

Adaptive manifold for imbalanced transductive few-shot learning

Michalis Lazarou¹ Yannis Avrithis² Tania Stathaki¹
 Imperial College London¹
 Institute of Advanced Research on Artificial Intelligence²

Abstract

Transductive few-shot learning algorithms have showed substantially superior performance over their inductive counterparts by leveraging the unlabeled queries. However, the vast majority of such methods are evaluated on perfectly class-balanced benchmarks. It has been shown that they undergo remarkable drop in performance under a more realistic, imbalanced setting.

To this end, we propose a novel algorithm to address imbalanced transductive few-shot learning, named Adaptive Manifold. Our method exploits the underlying manifold of the labeled support examples and unlabeled queries by using manifold similarity to predict the class probability distribution per query. It is parameterized by one centroid per class as well as a set of graph-specific parameters that determine the manifold. All parameters are optimized through a loss function that can be tuned towards class-balanced or imbalanced distributions. The manifold similarity shows substantial improvement over Euclidean distance, especially in the 1-shot setting.

Our algorithm outperforms or is on par with other state of the art methods in three benchmark datasets, namely miniImageNet, tieredImageNet and CUB, and three different backbones, namely ResNet-18, WideResNet-28-10 and DenseNet-121. In certain cases, our algorithm outperforms the previous state of the art by as much as 4.2%.

1. Introduction

One of the fundamental challenges of deep learning is its reliance on large labeled datasets. Even though weak or self-supervision is gaining momentum, an even greater challenge is the difficulty of obtaining the data itself, even unlabeled. This is the case in applications where the data is scarce, for example in rare animal species [1].

The *few-shot learning* paradigm has attracted significant interest because it investigates the question of how to make deep learning models acquire knowledge from limited data [39, 35, 9]. Different methodologies have been proposed to address few-shot learning such as *meta-*

learning [35, 9, 14], *transfer learning* [37, 24, 20] and *synthetic data generation* [18, 23, 17]. The vast majority of these methods focus on the *inductive* setting, where the assumption is that at inference, every query example is classified independently of the others.

Recent studies have explored the *transductive* few-shot learning setting, where all query examples can be exploited together at test time, showing remarkable improvement in performance [16, 12, 40, 45, 29]. Some approaches exploit all query examples at the same time by utilizing the data manifold through label propagation [16] or through the properties of the oblique manifold [29]. Other approaches utilize the available query examples to improve the class centroids by specialized loss functions [2], using soft K-means [12] or by minimizing the cross-class and intra-class variance [21].

While the query set of transductive few-shot learning benchmarks is unlabeled, it is still curated in the sense that the tasks are perfectly class-balanced. Several state of the art methods are in fact based on this assumption and use class balancing approaches to improve their performance [16, 45, 12, 2]. However, it has been argued that this is not a realistic setting [38]. As a way to address this flaw, the latter study introduced a new *imbalanced transductive few-shot learning* setting, comparing numerous state of the art methods under a fair setting and showing that their performance drops dramatically.

In this work, focusing on this imbalanced transductive setting [38], we introduce a new algorithm, called *Adaptive Manifold* (AM), that combines the merits of class centroid approaches and data manifold exploitation approaches. In particular, as illustrated in **Figure 1**, we initialize the class centroids from the labeled support examples and we propagate the labels along the data manifold, using a k -nearest neighbour graph [13]. Using the loss function proposed in [38] we iteratively update both the class centroids as well as the graph parameters. Our algorithm outperforms other state of the art methods in the imbalanced transductive few-shot learning setting.

In summary, we make the following contributions:

- We are the first, to the best of our knowledge, to obtain class centroids through manifold class similarity on a k -

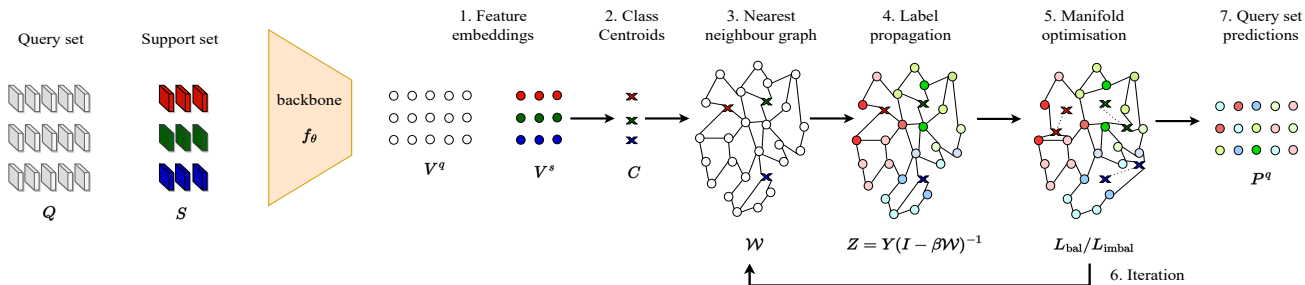


Figure 1. *Overview of our method.* 1) Given a support set, S , and a query set, Q , we extract features V^s and V^q using the pre-trained backbone f_θ . 2) We calculate class centroids, C , using (1). 3) We calculate the k -nearest neighbour graph using (3), (4), (5) and (6). 4) We perform label propagation using (8). 5) We optimize the manifold parameters, Φ , using (14) or (17). 6) We iterate the procedure from graph construction for r steps. 7) We predict pseudo-labels using P^q .

nearest neighbour graph and optimize jointly the class centroids along with graph-specific parameters.

- We achieve new state of the art performance on the imbalanced transductive few-shot setting under multiple benchmark datasets and networks, outperforming by as much as 4.2% the previous state of the art in the 1-shot setting.
- Our method can also perform on par or even outperform many state of the art methods in the standard balanced transductive few-shot setting.

2. Related work

2.1. Few-shot learning

Learning from limited data is a long-standing problem [8]. A large number of the current methods focus on the *meta-learning* paradigm. These can be grouped into three directions: model-based [25, 34, 26, 11], optimization based [9, 27, 30, 33] and metric-based [39, 35, 14, 36]. *Model-based* methods utilize specialized networks such as memory augmented [34] and meta-networks [26] to aid the meta-learning process. *Optimization-based* methods focus on learning a robust model initialization, through gradient-based solutions [9, 27], closed-form solutions [1] or an LSTM [31]. *Metric-based* approaches operate in the embedding space, based on similarities of a query example with class centroids [35], using a learned similarity function [36] or a Siamese network to compare image pairs [14].

