on self-supervised learning techniques to enhance the representation and learn a relevant feature manifold for few-shot classification setting. We now briefly describe the recent developments in self-supervision techniques.

C. Doersch *et al.* [9] took inspiration from spatial context of a image to derive supervisory signal by defining the surrogate task of relative position prediction of image patches. Motivated by the task of context prediction, the pretext task was extended to predict the permutation of the shuffled image patches [41, 39, 43]. [18] leveraged the rotation invariance of images to create the surrogate task of predicting the rotation angle of the image. Also, the authors of [13] proposed to decouple representation learning of the rotation as pretext task from class discrimination to obtain better results. Along the lines of context-based prediction, [48] uses generation of the contents of image region based on context pixel (i.e. in-painting) and in [69, 70] the authors propose to use gray-scale image colorization as a pretext task.

Apart from enforcing structural constraints, [6] uses cluster assignments as supervisory signals for unlabeled data and works by alternating between clustering of the image descriptors and updating the network by predicting the cluster assignments. [47] defines pretext task that uses low-level motion-based grouping cues to learn visual representation. Also, [42] proposes to obtain supervision signal by enforcing the additivity of visual primitives in the patches of images and [44] proposed to learn feature representations by predicting the future in latent space by employing autoregressive models.

Some of the pretext tasks also work by enforcing constraints on the representation of the feature. A prominent example is the exemplar loss from [11] that promotes representation of image to be invariant to image augmentations. Additionally, some research effort have also been put in to define the pretext task as a combination of multiple pretext task [10, 32]. For instance, in [32] representation learning is augmented with pretext tasks of jigsaw puzzle [41], colorization [69, 70] and in-painting [48].

Generalization: Employing regularization techniques for training deep neural networks to improve their generalization performances have become standard practice in the deep learning community. Few of the commonly used regularization techniques are - dropout [59], cutout [8], Mixup [28], Manifold Mixup [62]. Mixup [28] is a specific case of Manifold Mixup [62] where the interpolation of only input data is applied. The authors in [62] claim that Manifold Mixup leads to smoother decision boundaries and flattens the class representations thereby leading to feature representation that improve the performance over a held-out validation dataset. We apply a few of these generalization techniques during the training of the backbone network over the base tasks and find that the features learned via such regularization lead to better generalization over novel tasks too. Authors of [36] provide a summary of popular regularization techniques used in deep learning.

3. Methodology

The few-shot learning setting is formalized by the availability of a dataset with data-label pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i) : i = 1, \dots, m\}$ where $\mathbf{x} \in \mathbb{R}^d$ and $y_i \in C$, C being the set of all classes. We have sufficient number of labeled data in a subset of C classes (called *base classes*), while very few labeled data for the other classes in C (called *novel classes*). Few-shot learning algorithms generally train in two phases: the first phase consists of training a network over base class data $\mathcal{D}_b = \{(\mathbf{x}_i, y_i), i = 1, \dots, m_b\}$ where $\{y_i \in C_b \subset C\}$ to obtain a feature extractor, and the second phase consists of adapting the network for novel class data $\mathcal{D}_n = \{(\mathbf{x}_i, y_i), i = 1, \dots, m_n\}$ where $\{y_i \in C_n \subset C\}$ and $C_b \cup C_n = C$. We assume that there are N_b base classes (cardinality of C_b) and N_n novel classes (cardinality of C_n). The general goal of few-shot learning algorithms is to learn rich feature representations from the abundant labeled data of base classes N_b , such that the features can be easily adapted for the novel classes using only few labeled instances.

In this work, in the first learning stage, we train a N_b -way neural network classifier:

$$g = c_{W_b} \circ f_\theta \quad (1)$$

on \mathcal{D}_b , where c_{W_b} is a cosine classifier [49, 17] and f_θ is the convolutional feature extractor, with θ parametrizing the neural network model. The model is trained with classification loss and an additional auxiliary loss which we explain soon. The second phase involves fine-tuning of the backbone model, f_θ , by freezing the feature extractor layers and training a new N_n -way cosine classifier c_{W_n} on data from k randomly sampled novel classes in \mathcal{D}_n with only classification loss. Figure 1 provides an overview of our approach S2M2 for few-shot learning.

Importantly, in our proposed methodology, we leverage self-supervision and regularization techniques [62, 18, 11] to learn general-purpose representation suitable for few-shot tasks. We hypothesize that using robust features which describes the feature manifold well is important to obtain better performance over the novel classes in the few-shot setting. In the subsequent subsections, we describe our training procedure to use self-supervision methods (such as rotation [18] and exemplar [11]) to obtain a suitable feature manifold, following which using Manifold Mixup regularization [62] provides a robust feature extractor backbone. We empirically show that this proposed methodology achieves the new state-of-the-art result on standard few-shot learning benchmark datasets.

