HAL
open science

Diversity with Cooperation: Ensemble Methods for Few-Shot Classification

Nikita Dvornik, Cordelia Schmid, Julien Mairal

To cite this version:

Nikita Dvornik, Cordelia Schmid, Julien Mairal. Diversity with Cooperation: Ensemble Methods for Few-Shot Classification. 2019. hal-02080004v1

HAL Id: hal-02080004
https://hal.science/hal-02080004v1

Preprint submitted on 26 Mar 2019 (v1), last revised 8 Aug 2019 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Diversity with Cooperation: Ensemble Methods for Few-Shot Classification

Nikita Dvornik, Cordelia Schmid, Julien Mairal

Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France

firstname.lastname@inria.fr

Abstract

Few-shot classification consists of learning a predictive model that is able to effectively adapt to a new class, given only a few annotated samples. To solve this challenging problem, meta-learning has become a popular paradigm that advocates the ability to "learn to adapt". Recent works have shown, however, that simple learning strategies without meta-learning could be competitive. In this paper, we go a step further and show that by addressing the fundamental high-variance issue of few-shot learning classifiers, it is possible to significantly outperform current meta-learning techniques. Our approach consists of designing an ensemble of deep networks to leverage the variance of the classifiers, and introducing new strategies to encourage the networks to cooperate, while encouraging prediction diversity. Evaluation is conducted on the mini-ImageNet and CUB datasets, where we show that even a single network obtained by distillation yields state-of-the-art results.

1 Introduction

Convolutional neural networks [17] have become standard tools in computer vision to model images, leading to outstanding results in many tasks such as classification [16], object detection [8, 19, 25], or semantic segmentation [8, 20, 26]. Massively annotated datasets such as ImageNet [27] or COCO [18] seem to have played a key role in this success. However, annotating a large corpus is expensive and not always feasible, depending on the task at hand. Improving the generalization capabilities of deep neural networks and removing the need for huge sets of annotations is thus of utmost importance.

While such a grand challenge may be addressed from different complementary points of views, e.g., large-scale unsupervised learning [4], self-supervised learning [7, 12], or by developing regularization techniques dedicated to deep networks [2, 35], we choose in this paper to focus on variance-reduction principles based on ensemble methods.

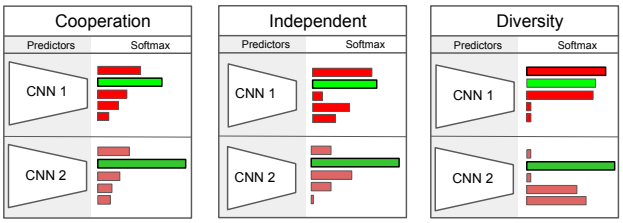


Figure 1: **Illustration of the cooperation and diversity strategies on two networks.** All networks receive the same image as input and compute corresponding class probabilities with softmax. Cooperation encourages the non-ground truth probabilities (in red) to be similar, after normalization, whereas diversity encourages orthogonality.

Specifically, we are interested in few-shot classification, where a classifier is first trained from scratch on a medium-sized annotated corpus—that is, without leveraging external data or a pre-trained network, and then we evaluate its ability to adapt to new classes, for which only very few annotated samples are provided (typically 1 or 5). Unfortunately, simply fine-tuning a convolutional neural network on a new classification task with very few samples has been shown to provide poor results [9], which has motivated the community to develop dedicated approaches.

The dominant paradigm in few-shot learning builds upon meta-learning [9, 24, 29, 31, 30, 32], which is formulated as a principle to learn how to adapt to new learning problems. These approaches split a large annotated corpus into classification tasks, and the goal is to transfer knowledge across tasks in order to improve generalization. While the meta-learning principle seems appealing for few-shot learning, its empirical benefits have not been clearly established yet. There is indeed strong evidence [5, 11, 23] that train-

ing CNNs from scratch using meta-learning performs substantially worse than if CNN features are trained in a standard fashion—that is, by minimizing a classical loss function relying on corpus annotations; on the other hand, learning only the last layer with meta-learning has been found to produce better results [11, 23]. Then, it was recently shown in [5] that simple distance-based classifiers could achieve similar accuracy as meta-learning approaches.

Our paper goes a step further and shows that meta-learning-free approaches can be improved and significantly outperform the current state of the art in few-shot learning. Our angle of attack consists of using ensemble methods to reduce the variance of few-shot learning classifiers, which is inevitably high given the small number of annotations. Given an initial medium-sized dataset (following the standard setting of few-shot learning), the most basic ensemble approach consists of first training several CNNs independently before freezing them and removing the last prediction layer. Then, given a new class (with few annotated samples), we build a mean centroid classifier for each network and estimate class probabilities—according to a basic probabilistic model—of test samples based on the distance to the centroids [21, 30]. The obtained probabilities are then averaged over networks, resulting in higher accuracy.

