Local Propagation for Few-Shot Learning

Yann Lifchitz
Inria, Univ Rennes, CNRS, IRISA
Safran

Yannis Avrithis
Inria, Univ Rennes, CNRS, IRISA

Sylvaine Picard
Safran

Abstract—The challenge in few-shot learning is that available data is not enough to capture the underlying distribution. To mitigate this, two emerging directions are (a) using local image representations, essentially multiplying the amount of data by a constant factor, and (b) using more unlabeled data, for instance by transductive inference, jointly on a number of queries. In this work, we bring these two ideas together, introducing *local propagation*. We treat local image features as independent examples, we build a graph on them and we use it to propagate both the features themselves and the labels, known and unknown. Interestingly, since there is a number of features per image, even a single query gives rise to transductive inference. As a result, we provide a universally safe choice for few-shot inference under both non-transductive and transductive settings, improving accuracy over corresponding methods. This is in contrast to existing solutions, where one needs to choose the method depending on the quantity of available data.

I. INTRODUCTION

Few-shot learning [1]–[3] is the problem of learning new tasks from few examples, possibly transferring knowledge from previous tasks. Against the mainstream paradigm of having lots of labeled data in deep learning, it limits not only the amount of supervision but also the amount of data. Given the variability of appearance in few-shot image classification benchmarks, learning from few examples without knowledge of the underlying distribution is truly challenging.

Few-shot learning has been little studied before deep learning [4]. Research on few-shot learning is recently becoming very popular, but is not very mature. On one hand, it is often connected to *meta-learning* [5] in the sense of learning to compare [1] or to optimize [6], giving rise to complex ideas involving second-order derivatives. On the other hand, it boils down to *representation learning*, using *e.g.* metric learning [7] or parametric classifiers [8], followed by nearest neighbor classifiers at inference [9].

While there are several approaches on *generating* more data [3], [10], global spatial pooling into compact image representations ignores the rich data that is hidden in each given example. Each image is inherently a collection of data, which has been exploited by *dense classification* (DC) at representation learning [11] and *naïve Bayes nearest neighbor* (NBNN) [12] at inference.

Using *unlabeled data* is another popular direction of research, leveraging existing results from transductive inference [13], semi-supervised learning [14] and self-supervised representation learning [15]. *Graph-based methods* are at the core of this effort, using for instance label propagation [13], feature propagation [16] and graph neural networks [14].
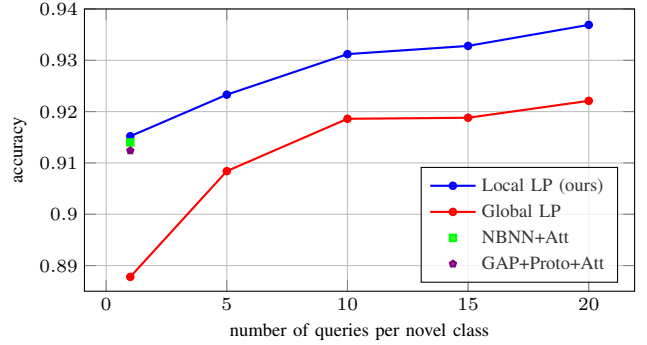

Fig. 1. CUB 5-way 5-shot classification accuracy *vs.* number of queries per novel class. Our local label propagation outperforms transductive and non-transductive baselines in all settings. By contrast, global label propagation only competes with non-trasductive methods when at least 10 unlabelled queries are available. We use spatial attention (also our contribution) and feature propagation [16] for all methods. We use feature pooling for local propagation. Please see section VI for more details.

This work is an attempt to bridge these two ideas, *i.e.*, local representations [12] and propagation [16], into a common framework. Essentially, NBNN [12] measures the *average similarity* of local representations of a given image to local representations of all images in a class; while feature or label propagation [16] replaces raw (Euclidean) similarities with similarities taking into account the manifold structure of the data distribution, and measures a single *manifold similarity* of a given image to a class. Our *local propagation* combines both by measuring the *average manifold similarity* of local representations of a given image to local representations of all images in a class.

Concretely, we learn a representation using DC [11] and we apply local propagation at inference, without meta-learning: We break down the convolutional activations of support and query images into pieces corresponding to different spatial positions, consider all these pieces as different examples, and then apply feature or label propagation [16] to these examples. Pieces originating in support examples inherit their labels as in DC [11] and NBNN [12], while pieces originating in queries are unlabeled. Since there are a number of unlabeled pieces per image, this gives rise to *transductive inference* even in the case of a single query image. As shown in Figure 1, this means that our method is a *universally safe choice* regardless of the amount of available unlabeled data.

In summary, we make the following contributions:
- We study graph-based propagation on *local* (pixel) and

semi-local (clusters) representations across of images for *feature and label propagation* for the first time.

- We apply this idea to few-shot learning, effectively *generating* more data and *propagating* through it, bringing even the case of single queries into *transductive inference*.
- We show that an extremely simple *spatial attention* mechanism is not only essential in our local propagation, but also brings significant gains in all baselines.
- We show consistent gains in most datasets and settings, including transductive and non-transductive.

II. RELATED WORK

Few shot learning. While studies before deep learning have been scarce [4], [17], [18], few-shot learning has become a very popular problem beginning mainly from *matching networks* [1] and *prototypical networks* [2]. Seen as a *meta-learning* problem of learning to compare in episodes, these solutions amount essentially to *metric learning*, and indeed such methods have been revisited in the context of few-shot learning [7], [19]. Simpler methods have highlighted the importance of *representation learning*. These include for instance nearest-neighbor classifiers without meta-learning [9] and simple variants of supervised classifiers like a *cosine classifier* [8], [20], [21].

More data. Since the main challenge in few-shot learning is the lack of data, several approaches focus on finding more. These include *augmentation* in the feature space [22] or by combining spatial elements of images [23], [24], *generation* in the feature space [3] or images [25], [26], *image-to-image translation* [10], [27], using *base-class data* [28], [29], or even *true additional data*, unlabeled [30] or weakly labeled [31]. By contrast, we generate more data "for free" by just looking more carefully within the existing data.

Transductive inference. Another possibility is to consider multiple queries jointly and exploit their distribution, even though they are unlabeled. This gives rise to transductive inference [13], [32], [33]. Most well-known are *transductive propagation networks* (TPN) [13], which use *label propagation* [34] in a meta-learning setting. Recently, this direction is becoming very popular [16], [35]–[40]. Most related to our work is the very recent *embedding propagation* [16], which propagates the features as well as the labels. We do the same without meta-learning and most importantly, all propagation is *local*.

Semi/self-supervised learning. Using unlabeled data in an *inductive setting* gives rise to *semi-supervised learning* by using pseudo-labels [41], graph-based methods [14], or feature-space augmentation [22], [42]. It is also common to use auxiliary unsupervised objectives like *rotation* [15], [22]. While we do not address an inductive setting, our work is a direct extension of graph-based methods, hence it can be applied to an inductive setting too, much like label propagation itself [43], or combined with any other objective.