3.1. Manifold Mixup for Few-shot Learning

Higher-layer representations in neural network classifiers have often been visualized as lying on a meaningful manifold, that provide the relevant geometry of data to solve a given task [2]. Therefore, linear interpolation of feature vectors in that space should be relevant from the perspective of classification. With this intuition, Manifold Mixup [62], a recent work, leverages linear interpolations in neural network layers to help the trained model generalize better. In particular, given input data \mathbf{x} and \mathbf{x}' with corresponding feature representations at layer l given by $f_\theta^l(\mathbf{x})$ and $f_\theta^l(\mathbf{x}')$ respectively. Assuming we use Manifold Mixup on the base classes in our work, the loss for training L_{mm} is then formulated as:

$$L_{mm} = \mathbb{E}_{(x,y) \in \mathcal{D}_b} \left[L(\text{Mix}_\lambda(f_\theta^l(\mathbf{x}), f_\theta^l(\mathbf{x}')), \text{Mix}_\lambda(y, y')) \right] \quad (2)$$

where

$$\text{Mix}_\lambda(a, b) = \lambda \cdot a + (1 - \lambda) \cdot b \quad (3)$$

The mixing coefficient λ is sampled from a $\beta(\alpha, \alpha)$ distribution and loss L is standard cross-entropy loss. We hypothesize that using Manifold Mixup on the base classes provides robust feature presentations that lead to state-of-the-art results in few-shot learning benchmarks.

Training using loss L_{mm} encourages the model to predict less confidently on linear interpolations of hidden representations. This encourages the feature manifold to have

broad regions of low-confidence predictions between different classes and thereby smoother decision boundaries, as shown in [62]. Also, models trained using this regularizer lead to flattened hidden representations for each class with less number of directions of high variance i.e. the representations of data from each class lie in a lower dimension subspace. The above-mentioned characteristics of the method make it a suitable regularization technique for generalizing to tasks with potential distribution shifts.

3.2. Charting the Right Manifold

We observed that Manifold Mixup does result in higher accuracy on few-shot tasks, as shown in Section 4.2.3. However, it still lags behind existing state-of-the-art performance, which begs the question: *Are we charting the right manifold?* In few-shot learning, novel classes introduced during test time can have a different data distribution when compared to base classes. In order to counter this distributional shift, we hypothesize that it is important to capture the right manifold when using Manifold Mixup for the base classes. To this end, we leverage self-supervision methods. Self-supervision techniques have been employed recently in many domains for learning rich, generic and meaningful feature representations. We show that the simple idea of adding auxiliary loss terms from self-supervised techniques while training the base classes provides feature representations that significantly outperform state-of-the-art for classifying on the novel classes. We now describe the self-supervised methods used in this work.

3.2.1 Self-Supervision: Towards the Right Manifold

In this work, we use two pretext tasks that have recently been widely used for self-supervision to support our claim. We describe each of these below.

Rotation [18]: In this self-supervised task, the input image is rotated by different angles, and the auxiliary aim of the model is to predict the amount of rotation applied to image. In the image classification setting, an auxiliary loss (based on the predicted rotation angle) is added to the standard classification loss to learn general-purpose representations suitable for image understanding tasks. In this work, we use a 4-way linear classifier, c_{W_r} , on the penultimate feature representation $f_\theta(\mathbf{x}^r)$ where \mathbf{x}^r is the image x rotated by r degrees and $r \in C_R = \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$, to predict one of 4 classes in C_R . In other words, similar to Eqn 1, our pretext task model is given by $g_r = c_{W_r} \circ f_\theta$. The self-supervision loss is given by:

$$L_{rot} = \frac{1}{|C_R|} * \sum_{\mathbf{x} \in \mathcal{D}_b} \sum_{r \in C_R} L(c_{W_r}(f_\theta(\mathbf{x}^r)), r) \quad (4)$$

$$L_{class} = \mathbb{E}_{(x,y) \in \mathcal{D}_b, r \in C_R} [L(x^r, y)] \quad (5)$$

where $|C_R|$ denotes the cardinality of C_R . As the self-supervision loss is defined over the given labeled data of \mathcal{D}_b , no additional data is required to implement this method. L is the standard cross-entropy loss, as before.

