# Few-shot Learning with Online Self-Distillation

Anonymous ICCV submission

Paper ID

## Abstract

*Few-shot learning has been a long-standing problem in learning to learn. This problem typically involves training a model on a extremely small amount of data and testing the model on the out-of-distribution data. The focus of recent few-shot learning research has been on the development of good representation models that can quickly adapt to test tasks. To that end, we come up with a model that learns representation through online self-distillation. Our model combines supervised training with knowledge distillation via a continuously updated teacher. We also identify that data augmentation plays an important role in producing robust features. Our final model is trained with CutMix augmentation and online self-distillation. On the commonly used benchmark miniImageNet, our model achieves 67.07% and 83.03% under the 5-way 1-shot setting and the 5-way 5-shot setting, respectively. It outperforms counterparts of its kind by 2.25% and 0.89%.*

## 1. Introduction

Few-shot learning is a crucial problem in learning to learn. In contrast to the common deep learning settings where a large amount of training data is available, few-shot learning often deals with scenarios where the training data is scarce. So this problem boils down to how to design models that can quickly adapt to test tasks. Recently, RFS [22] proposes a simple supervised-training baseline that outperforms meta-learning algorithms. It learns a representation model on the joint set of training tasks and improves the representations through self-distillation. The success of this method indicates that a good embedding is more important than sophisticated meta-learning algorithms.

However, RFS relies on a two-stage training pipeline consisting of supervised training and self-distillation, which reduces its practicability. To that end, we come up with a one-stage method that incorporates supervised training and knowledge distillation into a unified pipeline. The teacher network in our model is an exponential moving average of the student network and is continuously updated through the training process. The student network is trained with a combination of cross-entropy loss and self-distillation loss. Our model is significantly simpler than RFS [22] and other variants. In addition, we identify that CutMix [25] can greatly improve the representation model. Without bells and whistles, our model achieves 67.07% under the 5-way 1-shot setting and and 83.03% under the 5-way 5-shot setting on the miniImageNet [3] dataset.

## 2. Preliminary

We establish preliminaries of few-shot learning by learning representation [22] in this section. First, we formulate the problem in §2.1. Then, we present the details of RFS [22] in §2.2. For ease of comparison to previous work, we use the same notation as [22].

### 2.1. Few-shot Learning formulation

In few-shot learning, the data consists of o a meta-training set $\mathcal{T} = \{(\mathcal{D}_i^{train}, \mathcal{D}_i^{test})\}_{i=1}^{I}$ and a meta-testing set $\mathcal{S} = \{(\mathcal{D}_j^{train}, \mathcal{D}_j^{test})\}_{j=1}^{J}$. The meta-training set and the meta-testing set do not share the same categories. Each task $\mathcal{D}_i^{train}$ contains a small number of example. $\mathcal{D}^{train} = \{(\mathbf{x}_t, y_t)\}_{t=1}^{T}$ and $\mathcal{D}^{test} = \{(\mathbf{x}_q, y_q)\}_{q=1}^{Q}$ are sampled from the same distribution. A base learner $\mathcal{A}$, given by $y_* = f_\theta(\mathbf{x}_*)$ , is trained on $\mathcal{D}^{train}$ and evaluated on $\mathcal{D}^{test}$. To reduce the dimensionality of $\mathbf{x}_*$, training examples and testing examples are mapped into a feature space by an embedding model $\mathbf{\Phi}_* = f_\phi(\mathbf{x}_*)$. The objective of the few-shot learning algorithms is to learn a good embedding model, so that the average test error of the base learner on a distribution of tasks is minimized. This is given by,

$$\phi = \arg\min_{\phi} \mathbb{E}_{\mathcal{T}}[\mathcal{L}^{meta}(\mathcal{D}^{test}; \theta, \phi)], \tag{1}$$

where $\theta = \mathcal{A}(\mathcal{D}^{train}; \phi)$. Finally, the model is evaluated over the distribution of the test tasks:

$$\mathbb{E}_{\mathcal{S}}[\mathcal{L}^{meta}(\mathcal{D}^{test}; \theta, \phi), \text{where } \theta = \mathcal{A}(\mathcal{D}^{train}; \phi)]. \tag{2}$$

