ADAPTIVE ANCHOR LABEL PROPAGATION FOR TRANSDUCTIVE FEW-SHOT LEARNING

Michalis Lazarou[1]   *Yannis Avrithis*[2]   *Guangyu Ren*[1]   *Tania Stathaki*[1]

[1]Department of Electronic and Electronic Engineering, Imperial College London
[2] Institute of Advanced Research on Artificial Intelligence (IARAI)

ABSTRACT

Few-shot learning addresses the issue of classifying images using limited labeled data. Exploiting unlabeled data through the use of transductive inference methods such as label propagation has been shown to improve the performance of few-shot learning significantly. Label propagation infers pseudo-labels for unlabeled data by utilizing a constructed graph that exploits the underlying manifold structure of the data. However, a limitation of the existing label propagation approaches is that the positions of all data points are fixed and might be sub-optimal so that the algorithm is not as effective as possible. In this work, we propose a novel algorithm that adapts the feature embeddings of the labeled data by minimizing a differentiable loss function optimizing their positions in the manifold in the process. Our novel algorithm, *Adaptive Anchor Label Propagation*, outperforms the standard label propagation algorithm by as much as 7% and 2% in the 1-shot and 5-shot settings respectively. We provide experimental results highlighting the merits of our algorithm on four widely used few-shot benchmark datasets, namely *mini*ImageNet, *tiered*ImageNet, CUB and CIFAR-FS and two commonly used backbones, ResNet12 and WideResNet-28-10. The source code can be found at `https://github.com/MichalisLazarou/A2LP`.

Index Terms— few-shot learning, label propagation, transductive inference

1. INTRODUCTION

Enabling deep learning models to learn from limited labeled data has attracted significant interest in the research community, evident from the research output in the field of few-shot learning [1, 2, 3]. Few-shot learning investigates how machine learning models can learn from limited data, for example given only one or a few training examples per class. Multiple approaches have been proposed to address the few-shot learning problem, some of them include meta-learning methods [1, 2], representation learning methods [4, 5] and synthetic data generation methods [6, 7, 8].

While most of the aforementioned methods focus on inductive few-shot learning, transductive inference approaches have also been proposed as a way to leverage both labeled training examples and unlabeled test examples, referred to as
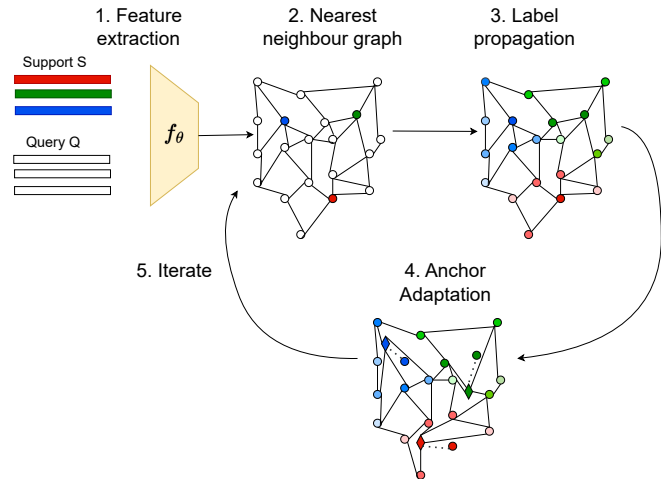
Fig. 1. Overview of our method. 1) Feature extraction of the support and query sets. 2) k-nearest neighbour graph construction. 3) Label propagation from labeled to unlabeled data. 4) Labeled anchor adaptation by minimizing a loss function. The diamonds represent the new position of the anchors and are connected with a dotted line to show the change from their original position. 5) We iterate this procedure over t-steps.

queries [9, 10, 11]. Label propagation [12, 13] is a well known transductive algorithm that has been used as a component in many few-shot transductive methods such as [9, 10, 14]. Label propagation exploits the data manifold by propagating labels from labeled to unlabeled data along the manifold. However, a limitation of the existing label propagation methods is that the labeled data are fixed and might not be in the optimal position to make the most accurate predictions.

To this end we investigate whether optimizing the position of the labeled data, referred as labeled anchors, can improve the performance of label propagation. More specifically we propose a novel algorithm named *Adaptive Anchor Label Propagation* (A^2LP) in which we adapt the feature embeddings of the labeled anchors by minimizing a multi-class cross entropy loss, placing them in a better position so that the performance of label propagation is improved. An overview of our algorithm can be seen in Figure 1.

