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

Puneet Mangla* IIT Hyderabad, India cs17btech11029@iith.ac.in

Mayank Singh* Adobe Inc, Noida, India msingh@adobe.com

Abhishek Sinha* Adobe Inc, Noida, India abhsinha@adobe.com

Nupur Kumari* Adobe Inc, Noida, India nupkumar@adobe.com

Vineeth N Balasubramanian IIT Hyderabad, India vineethnb@iith.ac.in

Balaji Krishnamurthy Adobe Inc, Noida, India kbalaji@adobe.com

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. 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. Our proposed method *S2M2* beats the current state-of-the-art accuracy on standard few-shot learning datasets like CIFAR-FS, CUB and *mini*-ImageNet by $3-8\%$. We also show that the features learned using our approach generalize to complex few-shot evaluation tasks and cross-domain scenarios.

1 Introduction and Related Work

A major research effort is being dedicated to fields such as transfer learning, domain adaptation, semi-supervised and unsupervised learning [1, 2, 3] to alleviate the requirement of enormous amount of examples for training Deep convolutional networks (CNN's). 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. Most of the few-shot learning approaches can be broadly divided into three main categories - initialization based, distance metric based and hallucination based methods.

Initialization based methods aim to learn an optimizer [4, 5] or a good model initialization [6, 7, 8] that can adapt for novel classes in few gradient steps and limited labelled examples. **Distance metric** based methods leverage the information about similarity between images (using distance metric like cosine similarity [9], euclidean distance [10]) to classify novel classes with few examples. **Hallucination based methods** [11, 12, 13] augment the limited training data for a new task by generating or hallucinating new data points.

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 like dropout [14], cutout [15], Mixup [16], manifold Mixup [17] enable the models to generalize to unseen test data that is disjoint from training data. In particular, Manifold

*Authors contributed equally

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Mixup[17] methodology showed improvement in classification task over images with standard deformations and augmentations. The authors also claimed that Manifold Mixup leads to smoother decision boundaries and flattens the class representations.

For learning robust visual features, a lot of self-supervision techniques [18, 19, 20] in the domain of semi-supervised learning also aim to predict the type of augmentations applied and enforce the feature representations to become invariant to image augmentations. [21] took inspiration from spatial context of an 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 [22, 23, 24]. [18] leveraged the rotation in-variance of images to create the surrogate task of predicting the rotation angle of the image.

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 [25], 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 learning using cosine classifiers. We also note that similar to our findings there is a parallel effort [26] along the lines of using only self-supervision techniques for boosting few-shot performance.

Our main contributions in this paper are the following:

- We observe that applying Manifold Mixup regularization over the feature manifold enriched via the self-supervision task of rotation [18] significantly improves the performance of few-shot tasks. The proposed methodology outperforms the state-of-the-art methods by 3-8% over CIFAR-FS, CUB and *mini*-ImageNet datasets.
- We show that the improvements made by our methodology become much more pronounced on increasing N in the N-way K-shot evaluation and also in the cross-domain few-shot task evaluation.

2 Methodology

We train our few-shot learning algorithm in two phases as is standard [25]: the first phase consists of training a classification network over base class data $\mathcal{D}_b = \{(\mathbf{x}_i, y_i), i = 1, \cdots, m_b\}$ where $\{y_i \in C_b\}$ to obtain a feature extractor f_θ, the second phase consists of fixing the parameter θ of feature extractor and learning a new classifier for novel class data $\mathcal{D}_n = \{(\mathbf{x}_i, y_i), i = 1, \cdots, m_n\}$ where $\{y_i \in C_n\}$. 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 transfer learning based few-shot algorithms like this 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 (disjoint from base classes) using only few labeled instances.

Importantly, in our proposed methodology, we leverage self-supervision and regularization techniques [17, 18, 20] 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[20]) to obtain a suitable feature manifold, following which using Manifold Mixup regularization [17] 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. Figure 1 in appendix provides an overview of our approach *S2M2* for few-shot learning.