While we show that the basic ensemble method where networks are trained independently already performs well, we introduce penalty terms that allow the networks to cooperate during training, while encouraging enough diversity of predictions, as illustrated in Figure 1. The motivation for cooperation is that of easier learning and regularization, where individual networks from the ensemble can benefit from each other. The motivation for encouraging diversity is classical for ensemble methods [6], where a collection of weak learners making diverse predictions often performs better together than a single strong one. Whereas these two principles seem in contradiction with each other at first sight, we show that both principles are in fact useful and lead to significantly better results than the basic ensemble method. Finally, we also show that a single network trained by distillation [14] to mimic the behavior of the ensemble also performs very well, which brings a significant speed-up at test time. In summary, our contributions are three-fold:

- We introduce mechanisms to encourage cooperation and diversity for learning an ensemble of networks. We study these two principles for few-shot learning and characterize the regimes where they are useful.
- We show that it is possible to significantly outperform current state-of-the-art techniques for few-shot classification without using meta-learning.
- As a minor contribution, we also show how to distill the knowledge from an ensemble into a single network with minor loss in accuracy on novel categories, by using additional unlabeled data.

The paper is organized as follows: Section 2 discusses related work; Section 3 introduces our approach; Section 4 is devoted to experiments and Section 5 concludes the paper.

2 Related Work

In this section, we discuss related work on few-shot learning, meta-learning, and ensemble methods.

Few-shot classification. Typical few-shot classification problems consist of two parts called meta-training and meta-testing [5]. During the meta-training stage, one is given a large-enough annotated dataset, which is used to train a predictive model. During meta-testing, novel categories are provided along with few annotated examples, and we evaluate the capacity of the predictive model to retrain or adapt, and then generalize on these new classes.

Meta-learning approaches typically sample few-shot learning classification tasks from the meta-training dataset, and train a model such that it should generalize on a new task that has been left aside. For instance, in [9] a "good network initialization" is learned such that a small number of gradient steps on a new problem is sufficient to obtain a good solution. In [24], the authors learn both the network initialization and an update rule (optimization model) represented by a Long-Term-Short-Memory network (LSTM). Inspired by few-shot learning strategies developed before deep learning approaches became popular [21], distance-based classifiers based on the distance to a centroid were also proposed, *e.g.*, prototypical networks [30], or more sophisticated classifiers with attention [32]. All these methods consider a classical backbone network, and train it from scratch using meta-learning.

Recently, these meta-learning were found to be sub-optimal [11, 22, 23]. Specifically, better results were obtained by training the network on the classical classification task using the meta-training data in a first step, and then only fine-tuning with meta-learning in a second step [22, 28, 34]. Others such as [11, 23] simply freeze the network obtained in the first step, and train a simple prediction layer with meta-learning, which results in similar performance. Finally, the paper [5] demonstrates that simple baselines without meta-learning—based on distance-based classifiers—work equally well. Our paper pushes such principles even further and shows that by appropriate variance-reduction techniques, these approaches can significantly outperform the current state of the art.

Ensemble methods. It is well known that ensemble methods reduce the variance of estimators and subsequently may

improve the quality of prediction [10]. To gain accuracy from averaging, various randomization or data augmentation techniques are typically used to encourage a high diversity of predictions [3, 6]. While individual classifiers of the ensemble may perform poorly, the quality of the average prediction turns out to be sometimes surprisingly high.

Even though ensemble methods are costly at training time for neural networks, it was shown that a single network trained to mimic the behavior of the ensemble could perform almost equally well [14]—a procedure called distillation—thus removing the overhead at test time. To improve the scalability of distillation in the context of highly-parallelized implementations, an online distillation procedure is proposed in [1]. There, each network is encouraged to agree with the averaged predictions made by other networks of the ensemble, which results in more stable models. The objective of our work is however significantly different. The form of cooperation they encourage between networks is indeed targeted to scalability and stability (due to industrial constraints), but online distilled networks do not necessarily perform better than the basic ensemble strategy. Our goal, on the other hand, is to improve the quality of prediction and do better than basic ensembles.

To this end, we encourage cooperation in a different manner, by encouraging predictions between networks to match in terms of class probabilities conditioned on the prediction not being the ground truth label. While we show that such a strategy alone is useful in general when the number of networks is small, encouraging diversity becomes crucial when this number grows. Finally, we show that distillation can help to reduce the computational overhead at test time.

3 Our Approach

In this section, we present our approach for few-shot classification, starting with preliminary components.

3.1 Preliminaries about mean-centroid classifiers

We now explain how to perform few-shot classification with a fixed feature extractor and a mean centroid classifier.

Few-shot classification with prototype classifier. During the meta-training stage, we are given a dataset D_b with annotations, which we use to train a prediction function f_θ represented by a CNN. Formally, after training the CNN on D_b, we remove the final prediction layer and use the resulting vector $\tilde{f}_\theta(x)$ as a set of visual features for a given image x. The parameters θ represent the weights of the network, which are frozen after this training step.

During meta-testing, we are given a new dataset $D_q = \{x_i, y_i\}_{i=1}^{nk}$, where n is a number of new categories and k is the number of available examples for each class. The (x_i, y_i)'s represent image-label pairs. Then, we build a mean centroid classifier, leading to the class prototypes

$$c_j = \frac{1}{k} \sum_{i=1}^{k} \tilde{f}_\theta(x_i), \qquad j = 1, ..., n. \tag{1}$$

Finally, a test sample x is assigned to the nearest centroid's class. Simple mean-centroid classifiers have proven to be effective in the context of few-shot classification [5, 21, 30], which is confirmed in the following experiment.