Our contributions are summarized below:

- We propose a novel variant of label propagation algorithm named Adaptive Anchor Label Propagation (A^2LP).

- We show the successful application of A^2LP in the transductive few-shot learning setting.

- A^2LP outperforms significantly the standard label propagation as well as multiple state of the art methods on four benchmark datasets using two different backbones.

2. RELATED WORK

2.1. Few-shot learning

The most commonly researched paradigm addressing few-shot learning is meta-learning. There are three distinct lines of research in the field of meta-learning: optimization-based meta-learning [2, 15], metric-based meta-learning [3, 1] and model-based meta-learning [16, 17]. Another research direction addressing few-shot learning investigates how to obtain better feature representations utilizing various training techniques to train neural networks such as mixup [4], self-distillation [5] and pretext tasks such as predicting image rotations [18]. Synthetic data generation methods have also been prominent in the few-shot learning literature as a way to address the data deficiency of few-shot learning by using synthetic data. These methods exploit generative models such as GANs [7], VAEs [8] or other proposed generative models such as [6].

2.2. Transductive few-shot learning

Transductive few-shot learning methods utilize both labeled and unlabeled examples during inference to make predictions, achieving remarkable performance improvements over inductive methods. Some ideas that have been proposed in literature include iterative query selection [10, 11] where rather than making predictions for all the queries at the same time, the most trustworthy queries are selected to augment the labeled dataset. This procedure is iterated until all queries are classified. Other methods utilize the manifold structure of the data through the use of label propagation to train an embedding network [9] or to assign pseudo-labels to the queries [10]. Embedding propagation [14] also exploits the data manifold in order to smoothen the decision boundaries and reduce the noise of the class representations. Other ideas build on the direction of [1] by exploiting the unlabeled queries to improve the class prototypes, for example by using soft k-means [19] or by selecting the most confident queries per class [20].

Our work investigates the well-known algorithm of label propagation [12]. As stated in 1, a limitation of the existing label propagation approaches is that the labeled data have fixed positions that might be sub-optimal. To this end, we propose the *Adaptive Anchor Label Propagation* (A^2LP) algorithm that iteratively adapts the feature embeddings of the labeled

anchors, optimizing their positions such that label propagation can provide more accurate predictions.

3. METHOD

3.1. Problem formulation

At the embedding network pre-training stage, we assume access to a labeled dataset, D_{base}, where every image has a label corresponding to one of the classes from C_{base} and an embedding network f_θ. D_{base} is used to pre-train the embedding network, f_θ, that maps every image from image space $\mathcal{X}$ to a vector in the embedding space of dimensionality d, $f_\theta : \mathcal{X} \to \mathbb{R}^d$.

At inference stage, we use the pre-trained embedding network, f_θ, to solve novel tasks sampled from a dataset D_{novel} where every image belongs to a class from C_{novel}. It should be noted that $C_{base} \cap C_{novel} = \emptyset$. Following the common few-shot classification setting [3, 1], we calculate the accuracy performance over numerous novel tasks. An N-way K-shot novel task consists of a support set, S, of K labeled examples per class from N classes sampled from D_{novel}. Therefore, the total number of labeled examples in S is $N_S := NK$. In the transductive few-shot learning setting we assume access to the whole query set, Q, which consists of all the unlabeled query examples. The query set, Q, consists of examples from the same classes as the support set, S, with M examples per class. Therefore, the total number of examples in Q is $N_Q := NM$. Given f_θ, S and Q the goal is to classify all the queries from Q to one of the N classes from S.

3.2. Feature embeddings

We embed all examples from S and Q using f_θ into the vector set $V := \{\mathbf{v}_i\}_{i=1}^T$ where $\mathbf{v}_i := f_\theta(x_i)$ and $T := N_S + N_Q$. We ℓ_2-normalize the vector set V.