Recent works [37, 5] have shown that the transfer learning paradigm can outperform meta-learning methods. *Transfer learning* methods decouple the training from the inference stage and aim at learning powerful representations through the use of well-designed pre-training regimes to train the backbone network. This often involves auxiliary loss functions along with the standard cross entropy loss, such as *knowledge-distillation* [37], mixup-based *data augmentation* [24] and *self-supervision*, such as predicting rota-

tions [10] and contrastive learning [41].

Another way to address the data deficiency is to augment the support set with synthetic data. *Synthetic data* can be generated either in the image space or in the feature space, by using a *hallucinator* trained on the base classes. The hallucinator can be trained using common generative models, such as *generative adversarial networks* (GANs) [43, 18, 22] and *variational autoencoders* (VAEs) [23]. Hallucinators have also been specifically designed for the few-shot learning paradigm [6, 42, 4, 17].

2.2. Transductive few-shot learning

Transductive few-shot learning studies the case where all queries are available at inference time and can be exploited to improve predictions. Several methods exploit the data manifold, by using label propagation [16, 19] or embedding propagation [32], or are based on Riemannian geometry, by using the oblique manifold [29]. Another direction is to use both labeled and unlabeled examples to improve the class centroids. For example, one may use soft k -means to iteratively update the class centroids [12], rectify prototypes by minimizing the inter-class and intra-class variance [21], or iteratively adapt class centroids by maximizing the mutual information between query features and their label predictions [2]. It has also been proposed to iteratively select the most confident pseudo-labeled queries, for example by interpreting this problem as label denoising [16] or by calculating the credibility of each pseudo-label [40].

2.3. Class balancing

The commonly used transductive few-shot learning benchmarks use perfectly class-balanced tasks [2]. Several methods exploit this bias by encouraging class-balanced predictions over queries, thereby improving their performance. One way is to optimize, through the Sinkhorn-Knopp algorithm, the query probability matrix, P , to have specific row and column sums \mathbf{p} and \mathbf{q} respectively [12, 16]. The row

sum \mathbf{p} amounts to the probability distribution of every query, while the column sum corresponds to the total number of queries per class. Another way is to maximize the entropy of the marginal distribution of predicted labels over queries, thus encouraging it to follow a uniform distribution [2].

However, the authors of [38] argue that using perfectly class-balanced tasks is unrealistic. They propose a more realistic imbalanced setting and protocol, benchmarking the performance of several methods. They also introduce a relaxed version of [2] based on α -divergence, which can effectively address class-imbalanced tasks.

3. Method

3.1. Problem formulation

Representation learning We assume access to a *base dataset* $D_{\text{base}} = \{(x_i, \mathbf{y}_i)\}_{i=1}^B$ of B images, where each image x_i has a corresponding one-hot encoded label \mathbf{y}_i over a set of *base classes* C_{base} . Denoting by \mathcal{X} the image space, we assume access to a network $f_\theta : \mathcal{X} \rightarrow \mathbb{R}^d$ has been trained on D_{base} , which maps an image $x \in \mathcal{X}$ to an embedding $f_\theta(x) \in \mathbb{R}^d$.

Inference We assume access to a *novel dataset* D_{novel} consisting of images with corresponding labels from a set C_{novel} of *novel classes*, where $C_{\text{novel}} \cap C_{\text{base}} = \emptyset$. We sample N -way K -shot tasks, each consisting of a labeled support set, $S = \{(x_i^s, \mathbf{y}_i^s)\}_{i=1}^L$, where each image x_i^s has a corresponding one-hot encoded label $\mathbf{y}_i^s = (y_{ji}^s)_{j=1}^N \in \{0, 1\}^N$ over C_{novel} , with N novel classes in total and K examples per class, such that the number of examples in S is $L = |S| = NK$. We focus on the transductive setting, therefore a task also contains an unlabeled *query set* $Q = \{x_i^q\}_{i=1}^M$ sampled from the same N classes as the support set S where the number of examples in Q is $M = |Q|$.

Feature extraction Given a novel task, we embed all images in S and Q using f_θ and a feature pre-processing function $\eta : \mathbb{R}^d \rightarrow \mathbb{R}^d$, to be discussed in section 4. Let $V^s = (\mathbf{v}_1^s \cdots \mathbf{v}_L^s)$ be the $d \times L$ matrix containing the embeddings of S , where $\mathbf{v}_i^s = \eta(f_\theta(x_i^s)) \in \mathbb{R}^d$. Similarly, let $V^q = (\mathbf{v}_1^q \cdots \mathbf{v}_M^q)$ be the $d \times M$ matrix containing the embeddings of Q , where $\mathbf{v}_i^q = \eta(f_\theta(x_i^q)) \in \mathbb{R}^d$. We also represent V^s, V^q as sets $\mathcal{V}^s = \{\mathbf{v}_i^s\}_{i=1}^L$, $\mathcal{V}^q = \{\mathbf{v}_i^q\}_{i=1}^M$. Both sets remain fixed in our method.

3.2. Class centroids

Following [35], we define a class centroid $\mathbf{c}_j \in \mathbb{R}^d$ in the embedding space for each class j in the support set S . The centroids are learnable variables but initialized by standard class prototypes [35]. That is, the centroid \mathbf{c}_j of class j is initialized by the mean

$$\mathbf{v}_j^c = \frac{1}{K} \sum_{\mathbf{v}_i^s \in \mathcal{V}^s} y_{ji}^s \mathbf{v}_i^s \quad (1)$$

of support embeddings assigned to class j . Let $C = (\mathbf{c}_1 \cdots \mathbf{c}_N)$ be the $d \times N$ matrix containing the learnable centroids of all N support classes. We also represent C as a set $\mathcal{C} = \{\mathbf{c}_j\}_{j=1}^N$.