2.1 Manifold Mixup for Few-shot Learning

Manifold Mixup [17], 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 $f^l_\theta(\mathbf{x})$ and $f^l_\theta(\mathbf{x}')$ respectively. Assuming we use Manifold Mixup on the base classes in our work, the loss for training L_{mm} is then formulated as:

$$L_{mm} = \mathbb{E}_{(x,y)\in\mathcal{D}_b}\Big[L\big(Mix_\lambda(f^l_\theta(\mathbf{x}), f^l_\theta(\mathbf{x}')), Mix_\lambda(y, y')\big)\Big] \tag{1}$$

Method	*mini*-Imagenet		CUB		CIFAR-FS	
	1-Shot	5-Shot	1-Shot	5-Shot	1-Shot	5-Shot
MAML [6]	54.69 ± 0.89	66.62 ± 0.83	71.29 ± 0.95	80.33 ± 0.70	58.9 ± 1.9	71.5 ± 1.0
ProtoNet [10]	54.16 ± 0.82	73.68 ± 0.65	71.88 ± 0.91	87.42 ± 0.48	55.5 ± 0.7	72.0 ± 0.6
RelationNet [27]	52.19 ± 0.83	70.20 ± 0.66	68.65 ± 0.91	81.12 ± 0.63	55.0 ± 1.0	69.3 ± 0.8
LEO [7]	61.76 ± 0.08	77.59 ± 0.12	68.22 ± 0.22	78.27 ± 0.16	-	-
DCO [28]	62.64 ± 0.61	78.63 ± 0.46	-	-	72.0 ± 0.7	84.2 ± 0.5
Manifold Mixup	57.16 ± 0.17	75.89 ± 0.13	73.47 ± 0.89	85.42 ± 0.53	69.20 ± 0.2	83.42 ± 0.15
Rotation	63.9 ± 0.18	81.03 ± 0.11	77.61 ± 0.86	89.32 ± 0.46	70.66 ± 0.2	84.15 ± 0.14
S2M2$_R$	$\mathbf{64.93 \pm 0.18}$	$\mathbf{83.18 \pm 0.11}$	$\mathbf{80.68 \pm 0.81}$	$\mathbf{90.85 \pm 0.44}$	$\mathbf{74.81 \pm 0.19}$	$\mathbf{87.47 \pm 0.13}$

Table 1: Comparison with prior/current state of the art methods on *mini*-ImageNet, CUB and CIFAR-FS dataset.

where

$$Mix_\lambda(a, b) = \lambda \cdot a + (1 - \lambda) \cdot b \tag{2}$$

The mixing coefficient λ is sampled from a $\beta(\alpha, \alpha)$ distribution and loss L is cross-entropy loss.

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 [17] leading to state-of-the-art results in few-shot learning benchmarks.

2.1.1 Self-Supervision: Towards the Right Manifold

We observed that Manifold Mixup does result in higher accuracy on few-shot tasks, as shown in Section 3.1. 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 method of rotation prediction [18] as the pretext task in our experiments. We also report the results on self-supervision task of exemplar-training [20] in appendix.

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 $r \in C_R = \{0°, 90°, 180°, 270°\}$, to predict one of 4 classes in C_R.

2.1.2 *S2M2*: Self-Supervised Manifold Mixup

We hypothesize that using self-supervision as an auxiliary loss while training the base classes, enables the backbone model, f_θ, to provide feature representations that generalize well to the novel classes. This is evidenced in our results presented in Section 3.1. Our overall methodology is summarized in the steps below.

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} refers to the loss of self-supervision task.

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. We refer our proposed methodology of *S2M2* as *S2M2*$_R$ when rotation task is used as the self-supervision task. In the appendix section, we also propose a variant of *S2M2* i.e. *S2M2*$_E$ which uses exemplar [20] as the self-supervision loss.

3 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 details[2].

[2]To improve reproducibility of our results, we will open-source our code after publication

Experimental Details We perform experiments on three standard datasets for few-shot image classification benchmark, *mini*-ImageNet [9], CUB [29] and CIFAR-FS [30]. We use WRN-28-10 [31] as feature backbone which is a Wide Residual Network of 28 layers and width factor 10. We evaluate experiments on 5-way 1-shot and 5-way 5-shot [9] classification setting i.e using 1 and 5 labeled instances of each of the 5 classes as training data and some instances each from the same classes as testing data (from novel classes). Details about the datasets, model architecture and evaluation is mentioned in appendix.

3.1 Performance evaluation over few-shot tasks

We now report the results [3] by using only Manifold Mixup, Self-supervised rotation and our proposed methodology S2M2$_R$ in table 1. We compare them with current state-of-the-art [7] [28] and other existing few-shot approaches [10] [27]. As we can observe from table, our approach *S2M2*$_R$ beats the most recent state-of-the-art results , LEO [7] and DCO [28], by a significant margin on all the three 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 *mini-Imagenet* dataset. We report the results of its variants (using exemplar training for self-supervision) in Appendix.

Table 2: Few-shot accuracy as N in N-way classification increases.