3.3. Nearest neighbour graph construction

Following [10], we construct the k-nearest neighbour graph of the vector set V by calculating the sparse affinity matrix, A defined as

$$A_{ij} := \begin{cases} [\cos(\mathbf{v}_i, \mathbf{v}_j)]_+^\gamma, & \text{if } i \neq j \wedge \mathbf{v}_i \in \text{NN}_k(\mathbf{v}_j) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $[\cos(\mathbf{v}_i, \mathbf{v}_j)]_+^\gamma$ is the nonnegative cosine similarity of vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ raised to the power of γ, where γ is a hyperparameter and $\text{NN}_k(\mathbf{v}_j)$ represents the k-nearest neighbours of $\mathbf{v}_j$ in V. We obtain the adjacency matrix $W := \frac{1}{2}(A + A^\top)$ before symmetrically normalizing it as

$$\mathcal{W} := D^{-1/2} W D^{-1/2}, \quad (2)$$

where D is the $T \times T$ degree matrix of $\mathcal{W}$ defined as

$$D := \text{diag}(\mathcal{W} \mathbf{1}_T) \quad (3)$$

where $\mathbf{1}_T$ is the all-ones vector of dimensionality T.

3.4. Label propagation

Following [21], we construct the $T \times N$ label matrix Y as

$$Y_{ij} := \begin{cases} 1, & \text{if } i \in N_S \wedge y_i = j \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

In other words, the first N_S rows of Y represent the one-hot labels of the support set, S, while the rest of the rows represent the query set Q and are filled with zeros. Label propagation is defined as

$$Z := (I - \alpha \mathcal{W})^{-1} Y, \quad (5)$$

where I is the identity matrix and $\alpha \in [0, 1)$ is a hyperparameter. By solving equation (5) we obtain the manifold similarity matrix Z, where the row vector z_i at row i expresses how similar example $\mathbf{v}_i$ is to each of the N support classes.

3.5. Adapting labeled anchors

We leverage the manifold similarity matrix Z in order to adapt the feature embeddings of the support vectors which serve as the labeled anchors, $V_S := \{\mathbf{v}_i\}_{i=1}^{N_S}$, and optimize their position in the k-nearest neighbour graph. We obtain the labeled anchor probability matrix P^a by calculating the probability distribution for every labeled anchor $\mathbf{v}_i \in V_S$ using the softmax function. Therefore the probability distribution p_i^a for the labeled anchor $\mathbf{v}_i$ is defined as:

$$p_i^a := \frac{\exp(\tau z_i)}{\sum_{j=1}^{N} \exp(\tau z_{ij})}, \quad (6)$$

where τ is a scalar and N is the total number of support classes. Using (6) we calculate the cross-entropy loss with respect to set V_S

$$L_{ce}(V_S) := - \sum_{i \in N_S} \sum_{j \in N} y_{ij} \log(p_{ij}^a), \quad (7)$$

where y_{ij} denotes the j-th component of the one-hot label y_i associated with $\mathbf{v}_i$ and p_{ij}^a denotes the j-th component of p_i^a. We update the feature embeddings of the labeled anchors V_S by minimizing (7) using gradient descent or any other gradient-based optimization algorithm with respect to V_S.

3.6. Iteration and Inference

We iterate the process from graph construction to adapting the labeled anchors for t-steps. Upon completion of the iteration process, we use the final Z to classify every query in Q to the class with the highest manifold class similarity [21]. The predicted labeled set of Q is denoted as $\hat{Y}_Q$.

Algorithm 1 summarizes our method for a single task.

Algorithm 1: Adaptive Anchor Label Propagation

   input : Pre-trained f_θ
   input : Support set, S
   input : Query set, Q
   output : Predicted labels for Q, $\hat{Y}_Q$

1 Calculate V by embedding S and Q using f_θ
2 ℓ_2-normalize every vector $\mathbf{v}$ in V
3 **for** t-steps **do**
4     $\mathcal{W} \leftarrow$ GRAPH$(f_\theta, S, Q; \gamma, k)$ ▷ Adjacency matrix (1),(2)
5     $Y \leftarrow$ LABEL(S) ▷ Label matrix (4)
6     $Z \leftarrow$ LP$(\mathcal{W}, Y; \alpha)$ ▷ Label propagation (5)
7     $P^a \leftarrow$ SOFTMAX$(Z[0 : N_S])$ ▷ Probability matrix (6)
8     $L_{ce}(V_S) \leftarrow$ CE$(P^a, Y[0 : N_S])$ ▷ Cross entropy loss (7)
9     $V_S \leftarrow$ UPDATE$(V_S, L_{ce}(V_S))$ ▷ Update labeled anchors
10 Predict $\hat{Y}_Q$

---

4. EXPERIMENTS

4.1. Setup

Datasets   We conduct exeeximents on four widely used few-shot learning benchmark datasets. These are: *mini*Imagenet [3], *tiered*Imagenet [4], CUB [22] and CIFAR-FS [22].