3.3. Nearest neighbour graph

We collect centroids, support and query embeddings in a single $d \times T$ matrix

$$V = (\mathbf{v}_1 \cdots \mathbf{v}_T) = (C \ V^s \ V^q), \quad (2)$$

where $T = N + L + M$. We also represent V as a set $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^T$. Following [13, 16], we construct a k -nearest neighbour graph of \mathcal{V} . We define edges between distinct nearest neighbours in \mathcal{V} that are not both centroids:

$$E = \{(\mathbf{v}_i, \mathbf{v}_j) \in \mathcal{V}^2 \setminus \mathcal{C}^2 : \mathbf{v}_i \in \text{NN}_k(\mathbf{v}_j)\}, \quad (3)$$

where $\text{NN}_k(\mathbf{v})$ is the set of k -nearest neighbours of \mathbf{v} in \mathcal{V} , excluding \mathbf{v} . Given E , we define the $T \times T$ *affinity matrix* $A = (a_{ij})$ as

$$a_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{v}_i - \mathbf{v}_j\|^2}{g_{ij}\sigma^2}\right), & \text{if } (\mathbf{v}_i, \mathbf{v}_j) \in E \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

where g_{ij} is a learnable pairwise scaling factor for every pair $(\mathbf{v}_i, \mathbf{v}_j)$, collectively represented by $T \times T$ matrix $G = (g_{ij})$, and σ^2 is a global scaling factor set equal to the standard deviation of $\|\mathbf{v}_i - \mathbf{v}_j\|^2$ for $(\mathbf{v}_i, \mathbf{v}_j) \in \mathcal{V}^2$ as in [32]. We symmetrize A into the $T \times T$ *adjacency matrix*, $W = \frac{1}{2}(A + A^\top)$. We calculate W_B which is a scaled version of W defined as:

$$W_B = W \circ B \quad (5)$$

where $B \in [0, 1]^{T \times T}$ is a learnable $T \times T$ matrix and \circ is the Hadamard product. We normalize W_B by

$$\mathcal{W} = D^{-1/2} W_B D^{-1/2}, \quad (6)$$

where $D = \text{diag}(W_B \mathbf{1}_T)$ is the $T \times T$ degree matrix of W_B .

3.4. Label Propagation

Labels Following [44], we define the $N \times T$ *label matrix*

$$Y = (Y^c \ Y^s \ Y^q) = (I_N \ \mathbf{0}_{N \times L} \ \mathbf{0}_{N \times M}). \quad (7)$$

That is, Y has one row per class and one column per example, which is an one-hot encoded label for every class centroid in \mathcal{C} and a zero vector for both support embeddings \mathcal{V}^s and query embeddings \mathcal{V}^q .

Label propagation Given the graph represented by \mathcal{W} and the label matrix Y , label propagation amounts to

$$Z = Y(I - \beta \mathcal{W})^{-1}, \quad (8)$$

where $\beta \in [0, 1)$ is a scalar hyperparameter that is referred to as α in the standard label propagation [44].

Predicted probabilities The resulting $N \times T$ matrix $Z = (\mathbf{z}_1 \cdots \mathbf{z}_T)$ is called *manifold class similarity matrix*, in the sense that column $\mathbf{z}_i \in \mathbb{R}^N$ expresses how similar embedding vector \mathbf{v}_i is to each of the N support classes. By taking softmax over columns

$$\mathbf{p}_i = \frac{\exp(\tau \mathbf{z}_i)}{\sum_{j=1}^N \exp(\tau z_{ji})}, \quad (9)$$

with $\tau > 0$ being a positive scale hyperparameter, we define the $N \times T$ *probability matrix*

$$P = (\mathbf{p}_1 \cdots \mathbf{p}_T) = (P^c \ P^s \ P^q). \quad (10)$$

Matrix P expresses the predicted probability distributions over the support classes. If $P = (p_{ji})$, element p_{ji} expresses the predicted probability of class j for example i . Similarly for class centroids $P^c = (p_{ji}^c) \in \mathbb{R}^{N \times N}$, support examples $P^s = (p_{ji}^s) \in \mathbb{R}^{N \times L}$ and queries $P^q = (p_{ji}^q) \in \mathbb{R}^{N \times M}$.

3.5. Loss function: Class balancing or not

The set of all learnable parameters is $\Phi = \{C, G, B\}$ is optimized jointly using a mutual information loss [2, 38]. We distinguish between class-balanced and imbalanced tasks.

3.5.1 Class-balanced tasks

Following [2], we optimize parameters Φ using three loss terms. The first is standard average cross-entropy over the labeled support examples:

$$L_{\text{CE}}(P^s) = -\frac{1}{L} \sum_{i=1}^L \sum_{j=1}^N y_{ji}^s \log(p_{ji}^s). \quad (11)$$

The second is the average, over queries, entropy of predicted class probability distributions per query

$$\bar{\mathcal{H}}(P^q) = -\frac{1}{M} \sum_{i=1}^M \sum_{j=1}^N p_{ji}^q \log(p_{ji}^q). \quad (12)$$

This term aims at minimizing the uncertainty of the predicted probability distribution of every query, hence encouraging confident predictions. The third term is

$$-\mathcal{H}(\bar{\mathbf{p}}^q) = \sum_{j=1}^N \bar{p}_j^q \log(\bar{p}_j^q), \quad (13)$$

where $\bar{p}_j^q = \frac{1}{M} \sum_{i=1}^M p_{ji}^q$ and $\bar{\mathbf{p}}^q = (\bar{p}_j^q)_{j=1}^N = P^q \mathbf{1}_M \in \mathbb{R}^N$ is a vector representing the average predicted probability distribution of set Q . By maximizing its entropy, this term aims at maximizing its uncertainty, encouraging it to be uniform, hence balancing over classes.

The complete loss function to be minimized w.r.t. Φ is

$$L_{\text{bal}} = \lambda_3 L_{\text{CE}}(P^s) + \lambda_2 \bar{\mathcal{H}}(P^q) - \lambda_1 \mathcal{H}(\bar{\mathbf{p}}^q), \quad (14)$$

where $\lambda_1, \lambda_2, \lambda_3$ are scalar hyperparameters.