Attention. It is common to use attention and adaptation mechanisms in the feature space [1], [21], [44]–[47]. However,

despite being the subject of a pioneering work in 2005 [4], looking at local information in images has not been studied more recently in few-shot learning, until *dense classification* (DC) [11] and *naïve Bayes nearest neighbor* (NBNN) [12]. We use the former for representation learning. The latter is similar to our work in using local representations at inference, the difference being that we apply propagation. These works have been followed by studies on *spatial attention* [48]–[51] and *alignment* [52]–[55]. We experiment with an extremely simple spatial attention mechanism in this work, which requires no learning and boosts significantly all baselines.

Local propagation. Whatever is propagated (similarities, features, or labels), there are two extremes in graph-based propagation. At one extreme, vertices are *global* representations of images, and the graph represents a dataset. This can be used *e.g.* for similarity search [56] or semi-supervised classification [34], [57]. At the other extreme, vertices are *local* representations of pixels in an image, which can be used *e.g.* for interactive [58], [59] or semantic [60] segmentation, or both [61]. *Regional representations* across images have been used for similarity search [62], but we believe we are the first to use *local* (pixel) or *semi-local* (clusters) representations across images for feature or label propagation.

III. PRELIMINARIES

Problem. A few-shot classification task comprises a dataset $D := \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$ of *support examples* $\mathbf{x}_i \in \mathcal{X}$ and *labels* $\mathbf{y}_i$, where $\mathcal{X}$ is an input space. Each label is represented by a *one-hot* vector $\mathbf{y}_i := \mathbf{e}_{\ell_i}$, where $\{\mathbf{e}_j\}_{j=1}^{c}$ is the standard basis of $\mathbb{R}^c$, $\ell_i \in C$ is a *label index* and $C := [c] := \{1, \ldots, c\}$ is a set of *novel (unseen) classes*. The number n of support examples is assumed to be small. The most common setting is s examples per novel class, with *e.g.* $s \in \{1, 5, 10\}$, so that $n = cs$, referred to as *c-way*, *s-shot* classification. The objective of the task is to learn a classifier $f : \mathcal{X} \mapsto \mathbb{R}_+^c$ on the support data D. This classifier maps a new *query example* $\mathbf{x}$ from $\mathcal{X}$ to a probability distribution $\mathbf{p} = f(\mathbf{x})$. A discrete *prediction* $\pi(\mathbf{p})$ in C follows, where

$$\pi(\mathbf{p}) := \arg\max_{j \in [c]} p_j \tag{1}$$

is the class of of maximum probability and p_j is the j-th element of $\mathbf{p}$.

Before we are presented with few-shot classification tasks, we are given a dataset $\dot{D} := \{(\dot{\mathbf{x}}_i, \dot{\mathbf{y}}_i)\}_{i=1}^{\dot{n}}$ of *training examples* $\dot{\mathbf{x}}_i \in \mathcal{X}$ and *labels* $\dot{\mathbf{y}}_i := \mathbf{e}_{\dot{\ell}_i}$, where $\dot{\ell}_i \in \dot{C}$ and $\dot{C} := [\dot{c}]$ is a set of *base classes*, disjoint from C. The number $\dot{n}$ of training examples is assumed to be large enough to learn a *representation* of data in $\mathcal{X}$, or otherwise accumulate knowledge that facilitates solving new tasks. We call this process *base training*. The problem of few-shot classification amounts to designing both the base training process given $\dot{D}$ and how to solve new tasks given D.

Transductive setting. It is possible that in each few-shot classification task we are given a *set* $Q := \{\mathbf{q}_i\}_{i=1}^{q}$ of query examples and a prediction is required for all queries in Q.

In this case, although query examples are unlabeled, we can take advantage of this additional data and learn a classifier f that is a function of both the labeled support data D and the unlabeled queries Q. This transductive setting implies *semi-supervised learning*.

Representation. The classifier is built on top of an *embedding function* $\phi_\theta : \mathcal{X} \to \mathbb{R}^{r \times d}$, with parameters θ that are learned at base training. Given an example $\mathbf{x} \in \mathcal{X}$, this function yields a $r \times d$ *feature tensor* $\phi_\theta(\mathbf{x})$, where r represents the dimensions of a spatial domain Ω and d the feature dimensions. For $\mathcal{X}$ comprising 2D images, the feature is a $w \times h \times d$ tensor that is the activation of the last convolutional layer, $r := w \times h$ is the spatial resolution and $\Omega := [w] \times [h]$ is the spatial domain. The feature can still be a vector in $\mathbb{R}^d$ in the special case $r = 1$, *e.g.* using *global spatial pooling*. The feature tensor $F := \phi_\theta(\mathbf{x})$ contains a feature vector $F(\mathbf{r}) = \phi_\theta(\mathbf{x})(\mathbf{r}) \in \mathbb{R}^d$ for each spatial position $\mathbf{r} \in \Omega$.

IV. Background

Cosine classifier. Initially used in face verification [63], [64], a simple form of base training that was introduced to few-shot learning independently by Qi *et al.* [20] and Gidaris and Komodakis [21], is to learn a parametric linear classifier that consists of a fully-connected layer without bias on top of the embedding function ϕ_θ, followed by softmax and cross-entropy. If $W := (\dot{\mathbf{w}}_j)_{j=1}^{\dot{c}}$ is the collection of class weights with $\dot{\mathbf{w}}_j \in \mathbb{R}^{r \times d}$, the classifier is defined by

$$f_{\theta,W}(\mathbf{x}) := \boldsymbol{\sigma} \left(\rho [\cos(\phi_\theta(\mathbf{x}), \dot{\mathbf{w}}_j)]_{j=1}^{\dot{c}} \right), \tag{2}$$

for $\mathbf{x} \in \mathcal{X}$, where cos is *cosine similarity*, $\rho \in \mathbb{R}^+$ is a trainable *scale parameter* and $\boldsymbol{\sigma} : \mathbb{R}^m \to \mathbb{R}_+^m$ is the *softmax function* $\boldsymbol{\sigma}(\mathbf{a}) := (e^{a_1}, \dots, e^{a_{\dot{c}}})/\sum_j e^{a_j}$ for $\mathbf{a} \in \mathbb{R}^{\dot{c}}$. The representations (features and class weights) either retain resolution $r > 1$ and are *flattened* to vectors of length rd, or are *pooled* to vectors of length d ($r = 1$), by global spatial pooling. Base training amounts to minimizing the *cost function*

$$J(\dot{D}; \theta, W) := \sum_{i=1}^n \ell(f_{\theta,W}(\dot{\mathbf{x}}_i), \dot{\mathbf{y}}_i) \tag{3}$$

over θ, W, where $\ell(\mathbf{p}, \mathbf{y}) := -\log \langle \mathbf{y}, \mathbf{p} \rangle$ for $\mathbf{y} \in \{0,1\}^{\dot{c}}$ and $\mathbf{p} \in \mathbb{R}_+^{\dot{c}}$, is the *cross-entropy* loss.