Tasks   We follow the paradigm of N-way, K-shot learning, where a task consists of a support set, S, of $N = 5$ classes and $K \in \{1, 5\}$ labeled examples per class. A task also includes a query set Q with $M = 15$ queries per class, making the total number of queries $N_Q = NM = 75$. We report the mean accuracy and the 95% confidence interval over 1000 sampled tasks from the novel dataset D_{novel}.

Networks   We experiment with two different network architectures, namely ResNet-12 and WideResNet-28-10. We use the publicly available pre-trained weights of both networks provided from [1]. ResNet-12 was trained following [11] and WideResNet-28-10 was trained following [4].

Implementation details and hyperparameters   Our implementation is based on python using the PyTorch framework [23]. We set $\gamma = 3$ in (1), $\tau = 15$ in (6), $\alpha = 0.8$ in (5) and $k = 20$ in (1). We set t-steps $= 1000$ in section 3.6 and use Adam optimizer with a learning rate $\eta = 0.0001$. We keep our hyperparameters fixed under all different datasets, backbones, 1-shot and 5-shot settings in order to keep our experiments simple and avoid overfitting.

4.2. Experimental results

Baselines   In order to have fair and objective experimental comparisons, we implement the following inference methods to serve as baselines: 1) the widely used prototypical classifier [1], 2) the weight imprinting method [24] followed by fine-tuning using L_{ce} (Imprinting+L_{ce}) and 3) the standard label propagation (LP) as it was used in [10, 12].

---

[1] https://github.com/MichalisLazarou/iLPC

ALGORITHM	*mini*IMAGENET		*tiered*IMAGENET		CIFAR-FS		CUB	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
RESNET-12								
Prototypical classifier	$54.23_{\pm0.61}$	$74.98_{\pm0.48}$	$67.21_{\pm0.72}$	$85.19_{\pm0.48}$	$60.65_{\pm0.71}$	$80.45_{\pm0.52}$	$75.13_{\pm0.62}$	$89.78_{\pm0.38}$
Imprinting+L_{ce}	$54.53_{\pm0.61}$	$74.31_{\pm0.49}$	$67.53_{\pm0.72}$	$84.86_{\pm0.48}$	$61.12_{\pm0.69}$	$80.32_{\pm0.53}$	$75.03_{\pm0.62}$	$89.49_{\pm0.37}$
LP	$59.15_{\pm0.66}$	$73.50_{\pm0.52}$	$72.67_{\pm0.74}$	$84.61_{\pm0.53}$	$66.60_{\pm0.74}$	$80.47_{\pm0.53}$	$79.73_{\pm0.61}$	$89.51_{\pm0.39}$
A^2LP	$\mathbf{64.35}_{\pm0.77}$	$\mathbf{75.00}_{\pm0.53}$	$\mathbf{80.49}_{\pm0.77}$	$\mathbf{85.80}_{\pm0.51}$	$\mathbf{73.27}_{\pm0.81}$	$\mathbf{81.32}_{\pm0.55}$	$\mathbf{87.15}_{\pm0.58}$	$\mathbf{90.35}_{\pm0.37}$
WIDERESNET-28-10								
Prototypical classifier	$65.35_{\pm0.63}$	$83.37_{\pm0.43}$	$73.47_{\pm0.70}$	$88.22_{\pm0.45}$	$74.14_{\pm0.67}$	$87.05_{\pm0.47}$	$81.06_{\pm0.64}$	$90.82_{\pm0.33}$
Imprinting+L_{ce}	$66.07_{\pm0.62}$	$83.34_{\pm0.42}$	$74.07_{\pm0.69}$	$88.55_{\pm0.43}$	$74.16_{\pm0.66}$	$87.15_{\pm0.47}$	$81.10_{\pm0.63}$	$91.17_{\pm0.33}$
LP	$69.50_{\pm0.64}$	$81.28_{\pm0.47}$	$78.14_{\pm0.72}$	$87.63_{\pm0.50}$	$78.66_{\pm0.67}$	$87.19_{\pm0.51}$	$85.11_{\pm0.63}$	$91.37_{\pm0.36}$
A^2LP	$\mathbf{76.48}_{\pm0.75}$	$\mathbf{83.57}_{\pm0.45}$	$\mathbf{82.82}_{\pm0.73}$	$\mathbf{88.80}_{\pm0.46}$	$\mathbf{82.15}_{\pm0.71}$	$\mathbf{87.95}_{\pm0.48}$	$\mathbf{86.45}_{\pm0.65}$	$\mathbf{91.66}_{\pm0.34}$

Table 1. *Transductive inference*. Comparisons of implemented baselines.