Exemplar [11]: Exemplar training aims at making the feature representation invariant to a wide range of image transformations such as translation, scaling, rotation, contrast and color shifts. In a given mini-batch M , we create 4 copies of each image through random augmentations. These 4 copies are the positive examples for each image and every other image in the mini-batch is a negative example. We then use hard batch triplet loss [26] with soft margin on $f_\theta(\mathbf{x})$ on the mini-batch to bring the feature representation of positive examples close together. Specifically, the loss is given as:

$$L_e = \frac{1}{4 * |M|} \sum_{\mathbf{x} \in M} \sum_{k=1}^4 \log \left(1 + \exp \left(- \max_{p \in \{1, \dots, 4\}} D(\mathbf{x}_k^i, \mathbf{x}_p^i) \right) + \min_{p \in \{1..4\}, i \neq j} D(\mathbf{x}_k^i, \mathbf{x}_p^j) \right) \quad (6)$$

Here, D is the Euclidean distance in the feature representation space $f_\theta(\mathbf{x})$ and \mathbf{x}_k^i is the k^{th} exemplar of \mathbf{x} with class label i (the appropriate augmentation). The first term inside the exp term is the maximum among distances between an image and its positive examples which we want to reduce. The second term is the minimum distance between the image and its negative examples which we want to maximize.

3.2.2 S2M2: Self-Supervised Manifold Mixup

The few-shot learning setting relies on learning robust and generalizable features that can separate base and novel classes. An important means to this end is the ability to compartmentalize the representations of base classes with generous decision boundaries, which allow the model to generalize to novel classes. Manifold Mixup provides an effective methodology to flatten representations of data from a given class into a compact region, thereby supporting this objective. However, while [62] claims that Manifold Mixup can handle minor distribution shifts, the semantic difference between base and novel classes in the few-shot setting may be more than what it can handle. We hence propose the use of self-supervision as an auxiliary loss while training the base classes, which allows the learned backbone model, f_θ , to provide feature representations with sufficient decision boundaries between classes, that allow the model to extend to the novel classes. This is evidenced in our results presented in Section 4.2.3. Our overall methodology is summarized in the steps below, and the pseudo-code of the proposed approach for training the backbone is presented in Algorithm 1.

Algorithm 1 S2M2 feature backbone training

begin

Input: $\{\mathbf{x}, y\} \in \mathcal{D}_b; \alpha; \{\mathbf{x}', y'\} \in \mathcal{D}_{val}$

Output: Backbone model f_θ

▷ Feature extractor backbone f_θ training

Initialize f_θ

for $epochs \in \{1, 2, \dots, 400\}$ **do**

Training data of size B - $(X(i), Y(i))$.

$L(\theta, X(i), Y(i)) = L_{class} + L_{ss}$

$\theta \rightarrow \theta - \eta * \nabla L(\theta, X(i), Y(i))$

end

$val_acc_prev = 0.0$

$val_acc_list = []$

▷ Fine-tuning f_θ with Manifold Mixup

while $val_acc > val_acc_prev$ **do**

Training data of size B - $(X(i), Y(i))$.

$L(\theta, X(i), Y(i)) = L_{mm} + 0.5(L_{class} + L_{ss})$

$\theta \rightarrow \theta - \eta * \nabla L(\theta, X(i), Y(i))$

$val_acc = Accuracy_{x, y \in \mathcal{D}_{val}}(W_n(f_\theta(x)), y)$

Append val_acc to val_acc_list

Update val_acc_prev with val_acc

end

return fine-tuned backbone f_θ .

Step 1: Self-supervised training: Train the backbone model using self-supervision as an auxiliary loss along with classification loss i.e. $L + L_{ss}$ where $L_{ss} \in \{L_e, L_{rot}\}$.

Step 2: Fine-tuning with Manifold Mixup: Fine-tune the above model with Manifold Mixup loss L_{mm} for a few more epochs.

After obtaining the backbone, a cosine classifier is learned over it to adapt to few-shot tasks. $S2M2_R$ and $S2M2_E$ are two variants of our proposed approach which uses L_{rot} and L_e as auxiliary loss in Step 1 respectively.

4. Experiments and Results

In this section, we present our results of few-shot classification task on different datasets and model architectures. We first describe the datasets, evaluation criteria and implementation details¹.