3.5.2 Imbalanced tasks

By encouraging the average predicted probability distribution to be uniform, the third term (13) is strongly biased towards class-balanced tasks. To make the loss more tolerant to imbalanced distributions, a relaxed version has been proposed based on the α -divergence [38]. In particular, the second (12) and third term (13) become respectively

$$\bar{\mathcal{H}}_\alpha(P^q) = -\frac{1}{\alpha-1} \frac{1}{M} \sum_{i=1}^M \sum_{j=1}^N (p_{ji}^q)^\alpha \quad (15)$$

$$-\mathcal{H}_\alpha(\bar{\mathbf{p}}^q) = \frac{1}{\alpha-1} \sum_{j=1}^N (\bar{p}_j^q)^\alpha \quad (16)$$

In this case, the complete loss function (14) to be minimized with respect to Φ is modified as

$$L_{\text{imbal}} = \lambda_3 L_{\text{CE}}(P^s) + \lambda_2 \bar{\mathcal{H}}_\alpha(P^q) - \lambda_1 \mathcal{H}_\alpha(\bar{\mathbf{p}}^q). \quad (17)$$

3.6. Manifold parameter optimization

In contrast to [2] and [38], rather than only optimizing the class centroids, we optimize the entire set of manifold parameters Φ , which includes the class centroids C as well as graph-specific parameters G (4) and B (5). We update Φ by minimizing (14) or (17) through any gradient-based optimization algorithm with learning rate ϵ . The entire procedure from graph construction in subsection 3.3 to manifold parameter optimization in subsection 3.6 is iterated for r steps. Algorithm 1 summarizes the complete optimization procedure of our method.

Algorithm 1: Adaptive Manifold (AM).

```

input : Pre-trained backbone  $f_\theta$ 
input : labeled support set  $S$  with  $|S| = L$ 
input : unlabeled query set  $Q$  with  $|Q| = M$ 

1  $(V^s, V^q) \leftarrow (f_\theta(S), f_\theta(Q))$ 
2  $C \leftarrow \text{CENTROIDS}(V^s)$  ▷ class centroids (1)
3  $\mathcal{V} \leftarrow \{C, V^s, V^q\}$ 
4  $(G, B) \leftarrow \text{INITIALIZE}()$ 
5  $\Phi \leftarrow \{C, G, B\}$ 
6 for  $r$  steps do
7    $A \leftarrow \text{AFFINITY}(\mathcal{V}; G, k)$  ▷ affinity matrix (4)
8    $W \leftarrow \frac{1}{2}(A + A^T)$  ▷ symmetric adjacency matrix
9    $W_B \leftarrow W \circ B$  ▷ scaled adjacency matrix (5)
10   $\mathcal{W} \leftarrow D^{-1/2} W_B D^{-1/2}$  ▷ adjacency matrix (6)
11   $Y \leftarrow (I_N \ \mathbf{0}_{N \times L} \ \mathbf{0}_{N \times M})$  ▷ label matrix (7)
12   $Z \leftarrow Y(I - \beta \mathcal{W})^{-1}$  ▷ label propagation (8)
13   $P \leftarrow \text{SOFTMAX}(Z)$  ▷ class probabilities (9)
14   $L_{\text{bal}}/L_{\text{imbal}} \leftarrow \text{LOSS}(P; \Phi)$  ▷ loss function (14) or (17)
15   $\Phi \leftarrow \text{UPDATE}(\Phi; L_{\text{bal}}/L_{\text{imbal}})$ 
16 return  $P^q$ 

```

3.7. Transductive Inference

Upon convergence of the optimization of manifold parameters Φ , we obtain the final query probability matrix P^q (10)

and for each query $x_i^q \in Q$, we predict the *pseudo-label*

$$\hat{y}_i^q = \arg \max_j p_{ji}^q \quad (18)$$

corresponding to the maximum element of the i -th column of matrix P^q .

4. Experiments

4.1. Setup

Datasets We experiment on three commonly used few-shot learning benchmark datasets, namely *miniImageNet* [39], *tieredImageNet* [3] and CUB [5]. In state of the art comparisons in the balanced setting, we also use CIFAR-FS [5, 15].

Backbones We use the three pre-trained backbones from the publicly available code [38], namely ResNet-18, WideResNet-28-10 (WRN-28-10) and DenseNet-121. All are trained using standard cross entropy loss on D_{base} for 90 epochs with learning rate 0.1, divided by 10 at epochs 45 and 66. Color jittering, random cropping and random horizontal flipping augmentations are used at training. We also carry out experiments using the publicly available code and pre-trained WRN-28-10 backbones provided by [16].

Tasks Unless otherwise stated, we consider N -way, K -shot tasks with $N = 5$ randomly sampled classes from C_{novel} and $K \in \{1, 5\}$ random labeled examples for the support set S . The query set Q contains $M = 75$ query examples in total. In the balanced setting, there are $\frac{M}{N} = \frac{75}{5} = 15$ queries per class. In the imbalanced setting, the total number of queries remains $M = 75$. Following [38], we sample imbalanced tasks by modeling the proportion of examples from each class in Q as a vector $\pi = (\pi_1, \dots, \pi_N)$ sampled from a symmetric Dirichlet distribution $\text{Dir}(\gamma)$ with parameter $\gamma = 2$. We follow [38] and [16], performing 10000 and 1000 tasks respectively when using the code and settings of each work.

Implementation details Our implementation is in Pytorch [28]. We carry out experiments for balanced and imbalanced transductive few-shot learning using the publicly available code provided by [38]¹. For additional experiments in the balanced setting, we use the publicly available code provided from [16]². We used Adam optimizer in for the manifold parameter optimization subsection 3.6.

Hyperparameters Following [38], we keep the same values of hyper-parameters τ , λ_1 , λ_2 , λ_3 , ϵ and r . We set $\epsilon = 0.0001$, $r = 1000$, $\tau = 15$ (9). In the imbalanced setting we set $\lambda_1 = \lambda_2 = \lambda_3 = 1$, while in the balanced

setting we set $\lambda_1 = \lambda_3 = 1$ and $\lambda_2 = 10$. Regarding hyper-parameter α (15),(16) we ablate it in section subsection 4.2 and set $\alpha = 2$ for 1-shot and $\alpha = 5$ for 5-shot for all experiments unless stated otherwise. For label propagation, we set $k = 20$ (3) for 1-shot and $k = 10$ for 5-shot; we initialize $G = J_T$ (4) and $B = J_T$ (5) where J_T is the $T \times T$ all-ones matrix; we initialize $\beta = 0.8$ (8) for 1-shot and $\beta = 0.9$ for 5-shot. We optimized k , β and the initialization of G and B on the *miniImageNet* validation set. To avoid hyperparameter overfitting, *all hyper-parameters are kept fixed across all datasets and backbones*.