Motivation for mean-centroid classifier. We report here an experiment showing that a more complex model than (1) does not necessarily lead to better results for few-shot learning. Consider indeed a parametrized version of (1):

$$c_j = \sum_{i=1}^{nk} \alpha_i^j \tilde{f}_\theta(x_i), \qquad j = 1, ..., n, \tag{2}$$

where the weights α_i^j can be learned with gradient descent by maximizing the likelihood of the probabilistic model

$$p_j(y = l|x) = \frac{\exp(-d(\tilde{f}_\theta(x), c_l))}{\sum_{j=1}^{n} \exp(-d(\tilde{f}_\theta(x), c_j)} \tag{3}$$

where $d(\cdot, \cdot)$ is a distance function, such as Euclidian distance or negative cosine similarity. Since the coefficients are learned from data and not set arbitrarily to $1/k$ as in (1), one would potentially expect this method to produce better classifiers if appropriately regularized. When we run the evaluation of the aforementioned classifiers on 1000 5-shot learning tasks sampled from `miniImagenet-test` (see experimental section for details about this dataset), we get similar results on average: $77.28 \pm 0.46\%$ for (1) vs. $77.01 \pm 0.50\%$ for (2), confirming that learning meaningful parameters in this very-low-sample regime is difficult.

3.2 Learning ensembles of deep networks

During meta-training, one needs to minimize the following loss function over a training set $\{x_i, y_i\}_{i=1}^{m}$:

$$L(\theta) = \frac{1}{m} \sum_{i=1}^{m} \ell(y_i, \sigma(f_\theta(x_i))) + \lambda \|\theta\|_2^2, \tag{4}$$

where f_θ is a CNN as before. The cost function $\ell(\cdot, \cdot)$ is the cross-entropy between ground-truth labels and predicted class probabilities $p = \sigma(f_\theta(x))$, where σ is the normalized exponential function, and λ is a weight decay parameter.

When training an ensemble of K networks f_{θ_k} independently, one would solve (4) for each network separately. While these terms may look identical, solutions provided

by deep neural networks will typically differ when trained with different initializations and random seeds, making ensemble methods appealing in this context.

In this paper, we are interested in ensemble of networks, but we also want to model relationships between its members; this may be achieved by considering a pairwise penalty function ψ, leading to the joint formulation:

$$L(\bar{\theta}) = \sum_{j=1}^{K} \left(\frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \sigma(f_{\theta_j}(x_i))) + \lambda \|\theta_j\|_2^2 \right) + \frac{\gamma}{n(K-1)} \sum_{i=1}^{n} \sum_{\substack{j,l \\ j \neq l}}^{K} \psi(y_i, f_{\theta_j}(x_i), f_{\theta_l}(x_i)), \quad (5)$$

where $\bar{\theta}$ is the vector obtained by concatenating all the parameters θ_j. By carefully designing the function ψ and setting up appropriately the parameter γ, it is possible to achieve desirable properties of the ensemble, such as diversity of predictions or collaboration during training.

3.3 Ensembles with diversity and cooperation

To reduce the high variance of few-shot learning classifiers, we use ensemble methods trained with a particular interaction function ψ, as in (5). Then, once the parameters θ_j have been learned during meta-training, classification in meta-testing is performed by considering a collection of K mean-centroid classifiers associated to the basic probabilistic model presented in Eq. (3). Given a test image, the K class probabilities are averaged. Such a strategy was found to perform empirically better than a voting scheme.

As we show in the experimental section, the choice of pairwise relationship function ψ significantly influences the quality of the ensemble. Here, we describe three different strategies, which all provide benefits in different regimes, starting by a criterion encouraging diversity of predictions.

Diversity. One way to encourage diversity consists of introducing randomization in the learning procedure, *e.g.*, by using data augmentation [3, 10] or various initializations. Here, we also evaluate the effect of an interaction function ψ that acts directly on the network predictions. Given an image x, two models parametrized by θ_i and θ_j respectively lead to class probabilities $p_i = \sigma(f_{\theta_i}(x))$ and $p_j = \sigma(f_{\theta_j}(x))$. During training, p_i and p_j are naturally encouraged to be close to the assignment vector e_y in $\{0,1\}^d$ with a single non-zero entry at position y, where y is the class label associated to x and d is the number of classes.

From [14], we know that even though only the largest entry of p_i or p_j is used to make predictions, other entries—typically not corresponding to the ground truth label y—carry important information about the network. It becomes then natural to consider the probabilities $\hat{p}_i$ and $\hat{p}_j$ conditioned on not being the ground truth label. Formally, these are obtained by setting to zero the entry y in p_i and p_j renormalizing the corresponding vectors such that they sum to one. Then, we consider the following diversity penalty

$$\phi(\hat{p}_i, \hat{p}_j) = \cos(\hat{p}_i, \hat{p}_j). \quad (6)$$

When combined with the loss function, the resulting formulation encourages the networks to make the right prediction according to the ground-truth label, but then they are also encouraged to make different second-best, third-best, and so on, choice predictions (see Figure 1). This penalty turns out to be particularly effective when the number of networks is large, as shown in the experimental section. It typically worsens the performance of individual classifiers on average, but make the ensemble prediction more accurate.