Datasets: We perform experiments on four standard datasets for few-shot image classification benchmark, *mini-ImageNet* [63], *tiered-ImageNet* [52], CUB [64] and CIFAR-FS [4]. *mini-ImageNet* consists of 100 classes from the ImageNet [53] which are split randomly into 64 base, 16 validation and 20 novel classes. Each class has 600 samples of size 84×84 . *tiered-ImageNet* consists of 608 classes randomly picked from ImageNet [53] which are split randomly into 351 base, 97 validation and 160 novel classes. In total, there are 779, 165 images of size 84×84 . **CUB** contains 200 classes with total 11, 788 images of size 84×84 . The base, validation and novel split is 100, 50 and

¹https://github.com/nupurkmr9/S2M2_fewshot

Method	<i>mini-ImageNet</i>		<i>tiered-ImageNet</i>		CUB		CIFAR-FS	
	1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
MAML [14]	54.69 \pm 0.89	66.62 \pm 0.83	51.67 \pm 1.81	70.30 \pm 0.08	71.29 \pm 0.95	80.33 \pm 0.70	58.9 \pm 1.9	71.5 \pm 1.0
ProtoNet [58]	54.16 \pm 0.82	73.68 \pm 0.65	53.31 \pm 0.89	72.69 \pm 0.74	71.88 \pm 0.91	87.42 \pm 0.48	55.5 \pm 0.7	72.0 \pm 0.6
RelationNet [61]	52.19 \pm 0.83	70.20 \pm 0.66	54.48 \pm 0.93	71.32 \pm 0.78	68.65 \pm 0.91	81.12 \pm 0.63	55.0 \pm 1.0	69.3 \pm 0.8
LEO [54]	61.76 \pm 0.08	77.59 \pm 0.12	66.33 \pm 0.05	81.44 \pm 0.09	68.22 \pm 0.22*	78.27 \pm 0.16*	-	-
DCO [37]	62.64 \pm 0.61	78.63 \pm 0.46	65.99 \pm 0.72	81.56 \pm 0.53	-	-	72.0 \pm 0.7	84.2 \pm 0.5
Baseline++	57.53 \pm 0.10	72.99 \pm 0.43	60.98 \pm 0.21	75.93 \pm 0.17	70.4 \pm 0.81	82.92 \pm 0.78	67.50 \pm 0.64	80.08 \pm 0.32
Manifold Mixup	57.16 \pm 0.17	75.89 \pm 0.13	68.19 \pm 0.23	84.61 \pm 0.16	73.47 \pm 0.89	85.42 \pm 0.53	69.20 \pm 0.2	83.42 \pm 0.15
Rotation	63.9 \pm 0.18	81.03 \pm 0.11	73.04 \pm 0.22	87.89 \pm 0.14	77.61 \pm 0.86	89.32 \pm 0.46	70.66 \pm 0.2	84.15 \pm 0.14
<i>S2M2_R</i>	64.93 \pm 0.18	83.18 \pm 0.11	73.71 \pm 0.22	88.59 \pm 0.14	80.68 \pm 0.81	90.85 \pm 0.44	74.81 \pm 0.19	87.47 \pm 0.13

Table 1. Comparison with prior/current state of the art methods on *mini-ImageNet*, *tiered-ImageNet*, CUB and CIFAR-FS dataset. The accuracy with * denotes our implementation of LEO using their publicly released code