### 2.2. Few-shot Learning by learning the representation

RFS [22] proposes a simple method to learn the representation of the embedding model. Tasks from the meta-training set are merged into a single classification task, which is given by

$$\begin{aligned} \mathcal{D}^{new} &= \{(\mathbf{x}_i, y_i)\}_{k=1}^{K} \\ &= \cup\{\mathcal{D}_1^{train}, \ldots, \mathcal{D}_i^{train}, \ldots, \mathcal{D}_I^{train}\}, \end{aligned} \tag{3}$$

where $\mathcal{D}_i^{train}$ is the task from $\mathcal{T}$. The embedding model is

$$\phi = \arg\min_{\phi} \mathcal{L}^{ce}(\mathcal{D}^{new}; \phi), \tag{4}$$

and trained by optimizing a cross-entropy loss $\mathcal{L}^{ce}$.

In addition to this ordinary supervised training, RFS also introduces a self-distillation stage to further improve the representation. After obtaining the embedding model $\phi$, a new embedding model parameterized by $\phi'$ is trained to minimize a weighted sum of the cross-entropy loss between the predictions and ground-truth labels and the Kullback–Leibler divergence (KL) between predictions and soft targets:

$$\begin{aligned} \phi' = \arg\min_{\phi'} (\alpha \mathcal{L}^{ce}(\mathcal{D}^{new}; \phi') + \\ \beta KL(f(\mathcal{D}^{new}; \phi'), f(\mathcal{D}^{new}; \phi))). \end{aligned} \tag{5}$$

Conceptually, this step is a variant of self-distillation where the teacher network and the student network have the same model architectures.

Once the training is finished, the model is evaluated on the meta-testing set. The base learner is trained on the task $\mathcal{D}_j^{train}$ sampled from meta-testing distribution, given by

$$\theta = \arg\min_{\{\boldsymbol{W}, \boldsymbol{b}\}} \sum_{t=1}^{T} \mathcal{L}_t^{ce}(\boldsymbol{W} f_{\phi'}(\mathbf{x}_t) + \boldsymbol{b}, y_t) + \mathcal{R}(\boldsymbol{W}, \boldsymbol{b}), \tag{6}$$

where the base learner is parameterized by $\theta = \{\boldsymbol{W}, \boldsymbol{b}\}$ and the embedding model $f_{\phi'}$ is fixed.

## 3. Method

We will detail our method in this section. We introduce the online self-distillation in §3.1. Then, we discuss one special data augmentation technique – CutMix [25] in §3.2.

### 3.1. Few-Shot Learning with Online Self-Distillation

We propose a training pipeline that combines supervised training with self-distillation, in contrast to existing methods that consist of separate stages. We use $\phi$ and $\phi'$ to denote the teacher network and the student network, respectively. Instead of learning $\phi$ in a pre-training stage, our method updates $\phi$ on-the-fly as well as distilling the knowledge from $\phi$ to $\phi'$ (Figure 1). Mathematically, we alternate between these two steps:

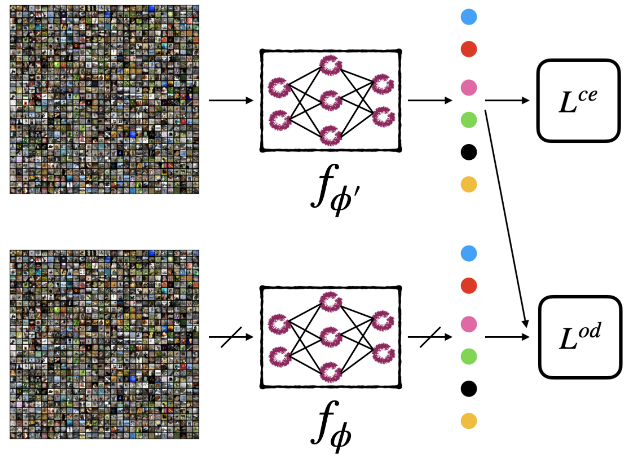


Figure 1: Overview of online self-distillation. Back-propagation and SGD are not performed in the $f_\phi$ branch.