Baselines In the imbalanced setting, we compare our method against the state of the art method α -TIM [38], basing our experiments on the publicly available code and comparing against all methods implemented in that code. In the balanced setting, we reproduce results of all methods provided in the official code from [38] and compare our method against them. Furthermore regarding the balanced setting, we compare our method against other state of the art methods such as [16, 45] by using the official code from [16].

Feature pre-processing We experiment with two commonly used feature pre-processing methods, denoted as η in subsection 3.1, namely ℓ_2 -normalization and the method used in [12, 16], which we refer to as PLC. ℓ_2 -normalization is defined as $\frac{\mathbf{v}}{\|\mathbf{v}\|_2}$ for $\mathbf{v} \in V$. PLC, standing for *power transform*, ℓ_2 -normalization, centering, performs element-wise power transform $\mathbf{v}^{\frac{1}{2}}$ for $\mathbf{v} \in V$, followed by ℓ_2 -normalization and centering, subtracting the mean over V .

In the balanced and imbalanced settings respectively, we refer to our method as AM, α -AM when using ℓ_2 -normalization and as AM_{PLC} , α - AM_{PLC} when using PLC pre-processing. TIM [2] and α -TIM [38] use only ℓ_2 -normalization originally. For fair comparisons, we also use PLC pre-processing on TIM and α -TIM, referring to them as TIM_{PLC} , α - TIM_{PLC} in the balanced and imbalanced settings respectively.

Reporting results In every table we denote the best performing results with bold regardless the pre-processing method used. Nevertheless, since our work is influenced by [2] and [38], we also compare with these two methods under the same feature pre-processing settings. In Table 2, Table 3, Table 5, Table 6 and Table 7, we use the code by [38], reporting the mean accuracy over 10000 tasks [38]. In Table 4, we use the code by [16], reporting the mean accuracy and 95% confidence interval over 1000 tasks. In the ablation study in Table 1 we use the code by [38], however, since we ablate our own method, we report both the mean accuracy and the 95% confidence interval.

4.2. Ablation study

Ablation of hyper-parameter α Figure 2 ablates α -AM and α -TIM with respect to α . It can be seen that the value

¹https://github.com/oveilleux/Realistic_Transductive_Few_Shot

²<https://github.com/MichalisLazarou/iLPC>

NN _k	COMPONENTS				IMBALANCED				BALANCED			
	C	G	B	PLC	RESNET-18		WRN-28-10		RESNET-18		WRN-28-10	
					1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
					60.21±0.27	74.24±0.21	63.34±0.27	76.19±0.21	59.09±0.21	71.54±0.19	62.38±0.21	73.46±0.19
✓					63.95±0.27	81.15±0.17	67.14±0.27	83.40±0.16	63.82±0.22	80.47±0.15	67.22±0.21	82.58±0.16
✓	✓				68.57±0.28	82.69±0.16	71.22±0.26	84.74±0.16	73.43±0.23	84.37±0.14	75.94±0.22	86.55±0.13
✓	✓	✓			70.16±0.29	82.62±0.17	72.89±0.28	84.89±0.16	75.59±0.27	84.80±0.15	78.72±0.25	87.11±0.13
✓	✓		✓		69.11±0.29	82.97±0.16	71.64±0.28	85.16±0.15	74.85±0.25	84.66±0.14	77.70±0.23	86.91±0.13
✓	✓	✓	✓		70.24 ±0.29	82.71±0.17	73.22 ±0.29	85.00±0.16	76.06±0.28	84.82±0.15	79.37±0.26	87.12±0.13
✓	✓	✓	✓	✓	69.97±0.29	83.31 ±0.17	71.98±0.29	85.66 ±0.15	77.35 ±0.27	85.47 ±0.14	80.99 ±0.26	87.86 ±0.13

Table 1. Ablation study of algorithmic components of both balanced and imbalanced versions of our method AM on *miniImageNet*. NN_k: *k*-nearest neighbour graph; otherwise, complete graph. C: learnable class centroids. G: learnable pairwise scaling factors G (4). B: learnable adjacency matrix B (5). PLC: feature pre-processing as defined in subsection 4.1.

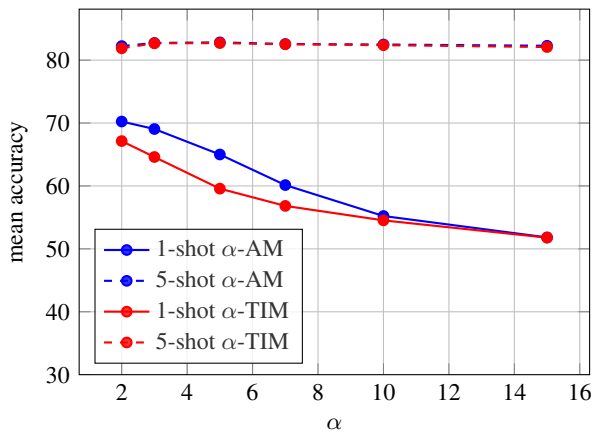


Figure 2. Effect of parameter α on α -AM and α -TIM, *miniImageNet* 1-shot and 5-shot.

of α has a lot more effect in the 1-shot than in the 5-shot setting. Also, α behaves similarly for both α -AM and α -TIM. Nevertheless, in the majority of the cases α -AM outperforms α -TIM. The optimal value of α is 2 for 1-shot and 5 for 5-shot. Therefore we set $\alpha = 2$ and $\alpha = 5$ for all 1-shot and 5-shot imbalanced experiments respectively unless stated otherwise.