Dataset	Method	ResNet-18		ResNet-34		WRN-28-10	
		1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
<i>mini-ImageNet</i>	Baseline++	53.56 \pm 0.32	74.02 \pm 0.13	54.41 \pm 0.21	74.14 \pm 0.19	57.53 \pm 0.10	72.99 \pm 0.43
	Mixup ($\alpha = 1$)	56.12 \pm 0.17	73.42 \pm 0.13	56.19 \pm 0.17	73.05 \pm 0.12	59.65 \pm 0.34	77.52 \pm 0.52
	Manifold Mixup	55.77 \pm 0.23	71.15 \pm 0.12	55.40 \pm 0.37	70.0 \pm 0.11	57.16 \pm 0.17	75.89 \pm 0.13
	Rotation	58.96 \pm 0.24	76.63 \pm 0.12	61.13 \pm 0.2	77.05 \pm 0.35	63.9 \pm 0.18	81.03 \pm 0.11
	Exemplar	56.39 \pm 0.17	76.33 \pm 0.14	56.87 \pm 0.17	76.90 \pm 0.17	62.2 \pm 0.45	78.8 \pm 0.15
	<i>S2M2_E</i>	56.80 \pm 0.2	76.54 \pm 0.14	56.92 \pm 0.18	76.97 \pm 0.18	62.33 \pm 0.25	79.35 \pm 0.16
	<i>S2M2_R</i>	64.06 \pm 0.18	80.58 \pm 0.12	63.74 \pm 0.18	79.45 \pm 0.12	64.93 \pm 0.18	83.18 \pm 0.11
CUB	Baseline++	67.68 \pm 0.23	82.26 \pm 0.15	68.09 \pm 0.23	83.16 \pm 0.3	70.4 \pm 0.81	82.92 \pm 0.78
	Mixup ($\alpha = 1$)	68.61 \pm 0.64	81.29 \pm 0.54	67.02 \pm 0.85	84.05 \pm 0.5	68.15 \pm 0.11	85.30 \pm 0.43
	Manifold Mixup	70.57 \pm 0.71	84.15 \pm 0.54	72.51 \pm 0.94	85.23 \pm 0.51	73.47 \pm 0.89	85.42 \pm 0.53
	Rotation	72.4 \pm 0.34	84.83 \pm 0.32	72.74 \pm 0.46	84.76 \pm 0.62	77.61 \pm 0.86	89.32 \pm 0.46
	Exemplar	68.12 \pm 0.87	81.87 \pm 0.59	69.93 \pm 0.37	84.25 \pm 0.56	71.58 \pm 0.32	84.63 \pm 0.57
	<i>S2M2_E</i>	71.81 \pm 0.43	86.22 \pm 0.53	72.67 \pm 0.27	84.86 \pm 0.13	74.89 \pm 0.36	87.48 \pm 0.49
	<i>S2M2_R</i>	71.43 \pm 0.28	85.55 \pm 0.52	72.92 \pm 0.83	86.55 \pm 0.51	80.68 \pm 0.81	90.85 \pm 0.44
CIFAR-FS	Baseline++	59.67 \pm 0.90	71.40 \pm 0.69	60.39 \pm 0.28	72.85 \pm 0.65	67.5 \pm 0.64	80.08 \pm 0.32
	Mixup ($\alpha = 1$)	56.60 \pm 0.11	71.49 \pm 0.35	57.60 \pm 0.24	71.97 \pm 0.14	69.29 \pm 0.22	82.44 \pm 0.27
	Manifold Mixup	60.58 \pm 0.31	74.46 \pm 0.13	58.88 \pm 0.21	73.46 \pm 0.14	69.20 \pm 0.2	83.42 \pm 0.15
	Rotation	59.53 \pm 0.28	72.94 \pm 0.19	59.32 \pm 0.13	73.26 \pm 0.15	70.66 \pm 0.2	84.15 \pm 0.14
	Exemplar	59.69 \pm 0.19	73.30 \pm 0.17	61.59 \pm 0.31	74.17 \pm 0.37	70.05 \pm 0.17	84.01 \pm 0.22
	<i>S2M2_E</i>	61.95 \pm 0.11	75.09 \pm 0.16	62.48 \pm 0.21	73.88 \pm 0.30	72.63 \pm 0.16	86.12 \pm 0.26
	<i>S2M2_R</i>	63.66 \pm 0.17	76.07 \pm 0.19	62.77 \pm 0.23	75.75 \pm 0.13	74.81 \pm 0.19	87.47 \pm 0.13

Table 2. Results on *mini-ImageNet*, CUB and CIFAR-FS dataset over different network architecture.

50 classes. **CIFAR-FS** is created by randomly splitting 100 classes of CIFAR-100 [34] into 64 base, 16 validation and 20 novel classes. The images are of size 32×32 .

Evaluation Criteria: We evaluate experiments on 5-way 1-shot and 5-way 5-shot [63] classification setting i.e using 1 and 5 labeled instances of each of the 5 classes as training data and Q instances each from the same classes as testing data. For *tiered-ImageNet*, *mini-ImageNet* and CIFAR-FS we report the average classification accuracy over 10000 tasks where $Q = 599$ for 1-Shot and $Q = 595$ for 5-Shot tasks respectively. For CUB we report average classification accuracy with $Q = 15$ over 600 tasks. We compare our approach *S2M2_R* against the current state-of-the-art methods, LEO [54] and DCO [37] in Section 4.2.3.

4.1. Implementation Details

We perform experiments on three different model architecture: ResNet-18, ResNet-34 [22] and WRN-28-10 [67]

which is a Wide Residual Network of 28 layers and width factor 10. For *tiered-ImageNet* we only perform experiments with WRN-28-10 architecture. Average pooling is applied at the last block of each architecture for getting feature vectors. ResNet-18 and ResNet-34 models have 512 dimensional output feature vector and WRN-28-10 has 640 dimensional feature vector. For training ResNet-18 and ResNet-34 architectures, we use Adam [33] optimizer for *mini-ImageNet* and CUB whereas SGD optimizer for CIFAR-FS. For WRN-28-10 training, we use Adam optimizer for all datasets.

4.2. Performance Evaluation over Few-shot Tasks

In this subsection, we report the result of few shot learning over our proposed methodology and its variants.