Method	10-way		15-way		20-way	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Baseline++	40.43	56.89	31.96	48.2	26.92	42.8
LEO [7]	45.26	64.36	36.74	56.26	31.42	50.48
DCO [28]	44.83	64.49	36.88	57.04	31.5	51.25
Manifold Mixup	42.46	62.48	34.32	54.9	29.24	48.74
Rotation	47.77	67.2	38.4	59.59	33.21	54.16
S2M2$_R$	**50.4**	**70.93**	**41.65**	**63.32**	**36.5**	**58.36**

Table 3: Cross-domain few-shot evaluation.

Method	***mini*-Imagenet $\Longrightarrow$ CUB**	
	1-Shot	5-Shot
DCO [28]	44.79 ± 0.75	64.98 ± 0.68
Baseline++	40.44 ± 0.75	56.64 ± 0.72
Manifold Mixup	46.21 ± 0.77	66.03 ± 0.71
Rotation	$\mathbf{48.42 \pm 0.84}$	68.40 ± 0.75
S2M2$_R$	48.24 ± 0.84	$\mathbf{70.44 \pm 0.75}$

4 Ablation Studies

To understand the significance of learned feature representation for few-shot tasks, we perform some ablations 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 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 2. We observe that our approach *S2M2*$_R$ outperforms other techniques by a significant margin. The improvement becomes more pronounced as N increases.

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 $\Longrightarrow$ CUB** (coarse-grained to fine-grained distribution) using Baseline++, Manifold Mixup [17], Rotation [19] and *S2M2*$_R$. We train the feature backbone with the base classes of *mini*-ImageNet and evaluate its performance over the novel classes of CUB (to highlight the domain-shift). We report the corresponding results in table 3.

5 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. 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 few-shot tasks.

[3] We implemented LEO for CUB dataset and report those results

References

[1] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

[2] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. In *ICLR*, 2018.

[3] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. In *Transactions on Knowledge and Data Engineering (TKDE)*, 2010.

[4] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *ICLR*, 2016.

[5] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989, 2016.

[6] Chelsea Finn, Pieter Abbeel, and Sergey 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.

[7] 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.

[8] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[9] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[10] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[11] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3018–3027, 2017.

[12] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7278–7286, 2018.

[13] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Rogerio Feris, Abhishek Kumar, Raja Giryes, and Alex M. Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. *NeurIPS*, 2018.

[14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[15] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[16] Yann N. Dauphin David Lopez-Paz Hongyi Zhang, Moustapha Cisse. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*, 2018.

[17] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447, 2019.

[18] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[19] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S ^{4} l: Self-supervised semi-supervised learning. *arXiv:1905.03670*, 2019.

[20] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NIPS*, 2014.

[21] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[22] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[23] T. Nathan Mundhenk, Daniel Ho, and Barry Y. Chen. Improvements to context based self-supervised learning. In *CVPR*, 2018.

[24] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *CVPR*, 2018.

[25] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.

[26] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Prez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. *arXiv preprint arXiv:1906.0518*, 2019.

[27] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. *CoRR*, abs/1711.06025, 2017.

[28] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. *CoRR*, abs/1904.03758, 2019.

[29] 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.

[30] Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *CoRR*, abs/1805.08136, 2018.

[31] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016.

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[33] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[35] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2051–2060, 2017.

[36] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[37] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811*, 2019.

[38] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.

[39] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015.

[40] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Appendix

5.1 Experimental Details

Datasets ***Mini*-ImageNet** consists of 100 classes from the ImageNet [32] which are split randomly into 64 base, 16 validation and 20 novel classes. Each class has 600 samples 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 50 classes respectively. **CIFAR-FS** is created by randomly splitting 100 classes of CIFAR-100 [33] into 64 base, 16 validation and 20 novel classes. The images are of size 32×32.

Implementation details Here, we also perform experiments on two additional model architectures : ResNet-18, ResNet-34 [34] apart from WRN-28-10 used in section 3.1. 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 optimizer for *mini*-ImageNet and CUB whereas SGD optimizer for CIFAR-FS. For WRN-28 training, we use Adam optimizer for all datasets.

Evaluation Criteria We evaluate experiments on 5-way 1-shot and 5-way 5-shot [9] 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 (from novel classes). For *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 [7] and DCO [28] in Section 3.1.