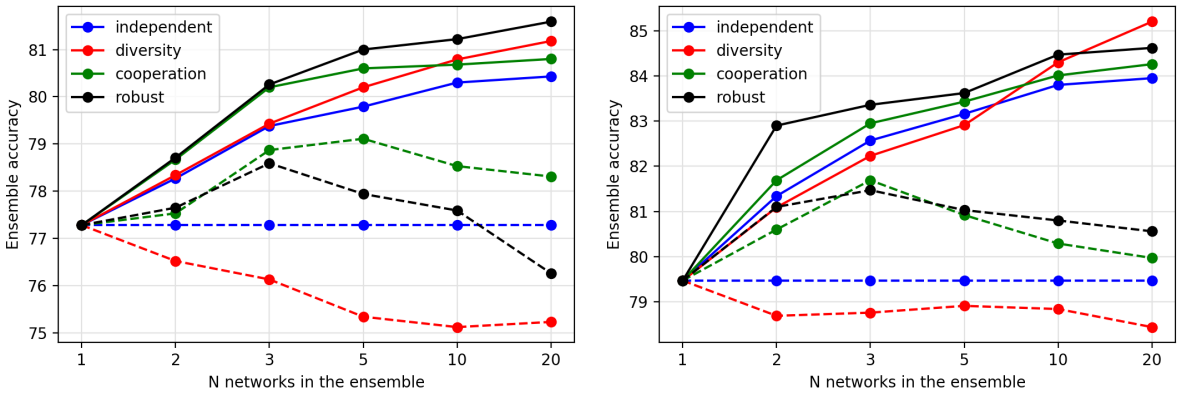
Cooperation. Apparently opposite to the previous principle, encouraging the conditional probabilities $\hat{p}_i$ to be similar—though with a different metric—may also improve the quality of prediction by allowing the networks to cooperate for better learning. Our experiments show that such a principle alone may be effective, but it appears to be mostly useful when the number of training networks is small, which suggests that there is a trade-off between cooperation and diversity that needs to be found.

Specifically, our experiments show that using the negative cosine—in other words, the opposite of (6)—is ineffective. However, a penalty such as the symmetrized KL-divergence turned out to provide the desired effect:

$$\phi(\hat{p}_i, \hat{p}_j) = \frac{1}{2}(\mathrm{KL}(\hat{p}_i || \hat{p}_j) + \mathrm{KL}(\hat{p}_j || \hat{p}_i)). \quad (7)$$

By using this penalty, we managed to obtain more stable and faster training, resulting in better performing individual networks, but also—perhaps surprisingly—a better ensemble. Unfortunately, we also observed that the gain of ensembling diminishes with the number of networks in the ensemble since the individual members become too similar.

Robustness and cooperation. Given experiments conducted with the two previous penalties, a trade-off between cooperation and diversity seems to correspond to two regimes (low vs. high number of networks). This motivated us to develop an approach designed to achieve the best trade-off. When considering the cooperation penalty (7), we try to increase diversity of prediction by several additional means. i) We randomly drop some networks from the ensemble at each training iteration, which causes the networks to learn on different data streams and reduces the speed of knowledge propagation. ii) We introduce Dropout within



(a) *Mini*ImageNet 5-shots

(b) CUB 5-shots

Figure 2: **Accuracies of different ensemble strategies (one for each color) for various numbers of networks.** Solid lines give the ensemble accuracy after aggregating predictions. The average performance of single models from the ensemble is plotted with a dashed line. Best viewed in color.

each network to increase randomization. iii) We feed each network with a different (crop, color) transformation of the same image, which makes the ensemble more robust to input image transformations. Overall, this strategy was found to perform best in most scenarios (see Figure 2).

3.4 Ensemble distillation

As most ensemble methods, our ensemble strategy introduces a significant computional overhead at training time. To remove the overhead at test time, we use a variant of knowledge distillation [14] to compress the ensemble into a single network f_w. Given the meta-training dataset D_b, we consider the following cost function on example (x, y):

$$\ell(x, y) = (1-\alpha) \cdot \hat{\ell}(e_y, \sigma(f_w(x))) - \alpha \cdot T^2 \cdot \hat{\ell}\left(\frac{1}{K}\sum_{k=1}^{K} \sigma\left(\frac{f_{\theta_k}(x)}{T}\right), \sigma\left(\frac{f_w(x)}{T}\right)\right), \quad (8)$$

where, $\hat{\ell}$ is cross-entropy, e_y is a one-hot embedding of the true label y. The second term performs distillation with parameter T (see [14]). It encourages the single model f_w to be similar to the average output of the ensemble. In our experiments, we are able to obtain a model with performance relatively close to that of the ensemble (see Section 4).

Modeling out-of-distribution behavior. When distillation is performed on the dataset D_b, the network f_w mimics the behavior of the ensemble on a specific data distribution. However, new categories are introduced at test time. Therefore, we also tried distillation by using additional unnannotated data, which yields slightly better performance.

4 Experiments

We now present experiments to study the effect of cooperation and diversity for ensemble methods, and start with experimental and implementation details.