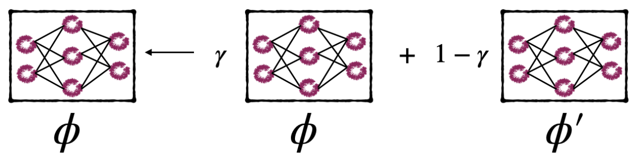


Figure 2: Update rules of the teacher network $f_\phi$.

$$\begin{aligned} \phi' = \arg\min_{\phi'} (\alpha \mathcal{L}^{ce}(\mathcal{D}^{new}; \phi') + \\ \beta KL(f(\mathcal{D}^{new}; \phi'), f(\mathcal{D}^{new}; \phi))), \end{aligned} \tag{7}$$

and

$$\phi = \gamma\phi + (1-\gamma)\phi' \tag{8}$$

where $\gamma = 0.99$ controls the velocity of the parameter update. Different from common machine learning models, $\phi$ is not updated through gradient descent but direct parameter update.

### 3.2. CutMix

We present a special data augmentation–CutMix [25]–that improves few-shot learning performance. The goal of CuxMix is to generate a new training example $(\bar{\mathbf{x}}, \bar{y})$ by combining examples $(\mathbf{x}_a, y_a)$ and $(\mathbf{x}_b, y_b)$, given by

$$\begin{aligned} \bar{\mathbf{x}} &= \mathcal{M} \odot \mathbf{x}_a + (1-\mathcal{M}) \odot \mathbf{x}_b \\ \bar{y} &= m y_a + (1-m) y_b, \end{aligned} \tag{9}$$

| model | backbone | miniImageNet 5-way 1-shot | miniImageNet 5-way 5-shot |
|---|---|---|---|
| MAML [6] | 32-32-32-32 | 48.70 ± 1.84 | 63.11 ± 0.92 |
| Matching Networks [23] | 64-64-64-64 | 43.56 ± 0.84 | 55.31 ± 0.73 |
| IMP [1] | 64-64-64-64 | 49.2 ± 0.7 | 64.7 ± 0.7 |
| Prototypical Networks† [19] | 64-64-64-64 | 49.42 ± 0.78 | 68.20 ± 0.66 |
| TAML [9] | 64-64-64-64 | 51.77 ± 1.86 | 66.05 ± 0.85 |
| SAML [8] | 64-64-64-64 | 52.22 ± n/a | 66.49 ± n/a |
| GCR [11] | 64-64-64-64 | 53.21 ± 0.80 | 72.34 ± 0.64 |
| KTN(Visual) [15] | 64-64-64-64 | 54.61 ± 0.80 | 71.21 ± 0.66 |
| PARN[24] | 64-64-64-64 | 55.22 ± 0.84 | 71.55 ± 0.66 |
| Dynamic Few-shot [7] | 64-64-128-128 | 56.20 ± 0.86 | 73.00 ± 0.64 |
| Relation Networks [21] | 64-96-128-256 | 50.44 ± 0.82 | 65.32 ± 0.70 |
| R2D2 [2] | 96-192-384-512 | 51.2 ± 0.6 | 68.8 ± 0.1 |
| SNAIL [12] | ResNet-12 | 55.71 ± 0.99 | 68.88 ± 0.92 |
| AdaResNet [13] | ResNet-12 | 56.88 ± 0.62 | 71.94 ± 0.57 |
| TADAM [14] | ResNet-12 | 58.50 ± 0.30 | 76.70 ± 0.30 |
| Shot-Free [17] | ResNet-12 | 59.04 ± n/a | 77.64 ± n/a |
| TEWAM [16] | ResNet-12 | 60.07 ± n/a | 75.90 ± n/a |
| MTL [20] | ResNet-12 | 61.20 ± 1.80 | 75.50 ± 0.80 |
| Variational FSL [26] | ResNet-12 | 61.23 ± 0.26 | 77.69 ± 0.17 |
| MetaOptNet [10] | ResNet-12 | 62.64 ± 0.61 | 78.63 ± 0.46 |
| Diversity w/ Cooperation [5] | ResNet-18 | 59.48 ± 0.65 | 75.62 ± 0.48 |
| Fine-tuning [4] | WRN-28-10 | 57.73 ± 0.62 | 78.17 ± 0.49 |
| LEO-trainval† [18] | WRN-28-10 | 61.76 ± 0.08 | 77.59 ± 0.12 |
| RFS-simple | ResNet-12 | 62.02 ± 0.63 | 79.64 ± 0.44 |
| RFS-distill | ResNet-12 | 64.82 ± 0.60 | 82.14 ± 0.43 |
| Ours-online-distill (w/o CutMix) | ResNet-12 | 64.33 ± 0.25 | 82.13 ± 0.17 |
| Ours-online-distill | ResNet-12 | **67.07 ± 0.26** | **83.03 ± 0.18** |
| Ours-online-distill-trainval † | ResNet-12 | **68.96 ± 0.26** | **84.22 ± 0.17** |