Prototypes. A popular classifier for few-shot classification tasks is the *prototype classifier*, introduced by Snell *et al.* [2]. If $\mathcal{I}_j := \{i \in [n] : \mathbf{y}_i = \mathbf{e}_j\}$ denotes the indices of support examples labeled in class j, then the *prototype* of this class j is given by the average features

$$\boldsymbol{\mu}_j := \frac{1}{|\mathcal{I}_j|} \sum_{i \in \mathcal{I}_j} \phi_\theta(\mathbf{x}_i) \tag{4}$$

of those examples for $j \in C$. Again, features are either flattened or pooled to vectors first. Then, denoting by $M := (\boldsymbol{\mu}_j)_{j=1}^c$ the collection of prototypes, a query $\mathbf{x} \in \mathcal{X}$ is mapped to $f_{\theta,M}(\mathbf{x})$, as defined by (2).

Naïve Bayes nearest neighbor (NBNN). In the revival [12] of the classic image-to-class approach [65], one collects, for each class $j \in [c]$, the features $V_j := \{\phi_\theta(\mathbf{x}_i)(\mathbf{r})\}_{i \in \mathcal{I}_j, \mathbf{r} \in \Omega}$ of all spatial positions of all support examples labeled in class j. Then, given a query $\mathbf{x} \in \mathcal{X}$ with feature tensor $F := \phi_\theta(\mathbf{x})$, for each class j, a score

$$s_j(F) := \sum_{\mathbf{r} \in \Omega} \sum_{\mathbf{v} \in \text{NN}_{V_j}(F(\mathbf{r}))} \cos(F(\mathbf{r}), \mathbf{v}) \tag{5}$$

is defined as the average cosine similarity over the features $F(\mathbf{r})$ at all spatial positions $\mathbf{r} \in \Omega$ and their k-nearest neighbors $\text{NN}_{V_j}(F(\mathbf{r}))$ in V_j. Then, the prediction for $\mathbf{x}$ is the class of maximum score.

V. Local Propagation

A. Base training

Dense classifier. We use a *dense classifier* for base training, introduced by Lifchitz *et al.* [11]. Rather than flattening or pooling, the classifier $f_{\theta,W} : \mathcal{X} \to \mathbb{R}_+^{r \times \dot{c}}$ maps an example $\mathbf{x}$ to a tensor of probabilities over spatial positions, by applying a cosine classifier (2) densely at each position

$$f_{\theta,W}(\mathbf{x}) := \left[\boldsymbol{\sigma} \left(\rho [\cos(\phi_\theta(\mathbf{x})(\mathbf{r}), \dot{\mathbf{w}}_j)]_{j=1}^{\dot{c}} \right) \right]_{\mathbf{r} \in \Omega}, \tag{6}$$

where the class weights W are shared over locations with $\dot{\mathbf{w}}_j \in \mathbb{R}^d$. Cross-entropy applies to all spatial positions using the same class label, and cost function (3) becomes

$$J(\dot{D}; \theta, W) := \sum_{i=1}^n \sum_{\mathbf{r} \in \Omega} \ell(f_{\theta,W}(\dot{\mathbf{x}}_i)(\mathbf{r}), \dot{\mathbf{y}}_i). \tag{7}$$

Local spatial pooling. Dense classification avoids global spatial pooling by going to the other extreme of applying the loss to every position. This happens regardless of whether the effective receptive field is large enough to represent the class at hand, so it assumes an appropriate resolution of the feature tensor. However, it has been observed that it helps to use input images of higher resolution than the standard benchmarks [66], which we follow. This results in features of accordingly higher resolution, where each position corresponds to small details. We solve this by applying *local spatial pooling* on the feature tensor, both before dense classification at base training as well as at new classification tasks.

B. Few-shot classification

Spatial attention. Before we can use features of all spatial positions as data, it is important to suppress the background, which appears frequently across positions and images, without being discriminative for the classification task. There are different approaches, such as learning a class-agnostic *spatial attention* mechanism [48], [49] or simply by a form of pooling over feature channels [67]. We follow the latter approach. In particular, given an example $\mathbf{x} \in \mathcal{X}$ with feature tensor $F := \phi_\theta(\mathbf{x})$, we select a subset of feature vectors $a(F) \subset \mathbb{R}^d$ at spatial positions $\mathbf{r} \in \Omega$ where the ℓ_2-norm is at least $\tau > 0$ relative to the maximum over the domain:

$$a(F) := \{F(\mathbf{r}) : \|F(\mathbf{r})\| \geq \tau \max_{\mathbf{t} \in \Omega} \|F(\mathbf{t})\|, \mathbf{r} \in \Omega\}. \tag{8}$$
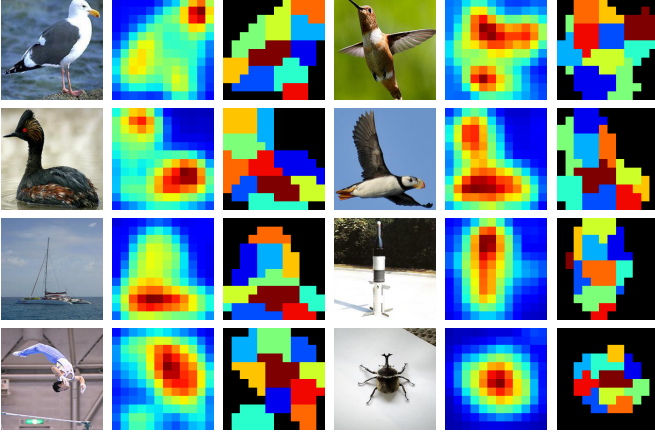
Fig. 2. Examples of images, each with the corresponding spatial attention heatmap and clusters used in feature pooling (black indicates regions below threshold in the heatmap). The first two lines correspond to CUB, the last two to *mini*ImageNet. We use $\tau = 0.3$ for spatial attention and $m = 10$ for feature pooling.
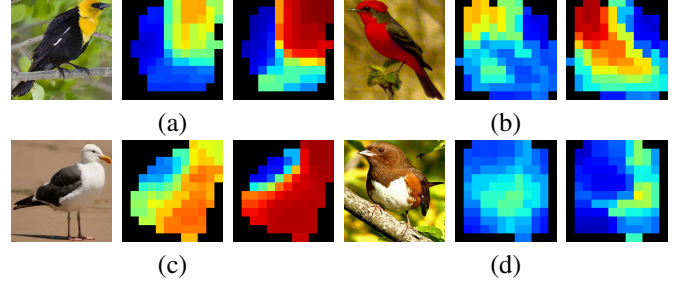

Fig. 3. Examples of CUB query images in 5-way 5-shot non-transductive tasks, each followed by the heatmap of predicted probability for the correct class using a prototype classifier, then using local label propagation. (a), (b) Local label propagation helps classifying to the correct class. (c) Both give a correct prediction. (d) Local label propagation fails.