4.2.1 Using Manifold Mixup Regularization

All experiments using Manifold Mixup [62] randomly sample a hidden layer (including input layer) at each step to

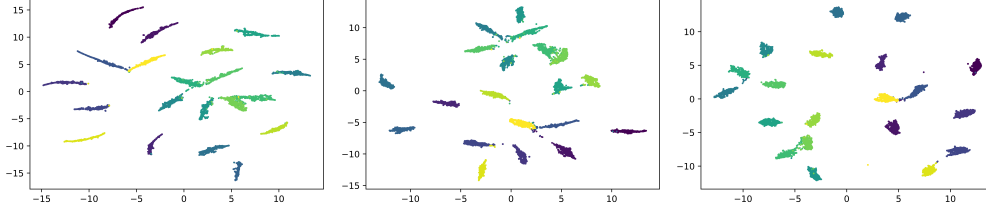


Figure 2. UMAP (2-dim) [38] plot for feature vectors of examples from novel classes of *mini-ImageNet* using Baseline++, Rotation, $S2M2_R$ (left to right).

apply mixup as described in equation 3 for the mini-batch with mixup coefficient (λ) sampled from a $\beta(\alpha, \alpha)$ distribution with $\alpha = 2$. We compare the performance of Manifold Mixup [62] with Baseline++ [7] and Mixup [28]. The results are shown in table 2. We can see that the boost in few-shot accuracy from the two aforementioned mixup strategies is significant when model architecture is deep (WRN-28-10). For shallower backbones (ResNet-18 and ResNet-34), the results are not conclusive.

4.2.2 Using Self-supervision as Auxiliary Loss

We evaluate the contribution of rotation prediction [18] and exemplar training [11] as an auxiliary task during backbone training for few-shot tasks. Backbone model is trained with both classification loss and auxiliary loss as explained in section 3.2.1. For exemplar training, we use random cropping, random horizontal/vertical flip and image jitter randomization [68] to produce 4 different positive variants of each image in the mini-batch. Since exemplar training is computationally expensive, we fine-tune the baseline++ model for 50 epochs using both exemplar and classification loss.

The comparison of above techniques with Baseline++ is shown in table 2. As we see, by selecting rotation and exemplar as an auxiliary loss there is a significant improvement from Baseline++ (7 – 8%) in most cases. Also, the improvement is more prominent for deeper backbones like WRN-28-10.

4.2.3 Our Approach: $S2M2$

We first train the backbone model using self-supervision (exemplar or rotation) as auxiliary loss and then fine-tune it with Manifold Mixup as explained in section 3.2.2. The results are shown in table 2. We compare our approach with current state-of-the-art [54, 37] and other existing few-shot methods [58, 61] in Table 1. As we can observe from table, our approach $S2M2_R$ beats the most recent state-of-the-art results, LEO [54] and DCO [37], by a significant margin on all four datasets. We find that using only rotation prediction as an auxiliary task during backbone training also outperforms the existing state-of-the-art methods on all datasets except CIFAR-FS.

Method	5-way		10-way		15-way		20-way	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Baseline++	57.53	72.99	40.43	56.89	31.96	48.2	26.92	42.8
LEO [54]	61.76	77.59	45.26	64.36	36.74	56.26	31.42	50.48
DCO [37]	62.64	78.63	44.83	64.49	36.88	57.04	31.5	51.25
Manifold-Mixup	57.16	75.89	42.46	62.48	34.32	54.9	29.24	48.74
Rotation	63.9	81.03	47.77	67.2	38.4	59.59	33.21	54.16
$S2M2_R$	64.93	83.18	50.4	70.93	41.65	63.32	36.5	58.36

Table 3. Mean few-shot accuracy on *mini-ImageNet* as N increases in N -way K -shot classification.

5. Discussion and Ablation Studies

To understand the significance of learned feature representation for few-shot tasks, we perform various experiments and analyze the findings in this section. We choose *mini-ImageNet* as the primary dataset with WRN-28-10 backbone for the following experiments.

Effect of varying N in N -way classification: For extensive evaluation, we test our proposed methodology in complex few-shot settings. We vary N in N -way K -shot evaluation criteria from 5 to 10, 15 and 20. The corresponding results are reported in table 3. We observe that our approach $S2M2_R$ outperforms other techniques by a significant margin. The improvement becomes more pronounced for $N > 5$. Figure 2 shows the 2-dimensional UMAP [38] plot of feature vectors of novel classes obtained from different methods. It shows that our approach has more segregated clusters with less variance. This supports our hypothesis that using both self supervision and Manifold Mixup regularization helps in learning feature representations with well separated margin between novel classes.