Algorithmic components We ablate all components of our method under both the imbalanced (17) and balanced (14) settings in Table 1. As it can be seen from the first and second rows, using a *k*-nearest neighbour graph gives significant improvement over using a dense graph. Adapting the centroids, C, brings further substantial improvement. Adapting the centroids, C, along with either G or B brings further performance improvement. Adapting both manifold parameters G and B along with C provides better performance than just adapting either G or B in most experiments, especially in the balanced setting. Using PLC pre-processing yields further performance improvement except in the 1-shot imbalanced setting.

METHOD	<i>miniImageNet</i>		<i>tieredImageNet</i>	
	1-shot	5-shot	1-shot	5-shot
RESNET-18				
Entropy-min [7]	58.50	74.80	61.20	75.50
LR+ICI [40]	58.70	73.50	74.60	85.10
PT-MAP [12]	60.10	67.10	64.10	70.00
LaplacianShot [46]	65.40	81.60	72.30	85.70
BD-CSPN [21]	67.00	80.20	74.10	84.80
TIM [2]	67.30	79.80	74.10	84.10
α -TIM [38]	67.40	82.50	74.40	86.60
α -TIM _{PLC} * [38]	63.38	82.80	70.17	86.82
α -AM	70.24	82.71	77.28	86.97
α -AM _{PLC}	69.97	83.31	76.44	87.19
WRN-28-10				
Entropy-min [7]	60.40	76.20	62.90	77.30
PT-MAP [12]	60.60	66.80	65.10	71.00
LaplacianShot [46]	68.10	83.20	73.50	86.80
BD-CSPN [21]	70.40	82.30	75.40	85.90
TIM [2]	69.80	81.60	75.80	85.40
α -TIM [38]	69.80	84.80	76.00	87.80
α -TIM _{PLC} * [38]	66.50	85.12	71.97	88.28
α -AM	73.22	85.00	78.94	88.44
α -AM _{PLC}	71.98	85.66	78.75	88.69

Table 2. Imbalanced transductive inference on *miniImageNet* and *tieredImageNet*. Results as reported by [38]. *: Results were reproduced using the official code provided by [38].

4.3. Comparison of state of the art

Imbalanced transductive few-shot learning Table 2 and Table 3 show that our method achieves new state of the art performance using both ResNet-18 and WRN-28-10 on all three datasets and both 1-shot and 5-shot settings. Impressively, we improve the 1-shot state of the art in all cases significantly, by as much as 4.2% on CUB with ResNet-18. Even though we outperform α -TIM without PLC pre-processing in every experiment, PLC brings further improve-

METHOD	CUB	
	1-shot	5-shot
RESNET-18		
PT-MAP [12]	65.10	71.30
Entropy-min [7]	67.50	82.90
LaplacianShot [46]	73.70	87.70
BD-CSPN [21]	74.50	87.10
TIM [2]	74.80	86.90
α -TIM [38]	75.70	89.80
α -TIM _{PLC} * [38]	70.95	89.56
α -AM	79.92	89.83
α -AM _{PLC}	78.62	89.86

Table 3. *Imbalanced transductive inference* on CUB. Results as reported by [38]. *: Results were reproduced using the official code provided by [38].

ment in 5-shot, while not being beneficial in 1-shot. Interestingly, PLC pre-processing does not have the same effect on α -TIM, providing only marginal improvement in the 5-shot while being detrimental in the 1-shot.

Balanced transductive few-shot learning Tables 5 and 6 show that AM_{PLC} outperforms all other methods, with its closest competitor being PT-MAP [12]. Notably, our superiority is not due to pre-processing because PT-MAP also uses PLC. AM_{PLC} also significantly outperforms both versions of TIM. Interestingly, the performance of TIM always drops when PLC pre-processing is used, while AM always improves. Even without PLC, AM significantly outperforms TIM by 2 – 4% in 1-shot, while being on par or slightly worse by 0.1 – 0.3% in 5-shot.

Since the official code provided by [38] does not provide more recent methods in the balanced setting, such as [16] and [45], we use the publicly available code and pre-trained WRN-28-10 provided by [16] to compare AM with the state of the art methods: TIM, EASE+SIAMESE [45], PT+MAP [12] and iLPC [16]. Table 4 shows that AM_{PLC} outperforms all methods in the majority of the experiments. Again, our superiority is not due to pre-processing since we provided results for TIM and EASE+SIAMESE using PLC while PT-MAP and iLPC use PLC as part of their method.

4.4. Effect of unlabeled data

We investigate the effect of the quantity of unlabeled queries M , comparing α -AM against α -TIM. We do not use PLC pre-processing here, *which is beneficial to α -TIM*. Figure 3 shows that in the 1-shot setting, α -AM outperforms α -TIM significantly by as much as 3.7% when $M = 300$ and generally the performance gap increases as the number of unlabeled data increases. Figure 4 shows that also in the 5-shot setting, as the number of unlabeled queries increases, our algorithm outperforms α -TIM with an increasing per-

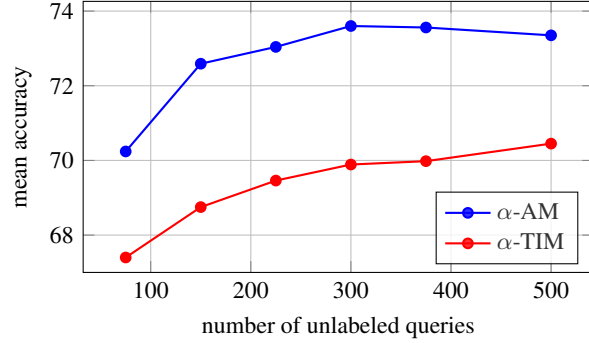


Figure 3. *Effect of number of unlabeled queries M on α -AM and α -TIM, miniImageNet 1-shot.*

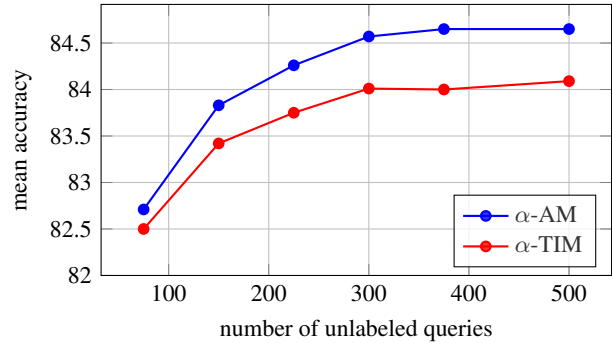


Figure 4. *Effect of number of unlabeled queries M on α -AM and α -TIM, miniImageNet 5-shot.*