Examples are shown in Figure 2. We find this mechanism particularly effective for its simplicity, not only for our method, but also for all baselines. No spatial attention is a special case where $\tau = 0$.

Feature pooling. Propagation tends to amplify elements that appear frequently in a dataset. Local propagation does the same for elements originating from different spatial positions, which in turn depends on the scale of objects relative to the spatial resolution. This can be particularly harmful with elements originating from background clutter and bypass condition (8), exactly because they appear frequently.

To obtain a fixed-size representation that only depends on the content, we perform pooling in the feature space into a fixed number of vectors per example. We do so by *clustering*: given an example $\mathbf{x} \in \mathcal{X}$ with selected feature vectors $a(\phi_\theta(\mathbf{x}))$ (8), we obtain m clusters by k-means. We represent the corresponding m *feature centroids* as columns in the $d \times m$ matrix $g_\theta(\mathbf{x})$. Examples are shown in Figure 2. We use this representation only for local propagation. Global propagation and no feature pooling are special cases where $m = 1$ and $m = w \times h$ respectively.

Local propagation. We develop this idea under the transductive setting because it is more general: The non-transductive is the special case where $q = 1$, the set of queries $Q = \{\mathbf{q}_1\}$ is singleton and we are making a prediction for $\mathbf{q}_1$. Given the support examples in D and queries Q, we represent the feature centroids of both as columns in the $d \times t$ matrix

$$V := \begin{pmatrix} g_\theta(\mathbf{x}_1) & \ldots & g_\theta(\mathbf{x}_n) & g_\theta(\mathbf{q}_1) & \ldots & g_\theta(\mathbf{q}_q) \end{pmatrix} \quad (9)$$

where $t := (n + q)m$. Following [62], we use the pairwise similarity function $s(\mathbf{v}_1, \mathbf{v}_2) := [\cos(\mathbf{v}_1, \mathbf{v}_2)]_+^\gamma$ where $\gamma > 1$, and construct the reciprocal *k-nearest neighbor graph* of the columns of V, represented by the $t \times t$ symmetric nonnegative *adjacency matrix* W_V with zero diagonal. Following [34], this matrix is symmetrically normalized as

$\mathcal{W}_V := D_V^{-1/2} W_V D_V^{-1/2}$, where $D_V := W_V \mathbf{1}_t$ is the *degree matrix* of the graph and $\mathbf{1}_t$ is the $t \times 1$ all-ones vector.

Extending [34], given any matrix $A \in \mathbb{R}^{u \times t}$ (or row vector for $u = 1$), its *propagation* on V is defined as

$$p_V(A) := A(1 - \alpha)(I - \alpha \mathcal{W}_V)^{-1}. \quad (10)$$

This is a smoothing operation on the graph of V, where parameter $\alpha \in [0, 1)$ controls the amount of smoothing: Columns of A corresponding to similar columns of V are averaged together. It is infinitely-recursive, as revealed by the series expansion of the matrix inverse [34].

The operation (10) is called *local propagation* because the graph is defined on local representations originating from different spatial positions of the given images. *Global propagation* is the special case of having $m = 1$ cluster per image. This is the same as *global average pooling* (GAP), with or without spatial attention.

Local feature propagation. Using $A = V$, $u = d$ in (10), *local feature propagation* amounts to propagating V on itself:

$$\widetilde{V} := p_V(V), \quad (11)$$

in the sense that similar feature vectors in columns of V are averaged together, becoming even more similar. No feature propagation is a special case where $\alpha = 0$, $\widetilde{V} = V$.

Local label propagation. Given the propagated features $\widetilde{V}$ (11), we form a new graph with normalized adjacency matrix $\mathcal{W}_{\widetilde{V}}$. Extending [34], we form the $c \times t$ zero-one *label matrix* Y with one row per class and one column per spatial position over support examples and queries. A column corresponding to a spatial position of a support example $\mathbf{x}_i$ is defined as the one-hot label vector $\mathbf{y}_i$; a column corresponding to a position of a query $\mathbf{q}_i$ is zero:

$$Y := \begin{pmatrix} \mathbf{y}_1 \mathbf{1}_m^\top & \ldots & \mathbf{y}_n \mathbf{1}_m^\top & \mathbf{0}_{c \times m} & \ldots & \mathbf{0}_{c \times m} \end{pmatrix} \quad (12)$$

where $\mathbf{0}_{c \times m}$ is the $c \times m$ zero matrix and there are q such matrices. Using $A = Y$, $u = c$ in (10), *local label propagation* then amounts to propagating Y on $\widetilde{V}$:

$$\widetilde{Y} := p_{\widetilde{V}}(Y), \quad (13)$$

such that spatial positions with similar feature vectors obtain similar class scores. This may make little difference on labeled

(support) examples, but is a mechanism for spatial positions of *unlabeled* (query) examples to obtain label information as propagated from spatial positions of labeled examples with similar features.

Inference. In $c \times t$ matrix $\widetilde{Y}$ (13), there is one row per class and one column per spatial position over support examples and queries. $\widetilde{Y}$ is nonnegative; by column-wise ℓ_1-normalizing it into $c \times t$ matrix $\hat{Y}$, we can interpret columns as probability distributions over classes per position. For each query example $\mathbf{q}_i$, if $\hat{Y}_i$ is the corresponding $c \times m$ submatrix of $\hat{Y}$, we average these distributions over positions, obtaining a distribution $\mathbf{p}_i := \hat{Y}_i \mathbf{1}_m / m$. Finally, as in (1), we make a discrete prediction $\pi(\mathbf{p}_i) = \arg\max_{j \in [c]} p_{ij}$ as the class of maximum probability. This operation is similar to NBNN (5), but the quantities being averaged have undergone propagation rather than being direct similarities. Figure 3 shows examples of predicted probability for the correct class per spatial location. Local label propagation results in spatially smooth predictions that covers a large portion of the object.

VI. EXPERIMENTS

A. Experimental setup

Datasets. We evaluate our method on two datasets that are common in few-shot learning. The first, *Mini*ImageNet, is a subset of ImageNet ILSVRC-12 [68]. It contains 600 images for each of its 100 classes. Following the work of Ravi and Larochelle [69], we use 64 classes for base training, 16 classes for validation and 20 classes for test. We resample all images to 224×224, similarily to [8], [66]. The second dataset, CUB-200-2011 [70], referred to as CUB below, was introduced to few-shot learning by Hilliard *et al.* [71]. It contains 11,788 images from 200 distinct bird species. Following the splits of Ye *et al.* [72], we use 100 classes for base training, 50 for validation and 50 for testing. We crop images using bounding box annotations and resample them to 224×224.