Cross-domain few-shot learning: We believe that in practical scenarios, there may be a significant domain-shift between the base classes and novel classes. Therefore, to further highlight the significance of selecting the right manifold for feature space, we evaluate the few-shot classification performance over cross-domain dataset : *mini-ImageNet* \Rightarrow *CUB* (coarse-grained to fine-grained distribution) using Baseline++, Manifold Mixup [62], Rotation [68] and $S2M2_R$. We train the feature backbone with the base classes of *mini-ImageNet* and evaluate its performance

Method	<i>mini-ImageNet</i> \Rightarrow CUB	
	1-Shot	5-Shot
DCO [37]	44.79 \pm 0.75	64.98 \pm 0.68
Baseline++	40.44 \pm 0.75	56.64 \pm 0.72
Manifold Mixup	46.21 \pm 0.77	66.03 \pm 0.71
Rotation	48.42 \pm 0.84	68.40 \pm 0.75
<i>S2M2_R</i>	48.24 \pm 0.84	70.44 \pm 0.75

Table 4. Comparison in cross-domain dataset scenario.

Method	Base + Validation	
	1-Shot	5-Shot
LEO [54]	61.76 \pm 0.08	77.59 \pm 0.12
DCO [37]	64.09 \pm 0.62	80.00 \pm 0.45
Baseline++	61.10 \pm 0.19	75.23 \pm 0.12
Manifold Mixup	61.10 \pm 0.27	77.69 \pm 0.21
Rotation	65.98 \pm 0.36	81.67 \pm 0.08
<i>S2M2_R</i>	67.13 \pm 0.13	83.6 \pm 0.34

Table 5. Effect of using the union of base and validation class for training the backbone f_{θ} .

over the novel classes of CUB (to highlight the domain-shift). We report the corresponding results in table 4.

Generalization performance of supervised learning over base classes: The results in table 2 and 3 empirically support the hypothesis that our approach learns a feature manifold that generalizes to novel classes and also results in improved performance on few-shot tasks. This generalization of the learned feature representation should also hold for base classes. To investigate this, we evaluate the performance of backbone model over the validation set of the ImageNet dataset and the recently proposed ImageNetV2 dataset[51]. ImageNetV2 was proposed to test the generalizability of the ImageNet trained models and consists of images having slightly different data distribution from the ImageNet. We further test the performance of backbone model over some common visual perturbations and adversarial attack. We randomly choose 3 of the 15 different perturbation techniques - pixelation, brightness, contrast, with 5 varying intensity values, as mentioned in the paper [25]. For adversarial attack, we use the FGSM [19] with $\epsilon = 1.0/255.0$. All the evaluation is over the 64 classes of *mini-ImageNet* used for training the backbone model. The results are shown in table 6. It can be seen that *S2M2_R* has the best generalization performance for the base classes also.

Effect of using the union of base and validation classes: We test the performance of few-shot tasks after merging the validation classes into base classes. In table 5, we see a considerable improvement over the other approaches using the same extended data, supporting the generalizability claim

Methods	I	I2	P	C	B	Adv
Baseline++	80.75	81.47	70.54	47.11	74.36	19.75
Rotation	82.21	83.91	71.9	50.84	76.26	20.5
Manifold Mixup	83.75	87.19	75.22	57.57	78.54	44.97
<i>S2M2_R</i>	85.28	88.41	75.66	60.0	79.77	28.0

Table 6. Validation set top-1 accuracy of different approaches over base classes and its perturbed variants (I:ImageNet; I2:ImageNetv2; P:Pixelation noise; C: Contrast noise; B: Brightness; Adv: Aversarial noise)

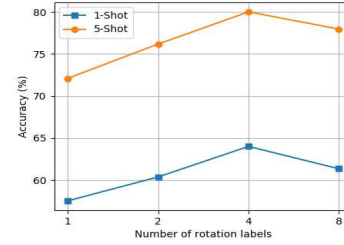


Figure 3. Effect of increasing the number of self-supervised (degrees of rotation) labels.

of the proposed method.

Different levels of self-supervision: We conduct a separate experiment to evaluate the performance of the model by varying the difficulty of self-supervision task; specifically the number of angles to predict in rotation task. We change the number of rotated versions of each image to 1 (0°), 2 (0°, 180°), 4 (0°, 90°, 180°, 270°) and 8 (0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°) and record the performance over the novel tasks for each of the corresponding 4 variants. Figure 3 shows that the performance improves with increasing the number of rotation variants till 4, after which the performance starts to decline.

6. Conclusion

We observe that learning feature representation with relevant regularization and self-supervision techniques lead to consistent improvement of few-shot learning tasks on a diverse set of image classification datasets. Notably, we demonstrate that feature representation learning using both self-supervision and classification loss and then applying Manifold Mixup over it, outperforms prior state-of-the-art approaches in few-shot learning. We do extensive experiments to analyze the effect of architecture and efficacy of learned feature representations in few-shot setting. This work opens up a pathway to further explore the techniques in self-supervision and generalization techniques to improve computer vision tasks specifically in low-data regime. Finally, our findings highlight the merits of learning a robust representation that helps in improving the performance on few-shot tasks.