4.1 Experimental Setup

Datasets. We use *Mini*ImageNet [24] which is derived from the original ImageNet [27] dataset and Caltech-UCSD Birds (CUB) 200-2011 [33]. *Mini*ImageNet consists of 100 categories—64 for training, 16 for validation and 20 for testing—with 600 images each. The CUB dataset consists of 11,788 images of birds of more than 200 species. We adopt train, val, and test splits from [34], which were originally created by randomly splitting all 200 species in 100 for training, 50 for validation, and 50 for testing.

Evaluation. In few-shot classification, the test set is used to sample N 5-way classification problems, where only k examples of each category are provided for training and 15 for evaluation. We follow [9, 11, 22, 23, 24] and test our algorithms for $k = 1$ and 5 and N is set to 1 000. Each time, classes and corresponding train/test examples are sampled at random. For all our experiments we report the mean accuracy (in %) over 1 000 tasks and 95% confidence interval.

Implementation details. For all experiments, we use the Adam optimizer [15] with an initial learning rate 10^{-4}, which is decreased by a factor 10 once during training when no improvement in validation accuracy is observed for p consecutive epochs. For *Mini*ImageNet, we use $p = 10$, and 20 for the CUB dataset. When distilling an ensemble into one network, p is doubled. We use random crops and color augmentation during training as well as weight decay

with parameter $\lambda = 5 \cdot 10^{-4}$. All experiments are conducted with the ResNet18 architecture [13], which allows us to train our ensembles of 20 networks on a single GPU. Input images are then re-scaled to the size 224×224, and organized in mini-batches of size 16. Validation accuracy is computed by running 5-shot evaluation on the validation set. During the meta-testing stage, we take central crops of size 224×224 from images and feed them to the feature extractor. No other preprocessing is used at test time. When building a mean centroid classifier, the distance d in (3) is computed as the negative cosine similarity [30], which is rescaled by a factor 10. For reproducibility purposes, our implementation will be made publicly available.

4.2 Ensembles for Few-Shot Classification

In this section, we study the effect of ensemble training with pairwise interaction terms that encourage cooperation or diversity. For that purpose, we analyze the link between the size of ensembles and their 1- and 5-shot classification performance on the *Mini*ImageNet and CUB datasets.

Details about the three strategies. When models are trained jointly, the data stream is shared across all networks and weight updates happen simultaneously. This is achieved by placing all models on the same GPU and optimizing the loss (5). When training a diverse ensemble, we use the cosine function (6) and selected the parameter $\gamma = 1$ that performed best on the validation set among the tested values (10^i, for $i = -2, \ldots, 2$) for $n = 5$ and $n = 10$ networks. Then, this value was kept for other values of n. To enforce cooperation between networks, we use the symmetrized KL function (7) and selected the parameter $\gamma = 10$ in the same manner. Finally, the robust ensemble strategy is trained with the cooperation relationship penalty and the same parameter γ, but we use Dropout with probability 0.1 before the last layer; each of the network is dropped from the ensemble with probability 0.2 at every iteration; different networks receive different transformation of the same image, i.e. different random crops and color augmentation.

Results. Tables 1 and 2 summarize the few-shot classification accuracies of ensembles trained with our strategies and compare with basic ensembles. On the *Mini*ImageNet dataset, the results for 1- and 5-shot classification are consistent with each other. Training with cooperation allows smaller ensembles ($n \leq 5$) to perform better, which leads to higher individual accuracy of the networks from the ensemble, as seen in Figure 2. However, when $n \geq 10$, cooperation is less effective, as opposed to the diversity strategy, which benefits from larger n. As we can see from Figure 2, individual members of the ensemble become worse, but the ensemble accuracy improves substantially. Finally, the robust strategy seems to perform best for all values of n in almost all settings. The situation for the CUB dataset is similar, although we notice that robust ensembles perform similarly as the diversity strategy for $n = 20$.

4.3 Distilling an ensemble into a single network

We distill robust ensembles of all sizes to study knowledge transferability with growing ensemble size. To do so, we use the meta-training dataset and optimize the loss (8) with parameters $T = 10$ and $\alpha = 0.8$. For the strategy using external data, we randomly add at each iteration 8 images (without annotations) from the COCO [18] dataset to the 16 annotated samples from the meta-training data. Those images contribute only to the distillation part of the loss (8). Tables 1, 2 display model accuracies for both *Mini*ImageNet and CUB datasets. For 5-shot classification on *Mini*ImageNet, the difference between ensemble and its distilled version is rather low (around 1%), while adding extra non-annotated data helps reducing this gap. Surprisingly, 1-shot classification accuracy is slightly higher for distilled models than for their corresponding full ensembles. On the CUB dataset, distilled models stop improving after $n = 5$, even though the performance of full ensembles keeps growing. This seems to indicate that the capacity of the single network may have been reached, which suggests using a more complex architecture here. Consistently with such hypothesis, adding extra data is not as helpful as for *Mini*ImageNet, most likely because data distributions of COCO and CUB are more different.

In Table 3, we also compare the performance of our distilled networks with other baselines from the literature, including current state-of-the-art meta-learning approaches, showing that our approach does significantly better on the *Mini*ImageNet dataset.

4.4 Study of relationship penalties

There are many possible ways to model relationship between the members of an ensemble. In this subsection, we study and discuss such particular choices.