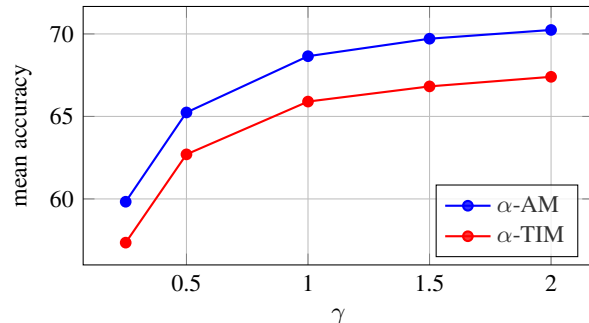


Figure 5. *Effect of class imbalance parameter γ in $Dir(\gamma)$ on α -AM and α -TIM, miniImageNet 1-shot. Class distributions are more imbalanced with lower γ .*

formance gap. This can be attributed to the fact that α -AM leverages the data manifold through the k -nearest neighbour graph while α -TIM works in Euclidean space.

4.5. Robustness against imbalance

We investigate the effect of increasing the class imbalance in Q by decreasing the value of γ used in $Dir(\gamma)$. Figure 5 shows that, while the performance of both α -AM and α -TIM drops as the classes become more imbalanced, α -AM consistently outperforms α -TIM.

METHOD	<i>mini</i> IMAGENET		<i>tiered</i> IMAGENET		CIFAR-FS		CUB	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
WRN-28-10								
PT+MAP [12]*	82.88 \pm 0.73	88.78 \pm 0.40	88.15 \pm 0.71	92.32 \pm 0.40	86.91 \pm 0.72	90.50 \pm 0.49	91.37 \pm 0.61	93.93 \pm 0.32
iLPC [16]*	83.05 \pm 0.79	88.82 \pm 0.42	88.50 \pm 0.75	92.46 \pm 0.42	86.51 \pm 0.75	90.60 \pm 0.48	91.03 \pm 0.63	94.11 \pm 0.30
EASE+SIAMESE [45] [†]	83.44 \pm 0.77	88.66 \pm 0.43	88.69 \pm 0.73	92.47 \pm 0.41	86.71 \pm 0.77	90.28 \pm 0.51	91.44 \pm 0.63	93.85 \pm 0.32
EASE+SIAMESE _{PLC} [45] [†]	82.13 \pm 0.81	87.34 \pm 0.46	88.42 \pm 0.73	92.19 \pm 0.41	86.74 \pm 0.78	90.22 \pm 0.51	91.49 \pm 0.63	93.32 \pm 0.32
TIM [2]	77.65 \pm 0.72	88.21 \pm 0.40	83.88 \pm 0.74	91.89 \pm 0.41	82.63 \pm 0.70	90.28 \pm 0.46	87.50 \pm 0.62	93.59 \pm 0.30
TIM _{PLC} [2]	75.77 \pm 0.67	88.37 \pm 0.40	83.22 \pm 0.70	92.13 \pm 0.40	80.52 \pm 0.70	90.25 \pm 0.46	85.58 \pm 0.61	93.48 \pm 0.31
AM	80.74 \pm 0.81	87.75 \pm 0.42	86.38 \pm 0.78	91.85 \pm 0.85	85.93 \pm 0.74	90.13 \pm 0.47	90.24 \pm 0.65	93.43 \pm 0.30
AM _{PLC}	83.40 \pm 0.74	89.08 \pm 0.40	88.31 \pm 0.73	92.60 \pm 0.39	86.91 \pm 0.74	90.80 \pm 0.46	91.32 \pm 0.60	94.14 \pm 0.29

Table 4. *Balanced transductive inference state of the art*. Results were reproduced using the official code provided by [16]. *: Results as reported by [16]. †: Our reproduction with official code from [45].

METHOD	<i>mini</i> IMAGENET		<i>tiered</i> IMAGENET	
	1-shot	5-shot	1-shot	5-shot
RESNET-18				
PT-MAP [12]	76.88	85.18	82.89	88.64
LaplacianShot [46]	70.24	82.10	77.28	86.22
BD-CSPN [21]	69.36	82.06	76.36	86.18
TIM [2]	73.81	84.91	80.13	88.61
TIM _{PLC} [2]	69.33	84.53	76.36	88.33
AM	76.06	84.82	82.42	88.61
AM _{PLC}	77.35	85.47	83.40	89.07
WRN-28-10				
PT-MAP [12]	80.35	87.37	84.84	89.86
LaplacianShot [46]	72.91	83.85	78.85	87.27
BD-CSPN [21]	72.16	83.78	77.88	87.23
TIM [2]	77.78	87.43	82.28	89.84
TIM _{PLC} [2]	73.52	86.95	78.23	89.56
AM	79.37	87.12	84.07	89.69
AM _{PLC}	80.99	87.86	85.26	90.30

Table 5. *Balanced transductive inference on miniImageNet and tieredImageNet*. All results were reproduced using the official code provided by [38].

METHOD	CUB	
	1-shot	5-shot
PT-MAP [12]	86.05	91.28
LaplacianShot [46]	79.55	88.96
BD-CSPN [21]	78.52	89.02
TIM [2]	82.87	91.58
TIM _{PLC} [2]	77.69	91.17
AM	85.59	91.24
AM _{PLC}	86.64	91.78

Table 6. *Balanced transductive inference on CUB*. All results were reproduced using the official code provided by [38].

METHOD	<i>mini</i> IMAGENET		<i>tiered</i> IMAGENET	
	1-shot	5-shot	1-shot	5-shot
DENSENET-121				
α -TIM [38]	70.41	85.58	76.55	88.33
α -TIM _{PLC} [38]	67.56	86.26	74.56	88.68
α -AM	73.67	85.47	79.95	89.34
α -AM _{PLC}	73.98	86.76	79.99	89.73

Table 7. *Imbalanced transductive inference on miniImageNet and tieredImageNet* using the DenseNet-121 backbone. All results were reproduced using the official code provided by [38].