Table 1: Comparison to prior work on miniImageNet. Results reported with input image size of 84x84. † results obtained by training on the union of training and validation sets.

where $\mathcal{M}$ is a binary mask and $\odot$ is element-wise multiplication. $m \in [0, 1]$ is sampled from a beta distribution. To generate the binary mask $\mathcal{M}$, we sample the bounding box $\mathbf{B} = (r_x, r_y, r_w, r_h)$ where

$$\begin{aligned} r_x &\sim Uniform(0, W), \qquad & r_w &= W\sqrt{1-m} \\ r_y &\sim Uniform(0, H), \qquad & r_h &= H\sqrt{1-m}. \end{aligned} \tag{10}$$

The binary mask is produced by filling 0 within the bounding box $\mathbf{B}$, otherwise 1.

## 4. Experiment

**Dataset.** We conduct experiments on the widely used benchmarks miniImageNet, CIFAR-FS, and FC100. miniImageNet is a subset of ImageNet; it contains 64, 16, 20 categories for training, validation, and testing, respectively. The CIFAR-FS and FC100 are both derivatives of the CIFAR-100 dataset. CIFAR-FS has 64, 16, 20 categories for training, validation, and testing while FC100 has 60, 20, 20 categories for training, validation, and testing.

**Model.** We use the same ResNet12 with MetaOptNet [10] and RFS [22]. This ResNet contains four blocks, where each block consists of three 3x3 convolutional kernels and one 2x2 max pooling layer. A global average pooling layer

| model | backbone | CIFAR-FS 5-way 1-shot | CIFAR-FS 5-way 5-shot | FC100 5-way 1-shot | FC100 5-way 5-shot |
|---|---|---|---|---|---|
| MAML [6] | 32-32-32-32 | 58.9 ± 1.9 | 71.5 ± 1.0 | - | - |
| Prototypical Networks [19] | 64-64-64-64 | 55.5 ± 0.7 | 72.0 ± 0.6 | 35.3 ± 0.6 | 48.6 ± 0.6 |
| Relation Networks [21] | 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 [14] | ResNet-12 | - | - | 40.1 ± 0.4 | 56.1 ± 0.4 |
| Shot-Free [17] | ResNet-12 | 69.2 ± n/a | 84.7 ± n/a | - | - |
| TEWAM [16] | ResNet-12 | 70.4 ± n/a | 81.3 ± n/a | - | - |
| Prototypical Networks [19] | ResNet-12 | 72.2 ± 0.7 | 83.5 ± 0.5 | 37.5 ± 0.6 | 52.5 ± 0.6 |
| MetaOptNet [10] | ResNet-12 | 72.6 ± 0.7 | 84.3 ± 0.5 | 41.1 ± 0.6 | 55.5 ± 0.6 |
| RFS-simple | ResNet-12 | 71.5 ± 0.8 | 86.0 ± 0.5 | 42.6 ± 0.7 | 59.1 ± 0.6 |
| RFS-distill | ResNet-12 | 73.9 ± 0.8 | 86.9 ± 0.5 | 44.6 ± 0.7 | 60.9 ± 0.6 |
| Ours-online-distill | ResNet-12 | **76.18 ± 0.21** | **87.1 ± 0.2** | **45.43 ± 0.24** | **61.7 ± 0.3** |