Input to relationship function. As noted by [14], class probabilities obtained by the softmax layer of a network seem to carry a lot of information and are useful for distillation. However, after meta-training, such probabilities are often close to binary vectors with a dominant value associated to the ground-truth label. To make small values more noticeable, distillation uses a parameter T, as in (8). Given such a class probability computed by a network, we experimented such a strategy consisting of introducing new probabilities $\hat{p} = \sigma(p/T)$, where the contributions of non ground-truth values are emphasized. When used within our diver-

5-shot						
Ensemble type	1	2	3	5	10	20
Independent	77.28 $\pm$ 0.46	78.27 $\pm$ 0.45	79.38 $\pm$ 0.43	80.02 $\pm$ 0.43	80.30 $\pm$ 0.43	80.57 $\pm$ 0.42
Diversity	77.28 $\pm$ 0.46	78.34 $\pm$ 0.46	79.18 $\pm$ 0.43	79.89 $\pm$ 0.43	80.82 $\pm$ 0.42	81.18 $\pm$ 0.42
Cooperation	77.28 $\pm$ 0.46	78.67 $\pm$ 0.46	80.20 $\pm$ 0.42	80.60 $\pm$ 0.43	80.72 $\pm$ 0.42	80.80 $\pm$ 0.42
Robust	77.28 $\pm$ 0.46	78.71 $\pm$ 0.45	80.26 $\pm$ 0.43	81.00 $\pm$ 0.42	81.22 $\pm$ 0.43	81.59 $\pm$ 0.42
Distilled Ensembles						
Robust-*dist*	–	79.44 $\pm$ 0.44	79.84 $\pm$ 0.44	80.01 $\pm$ 0.42	80.25 $\pm$ 0.44	80.63 $\pm$ 0.42
Robust-*dist*++	–	79.16 $\pm$ 0.46	80.00 $\pm$ 0.44	80.25 $\pm$ 0.42	80.35 $\pm$ 0.44	81.19 $\pm$ 0.43
1-shot						
Ensemble type	1	2	3	5	10	20
Independent	58.71 $\pm$ 0.62	60.04 $\pm$ 0.60	60.83 $\pm$ 0.63	61.34 $\pm$ 0.61	61.93 $\pm$ 0.61	62.06 $\pm$ 0.61
Diversity	58.71 $\pm$ 0.63	59.95 $\pm$ 0.61	61.27 $\pm$ 0.62	61.43 $\pm$ 0.61	62.23 $\pm$ 0.61	62.47 $\pm$ 0.62
Cooperation	58.71 $\pm$ 0.62	60.20 $\pm$ 0.61	61.46 $\pm$ 0.61	61.61 $\pm$ 0.61	62.06 $\pm$ 0.61	62.12 $\pm$ 0.62
Robust	58.71 $\pm$ 0.62	60.91 $\pm$ 0.62	62.36 $\pm$ 0.60	62.70 $\pm$ 0.61	62.97 $\pm$ 0.62	63.95 $\pm$ 0.61
Distilled Ensembles						
Robust-*dist*	–	62.33 $\pm$ 0.62	62.64 $\pm$ 0.60	63.14 $\pm$ 0.61	63.01 $\pm$ 0.62	63.06 $\pm$ 0.61
Robust-*dist* ++	–	62.07 $\pm$ 0.62	62.81 $\pm$ 0.60	63.39 $\pm$ 0.61	63.20 $\pm$ 0.62	63.73 $\pm$ 0.62

Table 1: **Few-shot classification accuracy on *Mini*ImageNet.** The first column gives the strategy, the top row indicates the number N of networks in an ensemble. Here, *dist* means that an ensemble was distilled into a single network, and '++' indicates that extra unannotated images were used for distillation. We performed 1 000 independent experiments on *Mini*ImageNet-test and report the average with 95% confidence interval. All networks are trained on *Mini*ImageNet-train set.