Network. We test our method on a ResNet-12 embedding network. Introduced in [46], this network is now commonly used in the few-shot learning community. With input images of size 224×224, the embedding features are tensors of resolution 14×14. To adapt the the larger images before applying a dense classifier [11], we apply average pooling on these feature tensors, with kernel size 3×3 and stride 1 without padding. The resulting tensors are of resolution 12×12.

Base Training. We train the network from scratch using stochastic gradient descent with Nesterov momentum on mini-batches of size 32. The learning rate schedule is set according to the 5-way 5-shot validation accuracy.

Evaluation protocol. For each dataset, we obtain a unique embedding network resulting from base training. All methods are then applied to the same features. For all experiments, we sample 2000 5-way few-shot tasks from the test set, each with 15 queries per class. We report average accuracy as well as 95% confidence interval. We evaluate two different settings: In the *non-transductive* setting, queries are treated as 75 distinct
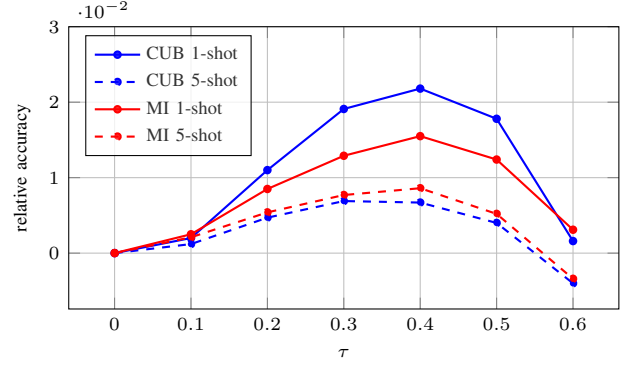

Fig. 4. *Spatial attention* on GAP+Proto [2]: 5-way few-shot classification accuracy *vs.* threshold τ, relative to $\tau = 0$ (no attention).

sets Q with only one query each, whereas in the *transductive* setting, there is a single set Q with all 75 queries.

Baselines. In the *non-transductive* setting, we compare our method with variants of four existing few-shot inference methods. The first, referred to as GAP+Proto, applies *global average pooling* (GAP) on feature tensors and then uses a prototype classifier [2] on the support set (4). The second is the inference mechanism of the matching network [1], while the third, referred to as Local Match, is a modified version as follows. For each support example $\mathbf{x}$ with feature tensor $F := \phi_\theta(\mathbf{x})$, we use local feature vectors $F(\mathbf{r})$ at all positions $\mathbf{r} \in \Omega$ as independent support examples, with the same label as $\mathbf{x}$. We do the same on queries and average the class score vectors over positions. The fourth is the inference mechanism of NBNN [12] (5). For each method, we experiment with and without our spatial attention mechanism (8). For Local Match and NBNN, we select a subset of local features per image. For GAP+Proto and Matching Net, we apply GAP to the selected subset only.

In the *transductive* setting, we compare with the inference mechanism of global label propagation [13], [16], with and without global embedding propagation [16]. These baselines are again evaluated with and without spatial attention. We always include spatial attention in our local propagation, but we experiment with and without feature pooling, with and without feature propagation.

B. Ablation studies

Overall, our method has five parameters. Two refer to optional components related to local information: the threshold τ for spatial attention and the number of clusters m for feature pooling. The other three refer to propagation, like all related methods dating back to [34]: the number of neighbors k in the graph, the exponent γ in the feature similarity function and α, controlling the amount of the smoothing. For all experiments, we perform a fairly exhaustive parameter search over a small set of possible values per parameter and we make choices according to validation accuracy. We present a summary of parameter search independently for τ and m, keeping other parameters fixed to the optimal.
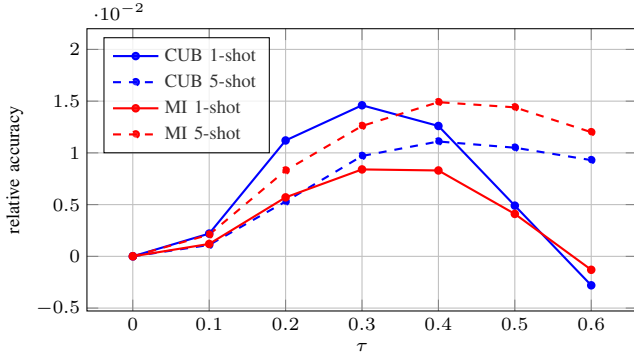
Fig. 5. *Spatial attention* on our local label propagation, including feature pooling and feature propagation: 5-way few-shot classification accuracy *vs.* threshold τ, relative to $\tau = 0$ (no attention). All other parameters fixed to optimal.

Fig. 6. *Feature pooling* on our local propagation: 5-way few-shot classification accuracy *vs.* number of clusters m, relative to $m = 10$ for better visualization. TR: transductive. We use spatial attention in all settings and feature propagation only in transductive. All other parameters fixed to optimal.

Spatial Attention. As shown in Figure 4, referring to GAP+Proto baseline, there is an optimal range of τ in $[0.3, 0.5]$, such that we filter out the uninformative local feature without removing too much information. The same behavior appears in Figure 5, referring to our best method for each setting. For the remaining of the experiments, we fix τ to 0.3.

Feature pooling. This is a compromise between global pooling and a full set of local features per image, which brings a consistent small improvement compared to both, while making local propagation more efficient by limiting the graph size. According to Figure 6, referring again to our best method for each setting, there is an optimal number m of clusters that depends on the dataset and setting (transductive or not, 1/5-shot). On CUB, we use $m = 40$ for 1-shot and $m = 60$ for 5-shot. On *mini*ImageNet, we use $m = 60$ in both cases.

Propagation parameters. Propagation has been extensively researched in the past, so we do not report the study of its parameters. It is known for instance that α should be close to 1 and there is a local maximum with respect to k, which depends on the quantity of the data [62]. After parameter search, for most experiments we set $\alpha = 0.9$, $\gamma = 4$, and $k = 5$, $k = 50$ respectively for global and local propagation.

C. Results

Table I presents a complete set of results our method and baselines in different settings, using different options. We discuss the effect of our contributions below.