Table 2: Comparison to prior work on CIFAR-FS and FC100.

is included at the end of the model to produce global features. The number of filters in each block is (64, 160, 320, 480). We use $\alpha = \beta = 0.5$ to balance the weights of the cross-entropy loss and the knowledge distillation loss. For other hyperparameters including batch size, learning rate and etc, we use the same configuration with RFS. The model is trained totally for 200 epochs. We use CutMix augmentation with $m$ sampled from $\mathrm{Beta}(0.2, 0.2)$.

**Results.** As shwon in Table 1 and Table 2, our method with CutMix achieves stage-of-the-art performance on all settings; this indicates the effectiveness of incorporating online self-distillation and CutMix. Without CutMix, our method outperforms RFS (w/o distillation, one stage) and is comparable to RFS (w/ distillation, two stage) while our method only uses one-stage training. In addition, our method uses the same evaluation protocol and does not introduce any further computational overhead.

## 5. Conclusion

We propose a one-stage online self-distillation pipeline for few-shot learning. Our method relies on distilling knowledge from a momentum-updated teacher to a student. Our method suggests that multi-stage self-distillation is not imperative. We also identify that CutMix significantly improves the representation. With these combined techniques, our method achieves new state-of-the-art on the commonly used datasets. We hope our method will shed new lights into the few-shot learning research.

## References

[1] Kelsey Allen, Evan Shelhamer, Hanul Shin, and Joshua Tenenbaum. Infinite mixture prototypes for few-shot learning. In *ICML*, 2019. 3

[2] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018. 3

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 1

[4] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *ICLR*, 2020. 3

[5] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *ICCV*, 2019. 3

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 3

[7] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 3

[8] Fusheng Hao, Fengxiang He, Jun Cheng, Lei Wang, Jianzhong Cao, and Dacheng Tao. Collect and select: Semantic alignment metric learning for few-shot learning. In *ICCV*, 2019. 3

[9] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *CVPR*, 2019. 3

[10] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019. 3

[11] Aoxue Li, Tiange Luo, Tao Xiang, Weiran Huang, and Liwei Wang. Few-shot learning with global class representations. In *ICCV*, 2019. 3

[12] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017. 3

[13] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. *arXiv preprint arXiv:1712.09926*, 2017. 3

[14] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NIPS*, 2018. 3

[15] Zhimao Peng, Zechao Li, Junge Zhang, Yan Li, Guo-Jun Qi, and Jinhui Tang. Few-shot image recognition with knowledge transfer. In *ICCV*, 2019. 3

[16] Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *ICCV*, 2019. 3

[17] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *ICCV*, 2019. 3

[18] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019. 3

[19] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 3

[20] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, 2019. 3

[21] 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 *CVPR*, 2018. 3

[22] 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. 1, 2, 3

[23] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NIPS*, 2016. 3

[24] Ziyang Wu, Yuwei Li, Lihua Guo, and Kui Jia. Parn: Position-aware relation networks for few-shot learning. In *ICCV*, 2019. 3

[25] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[26] Jian Zhang, Chenglong Zhao, Bingbing Ni, Minghao Xu, and Xiaokang Yang. Variational few-shot learning. In *ICCV*, 2019. 3