5-shot						
Full Ensemble	1	2	3	5	10	20
Independent	79.47 $\pm$ 0.49	81.34 $\pm$ 0.46	82.57 $\pm$ 0.46	83.16 $\pm$ 0.45	83.80 $\pm$ 0.45	83.95 $\pm$ 0.46
Diversity	79.47 $\pm$ 0.49	81.09 $\pm$ 0.45	82.23 $\pm$ 0.46	82.91 $\pm$ 0.46	84.30 $\pm$ 0.44	85.20 $\pm$ 0.43
Cooperation	79.47 $\pm$ 0.49	81.69 $\pm$ 0.46	82.95 $\pm$ 0.47	83.43 $\pm$ 0.47	84.01 $\pm$ 0.44	84.26 $\pm$ 0.44
Robust	79.47 $\pm$ 0.49	82.90 $\pm$ 0.46	83.36 $\pm$ 0.46	83.62 $\pm$ 0.45	84.47 $\pm$ 0.46	84.62 $\pm$ 0.44
Distilled Ensembles						
Robust-*dist*	–	82.72 $\pm$ 0.47	82.95 $\pm$ 0.46	83.27 $\pm$ 0.46	83.61 $\pm$ 0.46	83.57 $\pm$ 0.45
Robust-*dist*++	–	82.53 $\pm$ 0.48	83.04 $\pm$ 0.45	83.37 $\pm$ 0.46	83.22 $\pm$ 0.46	83.21 $\pm$ 0.44
1-shot						
Ensemble type	1	2	3	5	10	20
Independent	64.25 $\pm$ 0.73	66.60 $\pm$ 0.72	67.64 $\pm$ 0.71	68.07 $\pm$ 0.70	68.93 $\pm$ 0.70	69.64 $\pm$ 0.69
Diversity	64.25 $\pm$ 0.73	65.99 $\pm$ 0.71	66.71 $\pm$ 0.72	68.19 $\pm$ 0.71	69.35 $\pm$ 0.70	70.07 $\pm$ 0.70
Cooperation	64.25 $\pm$ 0.73	67.21 $\pm$ 0.71	67.93 $\pm$ 0.70	68.22 $\pm$ 0.70	68.69 $\pm$ 0.70	68.80 $\pm$ 0.68
Robust	64.25 $\pm$ 0.73	67.33 $\pm$ 0.71	68.01 $\pm$ 0.72	68.53 $\pm$ 0.70	68.59 $\pm$ 0.70	69.47 $\pm$ 0.69
Distilled Ensembles						
Robust-*dist*	–	67.47 $\pm$ 0.71	67.29 $\pm$ 0.72	68.09 $\pm$ 0.70	68.71 $\pm$ 0.71	68.77 $\pm$ 0.71
Robust-*dist*++	–	67.01 $\pm$ 0.74	67.62 $\pm$ 0.72	68.68 $\pm$ 0.71	68.38 $\pm$ 0.70	68.68 $\pm$ 0.69

Table 2: **Few-shot classification accuracy on CUB.** The first column gives the type of ensemble and the top row indicates the number of networks in an ensemble. Here, *dist* means that an ensemble was distilled into a single network, and '++' indicates that extra unannotated images were used for distillation. We performed 1000 independent experiments on CUB-test and report the average with 95% confidence interval. All networks are trained on CUB-train set.

Method	Meta	Network	5-shot	1-shot
Cosine + Attention [11]	✓	ResNet18	73.00 $\pm$ 0.64	56.20 $\pm$ 0.86
PPA [23]	✓	ResNet18	73.74 $\pm$ 0.19	59.60 $\pm$ 0.41
TADAM [22]	✓	ResNet18	76.70 $\pm$ 0.30	58.50 $\pm$ 0.30
LEO [28]	✓	WideResNet	77.59 $\pm$ 0.12	61.76 $\pm$ 0.08
FEAT [34]	✓	WideResNet	78.32 $\pm$ 0.16	61.72 $\pm$ 0.11
Linear Classifier [5]		ResNet18	74.27 $\pm$ 0.63	51.75 $\pm$ 0.80
Cosine Classifier [5]		ResNet18	75.68 $\pm$ 0.63	51.87 $\pm$ 0.77
Robust 20-*dist* (ours)		ResNet18	80.63 $\pm$ 0.43	63.06 $\pm$ 0.63
Robust 20-*dist*++ (ours)		ResNet18	**81.19** $\pm$ 0.43	**63.73** $\pm$ 0.62
Robust 20 Full		ResNet18	81.59 $\pm$ 0.42	63.95 $\pm$ 0.61

Table 3: **Comparison of distilled ensembles to other methods on 1- and 5-shot *mini*ImageNet.** The two last columns display the accuracy on 1- and 5-shot learning tasks. Column 2 indicates if a method is using meta-learning and the third column indicates the network used as a backbone. To evaluate our methods we performed $1\,000$ independent experiments on *Mini*ImageNet-test and report the average and 95% confidence interval. Here, '++' means that extra non-annotated images were used to perform distillation. The last model is a full ensemble and should not be directly compared to the rest of the table.

sity (6) or cooperation (7) penalties, we however did not see any improvement over the basic ensemble method. Instead, we found that computing the class probabilities conditioned on not being the ground truth label, as explained in Section 3.3, would perform much better.

This is illustrated on the following experiment with two network ensembles of size $n = 5$, where we compared the two strategies. We enforce similarity on the full probability vectors in the first one, computed with softmax at $T = 10$ following [1], and with conditionally non-ground-truth probabilities for the second one as defined in Section 3.3. When using the cooperation training formulation, the second strategy turns out to perform about 1% better than the first one (79.79 % vs 80.60%), when tested on *Mini*ImageNet. Similar observations have been made using the diversity criterion. In comparison, the basic ensemble method without interactions achieves about 80%.

Choice of relationship function. In principle, any similarity measure taken with the negative sign should potentially enforce cooperation, when used as a penalty. For promoting diversity, the same should hold in principle but with the opposite sign as well.

Here, we show that in fact, selecting the right criterion for comparing probability vectors (cosine similarity, L2 distance, symmetrized KL divergence), is crucial depending on the desired effect (cooperation or diversity). In Table 4, we perform such a comparison for an ensemble with $n = 5$ networks on the *Mini*ImageNet dataset for a $5-$shot classification task, when plugging the above function in the formulation 5, with a specific sign. The parameter γ for each experiment is chosen such that the performance on the validation set is maximized.