Spatial attention. We use spatial attention with our method but we also combine it with baselines for fair comparison. It is an extremely simple mechanism that consistently improves few-shot classification accuracy in most cases, including global or local, with propagation or not, transductive or not. The only exception is Local Match on *mini*ImageNet. The gain is more pronounced on 1-shot tasks, which is expected as information selection is more important when information is scarce. It reaches 3% for the baselines and 2% for propagation on CUB, as well as 1% on *mini*ImageNet.
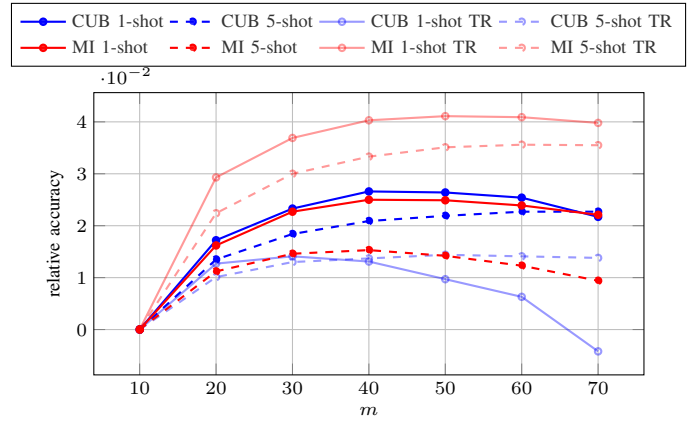
Feature pooling. Clustering the set of local features into a given number of clusters for each image is bringing small accuracy improvements when combined with propagation, local or global. In particular, spatial attention and feature pooling brings a 0.30% to 0.75% increase of accuracy compared to spatial attention alone on CUB (transductive and non-transductive). An exception is *mini*ImageNet non-transductive where feature pooling gives slightly worse accuracy by an insignificant margin.

Label propagation. In the *non-transductive* setting, global label propagation fails. Its performance is similar or inferior to GAP+Protonet. This is to be expected, as this is method a transductive method, so it is not a natural choice given only one query. By contrast, our local label propagation succeeds even in this setting, with up to 2.7% improvement on CUB 5-way 1-shot compared to GAP+Proto. One exception is *mini*ImageNet 5-way 1-shot, where GAP+Proto is better by a small margin; in this case however, the other local baselines (Local Match and NBNN) are worse than both GAP+Proto and our local label propagation, by a larger margin.

In the *transductive* setting, label propagation, global or local, always helps by using unlabeled data. Our local label propagation with spatial attention and feature pooling improves 5-way 1-shot accuracy over the non-transductive setting by 6% and 5.5%, on CUB and *mini*ImageNet respectively. This improvement is lower for 5-shot tasks as more labeled data are used. Compared with global label propagation, it improves by up to 1.5% on 5-shot, CUB and *mini*ImageNet.

Feature propagation. In the *non-transductive* setting, feature propagation is mostly harmful, especially when used with our local label propagation, which remains the best option, together with feature pooling. In the *transductive* setting however, it helps both global and local label propagation, the only exception being 5-shot, *mini*ImageNet. In the case of local label propagation with feature pooling, the gain is up to 2% and 1.5% on 1-shot, CUB and *mini*ImageNet respectively.

Therefore this combination is the most effective, improving over our best non transductive result by 8% and 6.5% on 1-shot, CUB and *mini*ImageNet respectively.

Universality. As shown in Figure 1, our local label propagation is a *universally safe choice* for few-shot inference under both transductive and non-transductive settings. This is in contrast to existing methods such as global label propagation, where the user needs to make decisions depending on the amount of unlabeled data that is available.

Comparison to existing methods. Table I also includes a number of recent few-shot learning methods. For fair comparison, all reported results are using the same ResNet12 as embedding network. We observe that our baseline GAP+Proto is better than these models on non-transductive 5-shot classification on *mini*ImageNet. Our method is then outperforming those models as well. In the transductive setting, global propagation is weaker than existing methods, but our best setting of local propagation (including spatial attention, feature pooling, feature propagation and label propagation) is stronger in general. The only exception is 1-shot classification on CUB, where LR+ICI [13] is stronger by a small margin.

In parallel with this work, two methods appeared very recently, which are stronger than our solution on *mini*ImageNet but weaker on CUB: (1) DGPN [40], which is yet another graph-based method and could be easily integrated with our local propagation. (2) DeepEMD [55], which is based on pairwise image alignment. This is more challenging to integrate, for instance one would need to use alignment in the definition of the graph itself. This can be interesting future work.

VII. Conclusion

Our *local propagation* framework takes the best of both worlds: more data from local representations and better use of this data from propagation. It provides a unified solution that works well given few labeled data and an arbitrary number of unlabeled data. As a result, it works better that solutions meant for the standard few-shot inference and at the same time better than solutions meant for transductive few-shot inference. Two secondary contributions are extremely simple and effective: (a) our *feature pooling* helps control the additional cost related to local features, while improving performance in most cases; (b) our *spatial attention* helps not only our method but all baselines too, by a significant margin on 1-shot classification. Our solution only affects inference, so it can easily be plugged into any alternative representation learning method. It is general enough to integrate other state-of-the-art solutions, like pairwise image alignment, other forms of propagation and propagation on several layers.

References

[1] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *NIPS*, 2016.

[2] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *NIPS*, 2017.

[3] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *CVPR*, 2018.