ALGORITHM	*mini*IMAGENET		*tiered*IMAGENET	
	1-shot	5-shot	1-shot	5-shot
WIDERESNET-28-10				
Prototypical classifier	$69.64_{\pm0.60}$	$84.61_{\pm0.42}$	$77.26_{\pm0.65}$	$89.22_{\pm0.42}$
Imprint+L_{ce}	$68.77_{\pm0.60}$	$84.24_{\pm0.42}$	$76.13_{\pm0.66}$	$88.95_{\pm0.42}$
LP	$74.24_{\pm0.66}$	$84.59_{\pm0.44}$	$82.48_{\pm0.70}$	$90.07_{\pm0.45}$
A^2LP	$\mathbf{75.94}_{\pm0.72}$	$\mathbf{85.67}_{\pm0.42}$	$\mathbf{83.68}_{\pm0.72}$	$\mathbf{90.53}_{\pm0.43}$

Table 2. *Transductive inference using PLC pre-processing*. Comparisons of implemented baselines.

ALGORITHM	*mini*IMAGENET		*tiered*IMAGENET	
	1-shot	5-shot	1-shot	5-shot
RESNET-12				
LR+ICI [11]	66.80	79.26	80.79	87.92
CAN+Top-k [20]	$67.19_{\pm0.55}$	$80.64_{\pm0.35}$	$73.21_{\pm0.58}$	$84.93_{\pm0.38}$
DPGN [25]	$67.77_{\pm0.32}$	$84.60_{\pm0.43}$	$72.45_{\pm0.51}$	$87.24_{\pm0.39}$
WIDERESNET-28-10				
EP [14]	$70.74_{\pm0.85}$	$84.34_{\pm0.53}$	$78.50_{\pm0.91}$	$88.36_{\pm0.57}$
SIB [26]	$70.00_{\pm0.60}$	$79.20_{\pm0.40}$	72.90	82.80
SIB+E^3BM [27]	$71.40_{\pm0.50}$	$81.20_{\pm0.40}$	$75.60_{\pm0.60}$	$84.30_{\pm0.40}$
LaplacianShot [28]	$74.86_{\pm0.19}$	$84.13_{\pm0.14}$	$80.18_{\pm0.21}$	$87.56_{\pm0.15}$
A^2LP	$\mathbf{76.48}_{\pm0.75}$	$83.57_{\pm0.45}$	$82.82_{\pm0.73}$	$88.80_{\pm0.46}$
A^2LP+PLC	$75.94_{\pm0.72}$	$\mathbf{85.67}_{\pm0.42}$	$\mathbf{83.68}_{\pm0.72}$	$\mathbf{90.53}_{\pm0.43}$

Table 3. *Transductive inference*. Comparisons of state of the art methods.

Discussion   It can be seen from Table 1 that A^2LP outperforms significantly all other baselines in both 1-shot and 5-shot settings under all datasets and backbones. Impressively, A^2LP outperforms the standard label propagation (LP) by more than 7% in the 1-shot CUB using ResNet-12 as shown in Table 1, highlighting the importance of adapting the labeled anchors' feature embeddings. Interestingly, simply fine-tuning the imprinted weights using L_{ce} (Imprinting+L_{ce}) cannot provide the same performance increase as A^2LP because A^2LP exploits the node connections of the constructed graph to adapt the labeled anchors in contrast to Imprint+L_{ce}.

We also investigate the robustness of A^2LP using the feature pre-processing technique proposed in [10, 19], referred to as PLC pre-processing. Instead of using only ℓ_2-normalization, PLC pre-processing consists of element-wise power transform $\mathbf{v}^{\frac{1}{2}}$ for $\mathbf{v} \in V$, followed by ℓ_2-normalization and centering by subtracting the mean of all vectors in V. It can be seen from Table 2 that A^2LP still outperforms all other baselines.