Purpose (Sign)	L2	-cos	$\mathrm{KL}_{\mathrm{sim}}$
Cooperation (+)	80.14 $\pm$ 0.43	80.29 $\pm$ 0.44	80.72 $\pm$ 0.42
Diversity (-)	80.54 $\pm$ 0.44	80.82 $\pm$ 0.42	79.81 $\pm$ 0.43

Table 4: **Evaluating different relationship criteria on mini-Imagenet 5-shot** The first row indicates which function was used as a relationship criteria, the first column indicates for which purpose the function is used and the corresponding sign. To evaluate our methods, we performed $1\,000$ independent experiments on CUB-test and report the average accuracy with 95% confidence intervals. All ensembles are trained on *Mini*ImageNet-train.

When looking for diversity, the cosine similarity performs slightly better than negative L2 distance, although the accuracies are within error bars. Using negative $\mathrm{KL}_{\mathrm{sim}}$ with various γ was either not distinguishable from independent training or was hurting the performance for larger values of γ (not reported on the table). As for cooperation, positive $\mathrm{KL}_{\mathrm{sim}}$ gives better results than L2 distance or negative cosine similarity. We believe that this behavior is due to important difference in the way these functions compare small values in probability vectors. While negative cosine or L2 losses would penalize heavily the largest difference, $\mathrm{KL}_{\mathrm{sim}}$ concentrates on values that are close to 0 in one vector and are greater in the second one.

4.5 Performance under domain shift

Finally, we evaluate the performance of ensemble methods under domain shift. We proceed by meta-training the models on the *Mini*ImageNet training set and evaluate the model on the CUB-test set. The following setting was first proposed by [5] and aims at evaluating the performance of algorithms to adapt when the difference between training and testing distributions is large. To compare to the results reported in the original work, we adopt their CUB test split. Table 5 compares our results to the ones listed in [5]. We can see that neither the full robust ensemble nor its distilled version are able to do better than training a linear classifier on top of a frozen network. Yet, it does significantly better than distance-based approaches (denoted by cosine classifier in the table). However, if a diverse ensemble is used, it achieves the best accuracy. This is not surprising and highlights the importance of having diverse models when ensembling weak classifiers.

Method	*mini*ImageNet → CUB
MatchingNet [32]	53.07 ± 0.74
MAML [9]	51.34 ± 0.72
ProtoNet [30]	62.02 ± 0.70
Linear Classifier [5]	**65.57** ± 0.70
Cosine Classifier [5]	62.04 ± 0.76
Robust 20-*dist*++ (ours)	64.23 ± 0.58
Robust 20 Full (ours)	65.04 ± 0.57
Diverse 20 Full (ours)	66.17 ± 0.55

Table 5: **5-shot classification accuracy under domain shift.** The last two models are full ensembles and should not be directly compared with the rest of the table. We performed 1 000 independent experiments on `CUB-test` from [5] and report the average and confidence interval here. All ensembles are trained on *Mini*ImageNet.

5 Conclusions

In this paper, we show that distance-based classifiers for few-shot learning suffer from high variance, which can be significantly reduced by using an ensemble of classifiers. Unlike traditional ensembling paradigms where diversity of predictions is encouraged by various randomization and data augmentation techniques, we show that encouraging the networks to cooperate during training is also important.

The overall performance of a single network obtained by distillation (with no computational overhead at test time) leads to state-of-the-art performance for few shot learning, without relying on the meta-learning paradigm. While such a result may sound negative for meta-learning approaches, it may simply mean that a lot of work remains to be done in this area to truly learn how to learn or to adapt.

Acknowledgment

This work was supported by a grant from ANR (MACARON project under grant number ANR-14-CE23-0003-01), by the ERC grant number 714381 (SOLARIS project), the ERC advanced grant ALLEGRO and gifts from Amazon and Intel.

References

[1] R. Anil, G. Pereyra, A. Passos, R. Ormandi, G. E. Dahl, and G. E. Hinton. Large scale distributed neural network training through online distillation. In *International Conference on Learning Representations (ICLR)*, 2018.

[2] A. Bietti, G. Mialon, D. Chen, and J. Mairal. A kernel perspective for regularizing deep neural networks. *preprint arXiv:1810.00363*, 2019.

[3] L. Breiman et al. Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, 24(6):2350–2383, 1996.

[4] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[5] 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 (ICLR)*, 2019.

[6] T. G. Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, 2000.

[7] C. Doersch and A. Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[8] N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid. Blitznet: A real-time deep network for scene understanding. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[9] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

[10] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. 2001.

[11] S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[17] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[21] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(11):2624–2637, 2013.

[22] B. Oreshkin, P. R. López, and A. Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[23] S. Qiao, C. Liu, W. Shen, and A. L. Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[24] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2017.

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[26] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[28] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell. Meta-learning with latent embedding optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[29] J. Schmidhuber, J. Zhao, and M. Wiering. Shifting inductive bias with success-story algorithm, adaptive levin search, and incremental self-improvement. *Machine Learning*, 28(1):105–130, 1997.

[30] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[31] S. Thrun. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. 1998.

[32] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[33] 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.

[34] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha. Learning embedding adaptation for few-shot learning. *arXiv preprint arXiv:1812.03664*, 2018.

[35] Y. Yoshida and T. Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.