METHOD	A	P	F	CUB 1-SHOT	CUB 5-SHOT	*mini*IMAGENET 1-SHOT	*mini*IMAGENET 5-SHOT
GAP+Proto [2]				$74.85_{\pm0.48}$	$90.38_{\pm0.27}$	$63.39_{\pm0.46}$	$81.21_{\pm0.32}$
GAP+Proto [2]	✓			$77.10_{\pm0.47}$	$91.24_{\pm0.26}$	$64.22_{\pm0.45}$	$81.71_{\pm0.31}$
Matching Net [1]				$74.85_{\pm0.48}$	$89.23_{\pm0.29}$	$63.39_{\pm0.46}$	$78.14_{\pm0.33}$
Matching Net [1]	✓			$77.10_{\pm0.47}$	$89.95_{\pm0.28}$	$64.22_{\pm0.45}$	$78.70_{\pm0.33}$
Local Match [1]				$75.92_{\pm0.46}$	$89.16_{\pm0.28}$	$64.05_{\pm0.46}$	$78.45_{\pm0.34}$
Local Match [1]	✓			$78.29_{\pm0.45}$	$90.60_{\pm0.26}$	$63.58_{\pm0.46}$	$78.01_{\pm0.35}$
NBNN [12]				$76.21_{\pm0.45}$	$89.59_{\pm0.27}$	$64.90_{\pm0.45}$	$79.74_{\pm0.32}$
NBNN [12]	✓			$79.14_{\pm0.44}$	$91.40_{\pm0.25}$	$65.18_{\pm0.45}$	$80.00_{\pm0.31}$
GLOBAL LABEL PROPAGATION, NON-TRANSDUCTIVE							
Propagation				$74.69_{\pm0.48}$	$87.96_{\pm0.30}$	$63.39_{\pm0.46}$	$75.89_{\pm0.36}$
Propagation	✓			$76.94_{\pm0.47}$	$89.14_{\pm0.30}$	$64.22_{\pm0.45}$	$76.40_{\pm0.36}$
Propagation	✓		✓	$77.23_{\pm0.46}$	$88.78_{\pm0.31}$	$63.41_{\pm0.45}$	$77.04_{\pm0.37}$
LOCAL LABEL PROPAGATION (THIS WORK), NON-TRANSDUCTIVE							
Propagation				$78.24_{\pm0.44}$	$91.07_{\pm0.26}$	$65.52_{\pm0.45}$	$80.49_{\pm0.31}$
Propagation	✓			$79.02_{\pm0.44}$	$91.81_{\pm0.25}$	$\mathbf{65.74_{\pm0.45}}$	$\mathbf{81.13_{\pm0.31}}$
Propagation	✓	✓		$\mathbf{79.77_{\pm0.44}}$	$\mathbf{92.07_{\pm0.25}}$	$65.59_{\pm0.45}$	$80.73_{\pm0.31}$
Propagation	✓	✓	✓	$79.32_{\pm0.44}$	$91.52_{\pm0.25}$	$64.43_{\pm0.45}$	$80.26_{\pm0.32}$
GLOBAL LABEL PROPAGATION, TRANSDUCTIVE							
Propagation				$83.64_{\pm0.48}$	$90.63_{\pm0.27}$	$70.07_{\pm0.51}$	$80.96_{\pm0.34}$
Propagation	✓			$85.52_{\pm0.46}$	$91.67_{\pm0.27}$	$70.67_{\pm0.51}$	$81.44_{\pm0.33}$
Propagation	✓		✓	$87.18_{\pm0.46}$	$91.88_{\pm0.27}$	$72.54_{\pm0.54}$	$81.38_{\pm0.35}$
LOCAL LABEL PROPAGATION (THIS WORK), TRANSDUCTIVE							
Propagation				$83.04_{\pm0.43}$	$91.89_{\pm0.25}$	$69.95_{\pm0.48}$	$82.13_{\pm0.31}$
Propagation	✓			$85.33_{\pm0.42}$	$92.50_{\pm0.25}$	$71.00_{\pm0.48}$	$\mathbf{82.87_{\pm0.30}}$
Propagation	✓	✓		$85.80_{\pm0.41}$	$92.92_{\pm0.24}$	$71.12_{\pm0.48}$	$82.83_{\pm0.31}$
Propagation	✓	✓	✓	$\mathbf{87.77_{\pm0.41}}$	$\mathbf{93.35_{\pm0.23}}$	$\mathbf{72.57_{\pm0.51}}$	$82.76_{\pm0.33}$
OTHER MODELS, NON-TRANSDUCTIVE							
SNAIL [44]				-	-	$55.71_{\pm0.99}$	$68.88_{\pm0.92}$
TADAM [46]				-	-	$58.50_{\pm0.30}$	$76.70_{\pm0.30}$
DC+IMP [11]				-	-	$62.53_{\pm0.19}$	$79.77_{\pm0.19}$
Neg-Cosine [73]				-	-	$62.33_{\pm0.82}$	$80.94_{\pm0.59}$
OTHER MODELS, TRANSDUCTIVE							
TPN [13]				-	-	59.46	75.65
LR+ICI [39]				88.06	92.53	66.80	79.26
EPNet [16]				$82.85_{\pm0.81}$	$91.32_{\pm0.41}$	$66.50_{\pm0.89}$	$81.06_{\pm0.60}$

TABLE I

5-WAY FEW-SHOT CLASSIFICATION ACCURACY, COMPARING OUR LOCAL (FEATURE AND LABEL) PROPAGATION TO BASELINES AND EXISTING WORK. A: SPATIAL ATTENTION (OUR WORK, ALSO APPLIED TO BASELINES). P: FEATURE POOLING (CLUSTERING) (OUR WORK). F: FEATURE PROPAGATION [16].

[4] E. Bart and S. Ullman, "Cross-generalization: Learning novel classes from a single example by feature replacement," in *CVPR*, 2005.

[5] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial Intelligence Review*, vol. 18, no. 2, pp. 77–95, 2002.

[6] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017.

[7] X. Li, L. Yu, C.-W. Fu, M. Fang, and P.-A. Heng, "Revisiting metric learning for few-shot image classification," *arXiv preprint arXiv:1907.03123*, 2019.

[8] W. Chen, Y. Liu, Z. Kira, Y. F. Wang, and J. Huang, "A closer look at few-shot classification," *ICLR*, 2019.

[9] Y. Wang, W.-L. Chao, K. Q. Weinberger, and L. van der Maaten, "SimpleShot: Revisiting nearest-neighbor classification for few-shot learning," *arXiv preprint arXiv:1911.04623*, 2019.

[10] Y. Wang, S. Khan, A. Gonzalez-Garcia, J. v. d. Weijer, and F. S. Khan, "Semi-supervised learning for few-shot image-to-image translation," in *CVPR*, 2020.

[11] Y. Lifchitz, Y. Avrithis, S. Picard, and A. Bursuc, "Dense classification and implanting for few-shot learning," *CVPR*, 2019.

[12] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, and J. Luo, "Revisiting local descriptor based image-to-class measure for few-shot learning," in *CVPR*, 2019.

[13] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. Hwang, and Y. Yang,

"Learning to propagate labels: Transductive propagation network for few-shot learning," in *ICLR*, 2018.

[14] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," in *ICLR*, 2018.

[15] S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez, and M. Cord, "Boosting few-shot visual learning with self-supervision," in *ICCV*, 2019.

[16] P. Rodríguez, I. Laradji, A. Drouin, and A. Lacoste, "Embedding propagation: Smoother manifold for few-shot classification," *arXiv preprint arXiv:2003.04151*, 2020.

[17] E. G. Miller, N. E. Matsakis, and P. A. Viola, "Learning from one example through shared densities on transforms," in *CVPR*, 2000.

[18] L. Fei-Fei, R. Fergus, and P. Perona, "A Bayesian approach to unsupervised one-shot learning of object categories," in *ICCV*, 2003.

[19] Z. Wu, A. A. Efros, and S. X. Yu, "Improving generalization via scalable neighborhood component analysis," in *ECCV*, 2018.

[20] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *CVPR*, 2018.

[21] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *CVPR*, 2018.

[22] P. Mangla, N. Kumari, A. Sinha, M. Singh, B. Krishnamurthy, and V. N. Balasubramanian, "Charting the right manifold: Manifold mixup for few-shot learning," in *WACV*, 2020.

[23] J.-W. Seo, H.-G. Jung, and S.-W. Lee, "Self-augmentation: Generalizing deep networks to unseen classes for few-shot learning," *arXiv preprint arXiv:2004.00251*, 2020.

[24] C. Le, Z. Chen, X. Wei, B. Wang, and L. Zhang, "Continual local replacement for few-shot learning," *arXiv preprint arXiv:2001.08366*, 2020.

[25] Z. Chen, Y. Fu, Y. Zhang, Y.-G. Jiang, X. Xue, and L. Sigal, "Semantic feature augmentation in few-shot learning," *arXiv preprint arXiv:1804.05298*, 2018.

[26] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song, "(metagan): an adversarial approach to few-shot learning," in *NeurIPS*, 2018.