Lastly, we compare A^2LP with several state of the art transductive few-shot learning methods. Since each method uses different training regimes and network architectures, in order to make the comparisons as fair as possible we compare the best version of every method including ours. It can be seen from Table 3 that A^2LP outperforms the other methods in 3 out of 4 settings, outperforming its closest competitor by $\sim 2\%$ in the 1-shot *tiered*Imagenet setting. Furthermore, either A^2LP or A^2LP with PLC pre-processing outperforms all competitors.

5. CONCLUSIONS

In this work we propose the novel algorithm Adaptive Anchor Label Propagation (A^2LP). Our algorithm significantly outperforms the label propagation as it was used in [10, 12] as well as several state of the art methods in the transductive few-shot learning setting. Through our investigation it is evident that optimizing the position of the labeled anchors is essential for improving the performance of label propagation. In the future, we are interested to investigate different loss functions for adapting the feature embeddings of the labeled anchors.

6. REFERENCES

[1] Jake Snell, Kevin Swersky, and Richard Zemel, "Prototypical networks for few-shot learning," in *NeurIPS*, 2017.

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

[3] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al., "Matching networks for one shot learning," in *NeurIPS*, 2016.

[4] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian, "Charting the right manifold: Manifold mixup for few-shot learning," in *WACV*, 2020.

[5] 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.

[6] Michalis Lazarou, Tania Stathaki, and Yannis Avrithis, "Tensor feature hallucination for few-shot learning," in *WACV*, 2022.

[7] Kai Li, Yulun Zhang, Kunpeng Li, and Yun Fu, "Adversarial feature hallucination networks for few-shot learning," in *CVPR*, 2020.

[8] Qinxuan Luo, Lingfeng Wang, Jingguo Lv, Shiming Xiang, and Chunhong Pan, "Few-shot learning via feature hallucination with variational inference," in *WACV*, 2021.

[9] Y Liu, J Lee, M Park, S Kim, E Yang, SJ Hwang, and Y Yang, "Learning to propagate labels: Transductive propagation network for few-shot learning," in *ICLR*, 2019.

[10] Michalis Lazarou, Tania Stathaki, and Yannis Avrithis, "Iterative label cleaning for transductive and semi-supervised few-shot learning," in *ICCV*, 2021.

[11] Yikai Wang, C. Xu, Chen Liu, Liyong Zhang, and Yanwei Fu, "Instance credibility inference for few-shot learning," *CVPR*, 2020.

[12] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum, "Label propagation for deep semi-supervised learning," in *CVPR*, 2019.

[13] Xiaojin Zhu and Zoubin Ghahramani, "Learning from labeled and unlabeled data with label propagation," Tech. Rep., 2002.

[14] Pau Rodríguez, Issam Laradji, Alexandre Drouin, and Alexandre Lacoste, "Embedding propagation: Smoother manifold for few-shot classification," in *ECCV*, 2020.

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

[16] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap, "Meta-learning with memory-augmented neural networks," in *ICML*, 2016.

[17] Tsendsuren Munkhdalai and Hong Yu, "Meta networks," in *ICML*, 2017.

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

[19] Yuqing Hu, Vincent Gripon, and Stéphane Pateux, "Leveraging the feature distribution in transfer-based few-shot learning," in *International Conference on Artificial Neural Networks*. Springer, 2021.

[20] Ruibing Hou, Hong Chang, MA Bingpeng, Shiguang Shan, and Xilin Chen, "Cross attention network for few-shot classification," in *NeurIPS*, 2019.

[21] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf, "Learning with local and global consistency," in *NIPS*, 2003.

[22] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang, "A closer look at few-shot classification," in *ICLR*, 2019.

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," 2017.

[24] Hang Qi, Matthew Brown, and David G Lowe, "Low-shot learning with imprinted weights," in *CVPR*, 2018.

[25] Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu, "Dpgn: Distribution propagation graph network for few-shot learning," in *CVPR*, 2020.

[26] Shell Xu Hu, Pablo Garcia Moreno, Yang Xiao, Xi Shen, Guillaume Obozinski, Neil Lawrence, and Andreas Damianou, "Empirical bayes transductive meta-learning with synthetic gradients," in *ICLR*, 2019.

[27] Yaoyao Liu, Bernt Schiele, and Qianru Sun, "An ensemble of epoch-wise empirical bayes for few-shot learning," in *ECCV*, 2020.

[28] Imtiaz Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed, "Laplacian regularized few-shot learning," in *ICML*, 2020.