4.6. Other backbones

Table 7 shows that by using the DenseNet-121 backbone, AM_{PLC} outperforms α -TIM and α -TIM_{PLC} in both 1-shot and 5-shot settings. As in the previous experiments, we observe a significant performance gap of roughly 3.5% in the 1-shot setting.

5. Conclusion

In this work we propose a novel method named as *Adaptive Manifold*, AM, that achieves new state of the art performance in imbalanced transductive few-shot learning, while outperforming several state of the art methods in the traditional balanced transductive few-shot learning. AM combines the complementary strengths of K-means-like iterative class centroid updates and exploiting the underlying data manifold through label propagation. Leveraging manifold class similarities to measure class probabilities for the unlabeled query examples and optimizing the manifold parameters through the loss function proposed by [38], we achieve a new state of the art performance in the imbalanced setting on different datasets using multiple backbones, outperforming previous methods by a large margin, especially in the 1-shot setting. The robustness of our method is validated by our findings that it can be combined effectively with PLC

pre-processing and that it can outperform its competitors in other settings such as with more unlabeled query examples and as well as the standard balanced few-shot setting.

References

- [1] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018. 1, 2
- [2] Malik Boudiaf, Imtiaz Ziko, Jérôme Rony, Jose Dolz, Pablo Piantanida, and Ismail Ben Ayed. Information maximization for few-shot learning. In H. Larochelle, M. Ranzato, R. Hassell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, 2020. 1, 2, 3, 4, 5, 6, 7, 8
- [3] Da Chen, Yuefeng Chen, Yuhong Li, Feng Mao, Yuan He, and Hui Xue. Self-supervised learning for few-shot image classification. *arXiv preprint arXiv:1911.06045*, 2019. 5
- [4] Mengting Chen, Yuxin Fang, Xinggang Wang, Heng Luo, Yifeng Geng, Xinyu Zhang, Chang Huang, Wenyu Liu, and Bo Wang. Diversity transfer network for few-shot learning. In *AAAI*, 2020. 2
- [5] 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. 2, 5
- [6] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. In *CVPR*, 2019. 2
- [7] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *ICLR*, 2020. 6, 7
- [8] Li Fei-Fei. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006. 2
- [9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 1, 2
- [10] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 2
- [11] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016. 2
- [12] Yuqing Hu, Vincent Gripon, and Stéphane Pateux. Leveraging the feature distribution in transfer-based few-shot learning. *arXiv preprint arXiv:2006.03806*, 2020. 1, 2, 5, 6, 7, 8
- [13] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *CVPR*, 2019. 1, 3
- [14] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML workshop*, 2015. 1, 2
- [15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009. 5
- [16] Michalis Lazarou, Tania Stathaki, and Yannis Avrithis. Iterative label cleaning for transductive and semi-supervised few-shot learning. In *ICCV*, 2021. 1, 2, 3, 5, 7, 8
- [17] Michalis Lazarou, Tania Stathaki, and Yannis Avrithis. Tensor feature hallucination for few-shot learning. In *WACV*, January 2022. 1, 2
- [18] Kai Li, Yulun Zhang, Kunpeng Li, and Yun Fu. Adversarial feature hallucination networks for few-shot learning. In *CVPR*, 2020. 1, 2
- [19] Yann Lifchitz, Yannis Avrithis, and Sylvaine Picard. Local propagation for few-shot learning. In *ICPR*. IEEE, 2021. 2
- [20] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc. Dense classification and implanting for few-shot learning. In *CVPR*, 2019. 1
- [21] Jinlu Liu, Liang Song, and Yongqiang Qin. Prototype rectification for few-shot learning. In *European Conference on Computer Vision*, 2019. 1, 2, 6, 7, 8
- [22] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *CVPR*, 2019. 2
- [23] Qinxuan Luo, Lingfeng Wang, Jingguo Lv, Shiming Xiang, and Chunhong Pan. Few-shot learning via feature hallucination with variational inference. In *WACV*, 2021. 1, 2
- [24] 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. 1, 2
- [25] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017. 2
- [26] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *ICML*, 2017. 2
- [27] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 2
- [28] 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. 5
- [29] Guodong Qi, Huimin Yu, Zhaohui Lu, and Shuzhao Li. Transductive few-shot classification on the oblique manifold. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8392–8402, 2021. 1, 2
- [30] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *NeurIPS*, 2019. 2
- [31] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016. 2
- [32] Pau Rodríguez, Issam Laradji, Alexandre Drouin, and Alexandre Lacoste. Embedding propagation: Smoother manifold for few-shot classification. *ECCV*, 2020. 2, 3
- [33] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018. 2
- [34] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016. 2
- [35] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 1, 2, 3
- [36] 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. 2

- [37] 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](#)
- [38] Olivier Veilleux, Malik Boudiaf, Pablo Piantanida, and Ismail Ben Ayed. Realistic evaluation of transductive few-shot learning. *Advances in Neural Information Processing Systems*, 34:9290–9302, 2021. [1](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [39] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, 2016. [1](#), [2](#), [5](#)
- [40] Yikai Wang, C. Xu, Chen Liu, Liyong Zhang, and Yanwei Fu. Instance credibility inference for few-shot learning. *CVPR*, 2020. [1](#), [2](#), [6](#)
- [41] Zhanyuan Yang, Jinghua Wang, and Yingying Zhu. Few-shot classification with contrastive learning. In *ECCV*. Springer, 2022. [2](#)
- [42] Hongguang Zhang, Jing Zhang, and Piotr Koniusz. Few-shot learning via saliency-guided hallucination of samples. In *CVPR*, 2019. [2](#)
- [43] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. MetaGAN: An adversarial approach to few-shot learning. *NeurIPS*, 2018. [2](#)
- [44] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NIPS*, 2003. [3](#)
- [45] Hao Zhu and Piotr Koniusz. Ease: Unsupervised discriminant subspace learning for transductive few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9078–9088, 2022. [1](#), [5](#), [7](#), [8](#)
- [46] Imtiaz Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed. Laplacian regularized few-shot learning. In *International conference on machine learning*, pages 11660–11670. PMLR, 2020. [6](#), [7](#), [8](#)