5.2 Performance over few-shot tasks by varying feature backbones

We compare the performance of Manifold Mixup [17] with Baseline++ [25] and mixup [16]. All experiments using Manifold Mixup randomly sample a hidden layer (including input layer) at each step to apply mixup as described in equation 1 for the mini-batch with mixup coefficient (λ) sampled from a $\beta(\alpha, \alpha)$ distribution with α = 2. For Mixup [16] the mixup coefficient is sampled from a uniform distribution ($\alpha = 1$).

The results are shown in table 4. 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.

Table 4 also reports the performance of using exemplar-training as self supervision task. Exemplar training [35] 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 , 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 [36] with soft margin on $f_\theta(\mathbf{x})$ on the mini-batch to bring the feature representation of positive examples close together. For this, we use random cropping, random horizontal/vertical flip and image jitter randomization [19] 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.

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.

5.3 Ablation Studies

We perform more ablations to show the efficacy of our proposed methodology. We choose *mini*-ImageNet as the primary dataset with WRN-28-10 backbone for the following experiments.

Generalization performance of supervised learning over base classes The results in table 4 and 2 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

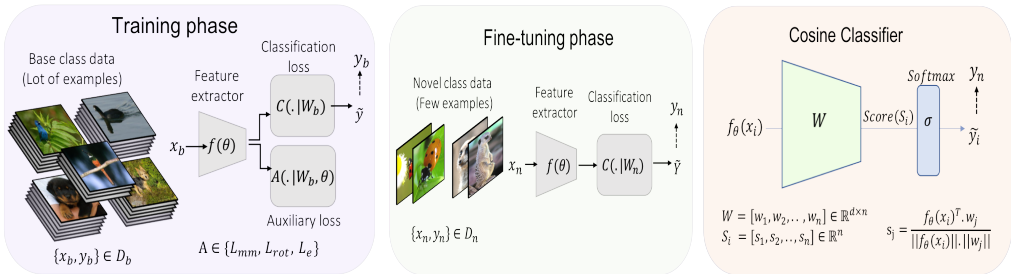

Figure 1: Flowchart for our proposed approach (*S2M2*) for few-shot learning. The auxiliary loss is derived from Manifold Mixup regularization and self-supervision tasks of rotation and exemplar.

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

Table 4: Results on *mini*-ImageNet, CUB and CIFAR-FS dataset over different network architecture.

Table 5: Validation set top-1 accuracy of different approaches over base classes and it's perturbed variants (I:ImageNet; I2:ImageNetv2; P:Pixelation noise; C: Contrast noise; B: Brightness; Adv: Adversarial noise)

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**
$S2M2_R$	**85.28**	**88.41**	**75.66**	**60.0**	**79.77**	28.0

Table 6: Effect of using the union of base and validation classes for training backbone feature extractor f_θ.

Method	Base + Validation	
	1-Shot	5-Shot
LEO [7]	61.76 ± 0.08	77.59 ± 0.12
DCO [28]	64.09 ± 0.62	80.00 ± 0.45
Baseline++	61.10 ± 0.19	75.23 ± 0.12
Manifold Mixup	61.10 ± 0.27	77.69 ± 0.21
Rotation	65.98 ± 0.36	81.67 ± 0.08
$S2M2_R$	$\mathbf{67.13 \pm 0.13}$	$\mathbf{83.6 \pm 0.34}$

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[37]. 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 [38]. For adversarial attack, we use the FGSM attack [39] 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 5.2. As it can be seen that our proposed technique 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 6, we see a considerable

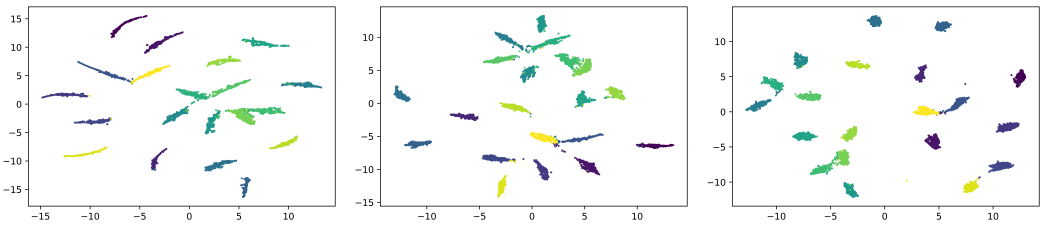

Figure 2: UMAP (2-dim) [40] plot for feature vectors of examples from novel classes of *mini*-Imagenet using Baseline++, Rotation, $S2M2_R$ (left to right).

improvement over the other approaches using the same extended data, supporting the generalizability claim of the proposed method.

Visualization of feature representations Fig 2 shows the 2-dimensional UMAP [40] 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.