[27] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, "Few-shot unsupervised image-to-image translation," in *ICCV*, 2019.

[28] A. Li, T. Luo, T. Xiang, W. Huang, and L. Wang, "Few-shot learning with global class representations," in *ICCV*, 2019.

[29] A. Afrasiyabi, J.-F. Lalonde, and C. Gagné, "Associative alignment for few-shot image classification," *arXiv preprint arXiv:1912.05094*, 2019.

[30] M. Douze, A. Szlam, B. Hariharan, and H. Jégou, "Low-shot learning with large-scale diffusion," in *CVPR*, 2018.

[31] A. Iscen, G. Tolias, Y. Avrithis, O. Chum, and C. Schmid, "Graph convolutional networks for learning with few clean and many noisy labels," *arXiv preprint arXiv:1910.00324*, 2019.

[32] M. Rohrbach, S. Ebert, and B. Schiele, "Transfer learning in a transductive setting," in *NIPS*, 2013.

[33] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *CoRR, abs/1803.02999*, 2018.

[34] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *NIPS*, 2003.

[35] J. Kim, T. Kim, S. Kim, and C. D. Yoo, "Edge-labeling graph neural network for few-shot learning," in *CVPR*, 2019.

[36] L. Qiao, Y. Shi, J. Li, Y. Wang, T. Huang, and Y. Tian, "Transductive episodic-wise adaptive metric for few-shot learning," in *ICCV*, 2019.

[37] Y. Hu, V. Gripon, and S. Pateux, "Exploiting unsupervised inputs for accurate few-shot classification," *arXiv preprint arXiv:2001.09849*, 2020.

[38] S. M. Kye, H. B. Lee, H. Kim, and S. J. Hwang, "Transductive few-shot learning with meta-learned confidence," *arXiv preprint arXiv:2002.12017*, 2020.

[39] Y. Wang, C. Xu, C. Liu, L. Zhang, and Y. Fu, "Instance credibility inference for few-shot learning," *arXiv preprint arXiv:2003.11853*, 2020.

[40] L. Yang, L. Li, Z. Zhang, E. Zhou, Y. Liu *et al.*, "DPGN: Distribution propagation graph network for few-shot learning," *arXiv preprint arXiv:2003.14247*, 2020.

[41] M. Ren, S. Ravi, E. Triantafillou, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," in *ICLR*, 2018.

[42] Z. Yu, L. Chen, Z. Cheng, and J. Luo, "TransMatch: A transfer-learning scheme for semi-supervised few-shot learning," *arXiv preprint arXiv:1912.09033*, 2019.

[43] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning," in *CVPR*, 2019.

[44] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," *ICLR*, 2018.

[45] M. Ren, R. Liao, E. Fetaya, and R. S. Zemel, "Incremental few-shot learning with attention attractor networks," *arXiv preprint arXiv:1810.07218*, 2018.

[46] B. N. Oreshkin, A. Lacoste, and P. Rodriguez, "TADAM: Task dependent adaptive metric for improved few-shot learning," *arXiv preprint arXiv:1805.10123*, 2018.

[47] H. Li, D. Eigen, S. Dodge, M. Zeiler, and X. Wang, "Finding task-relevant features for few-shot learning by category traversal," in *CVPR*, 2019.

[48] D. Wertheimer and B. Hariharan, "Few-shot learning with localization in realistic settings," in *CVPR*, 2019.

[49] H. Zhang, J. Zhang, and P. Koniusz, "Few-shot learning via saliency-guided hallucination of samples," in *CVPR*, 2019.

[50] H. Xv, X. Sun, J. Dong, S. Zhang, and Q. Li, "Multi-level similarity learning for low-shot recognition," *arXiv preprint arXiv:1912.06418*, 2019.

[51] Y. Lifchitz, Y. Avrithis, and S. Picard, "Few-shot few-shot learning and the role of spatial attention," *arXiv preprint arXiv:2002.07522*, 2020.

[52] R. Hou, H. Chang, M. Bingpeng, S. Shan, and X. Chen, "Cross attention network for few-shot classification," in *NeurIPS*, 2019.

[53] F. Hao, F. He, J. Cheng, L. Wang, J. Cao, and D. Tao, "Collect and select: Semantic alignment metric learning for few-shot learning," in *ICCV*, 2019.

[54] Z. Wu, Y. Li, L. Guo, and K. Jia, "PARN: Position-aware relation networks for few-shot learning," in *ICCV*, 2019.

[55] C. Zhang, Y. Cai, G. Lin, and C. Shen, "DeepEMD: Few-shot image classification with differentiable earth mover's distance and structured classifiers," *arXiv preprint arXiv:2003.06777*, 2020.

[56] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on data manifolds," in *NIPS*, 2003.

[57] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Carnegie Mellon University, Tech. Rep., 2002.

[58] L. Grady, "Random walks for image segmentation," *IEEE Trans. PAMI*, vol. 28, no. 11, pp. 1768–1783, 2006.

[59] T. H. Kim, K. M. Lee, and S. U. Lee, "Generative image segmentation using random walks with restart," in *ECCV*. Springer, 2008, pp. 264–275.

[60] G. Bertasius, L. Torresani, S. X. Yu, and J. Shi, "Convolutional random walk networks for semantic image segmentation," in *CVPR*, 2017.

[61] P. Vernaza and M. Chandraker, "Learning random-walk label propagation for weakly-supervised semantic segmentation," in *CVPR*, 2017.

[62] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum, "Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations," in *CVPR*, 2017.

[63] R. Ranjan, C. D. Castillo, and R. Chellappa, "L2-constrained softmax loss for discriminative face verification," *arXiv preprint arXiv:1703.09507*, 2017.

[64] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L2 hypersphere embedding for face verification," in *ACM Multimedia*, 2017.

[65] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *CVPR*, 2008.

[66] N. Dvornik, C. Schmid, and J. Mairal, "Diversity with cooperation: Ensemble methods for few-shot classification," *ICCV*, 2019.

[67] Y. Kalantidis, C. Mellina, and S. Osindero, "Cross-dimensional weighting for aggregated deep convolutional features," in *ECCVW*, 2016.

[68] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *arXiv*, 2014.

[69] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," *ICLR*, 2017.

[70] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[71] N. Hilliard, L. Phillips, S. Howland, A. Yankov, C. D. Corley, and N. O. Hodas, "Few-shot learning with metric-agnostic conditional embeddings," *CoRR*, vol. abs/1802.04376, 2018.

[72] H. Ye, H. Hu, D. Zhan, and F. Sha, "Learning embedding adaptation for few-shot learning," *CoRR*, vol. abs/1812.03664, 2018.

[73] B. Liu, Y. Cao, Y. Lin, Q. Li, Z. Zhang, M. Long, and H. Hu, "Negative margin matters: Understanding margin in few-shot classification," *arXiv preprint arXiv:2003.12060*, 2020.