References

- [1] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989, 2016.
- [2] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- [3] L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. *ICLR*, 2018.
- [4] L. Bertinetto, J. F. Henriques, P. H. S. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. *CoRR*, abs/1805.08136, 2018.
- [5] C. M. Bishop. Neural networks for pattern recognition. *Oxford university press*, 1995.
- [6] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.
- [7] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. Wang, and J.-B. Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.
- [8] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.
- [9] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.
- [10] C. Doersch and A. Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017.
- [11] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NIPS*, 2014.
- [12] F. Ebert, S. Dasari, A. X. Lee, S. Levine, and C. Finn. Robustness via retrying: Closed-loop robotic manipulation with self-supervised learning. In *CoRL*, 2018.
- [13] Z. Feng, C. Xu, and D. Tao. Self-supervised representation learning by rotation feature decoupling. In *CVPR*, 2019.
- [14] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.
- [15] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.
- [16] V. Garcia and J. Bruna. Few-shot learning with graph neural networks. *ICLR*, 2017.
- [17] S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. *CVPR*, 2018.
- [18] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.
- [19] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015.
- [20] B. Hariharan and R. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3018–3027, 2017.
- [21] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
- [23] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International conference on computer vision (ICCV)*, 2015.
- [24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [25] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.
- [26] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.
- [27] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012.
- [28] Y. N. D. D. L.-P. Hongyi Zhang, Moustapha Cisse. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*, 2018.
- [29] Y.-C. Hsu, Z. Lv, and Z. Kira. Learning to cluster in order to transfer across domains and tasks. In *ICLR*, 2018.
- [30] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML*, 2015.
- [31] E. Jang, C. Devin, V. Vanhoucke, and S. Levine. Grasp2vec: Learning object representations from self-supervised grasping. In *CoRL*, 2018.
- [32] D. Kim, D. Cho, D. Yoo, and I. S. Kweon. Learning image representations by completing damaged jigsaw puzzles. In *WACV*, 2018.
- [33] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [34] A. Krizhevsky et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.
- [35] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
- [36] J. Kukačka, V. Golkov, and D. Cremers. Regularization for deep learning: A taxonomy. *arXiv:1710.10686*, 2017.
- [37] K. Lee, S. Maji, A. Ravichandran, and S. Soatto. Meta-learning with differentiable convex optimization. *CoRR*, abs/1904.03758, 2019.
- [38] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
- [39] T. N. Mundhenk, D. Ho, and B. Y. Chen. Improvements to context based self-supervised learning. In *CVPR*, 2018.

- [40] A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.
- [41] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.
- [42] M. Noroozi, H. Pirsiavash, and P. Favaro. Representation learning by learning to count. In *ICCV*, 2017.
- [43] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *CVPR*, 2018.
- [44] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018.
- [45] A. Owens and A. A. Efros. Audio-visual scene analysis with self-supervised multisensory features. *ECCV*, 2018.
- [46] S. J. Pan and Q. Yang. A survey on transfer learning. In *Transactions on Knowledge and Data Engineering (TKDE)*, 2010.
- [47] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *CVPR*, 2017.
- [48] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.
- [49] H. Qi, M. Brown, and D. G. Lowe. Low-shot learning with imprinted weights. *CVPR*, 2018.
- [50] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. *ICLR*, 2016.
- [51] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811*, 2019.
- [52] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel. Meta-learning for semi-supervised few-shot classification. *ICLR*, 2018.
- [53] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.
- [54] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019.
- [55] N. Sayed, B. Brattoli, and B. Ommer. Cross and learn: Cross-modal self-supervision. In *GCPR*, 2018.
- [56] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, R. Feris, A. Kumar, R. Giryes, and A. M. Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. *NeurIPS*, 2018.
- [57] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, and S. Levine. Time-contrastive networks: Self-supervised learning from video. *arXiv preprint arXiv:1704.06888*, 2017.
- [58] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.
- [59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [60] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.
- [61] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. *CoRR*, abs/1711.06025, 2017.
- [62] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447, 2019.
- [63] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.
- [64] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.
- [65] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7278–7286, 2018.
- [66] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [67] S. Zagoruyko and N. Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016.
- [68] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer. S⁴I: Self-supervised semi-supervised learning. *arXiv:1905.03670*, 2019.
- [69] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *ECCV*, 2016.
- [70] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017.