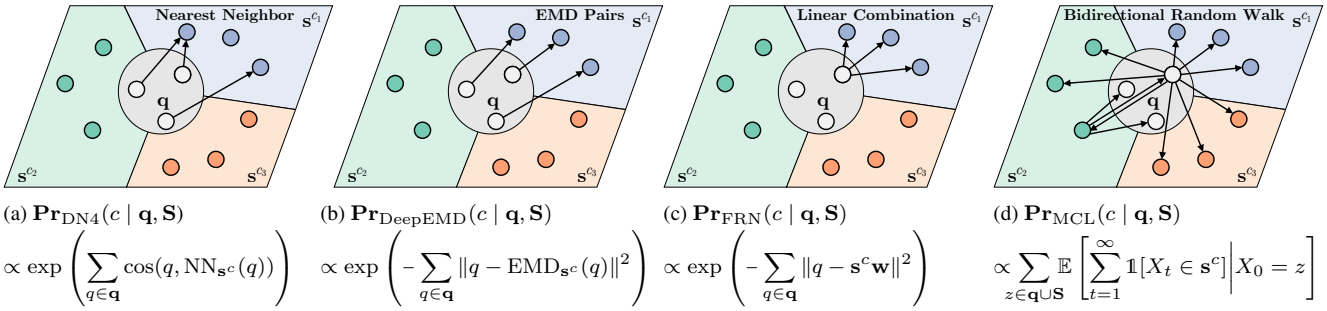
Learning to Affiliate: Mutual Centralized Learning for Few-shot Classification

Yang Liu[1], Weifeng Zhang[1], Chao Xiang[1], Tu Zheng[1,2], Deng Cai[1,2], Xiaofei He[1,2]
[1]State Key Lab of CAD&CG, Zhejiang University [2]Fabu Inc., Hangzhou, China
{lyng_95, zhangwf, chaoxiang, zhengtu}@zju.edu.cn dengcai@cad.zju.edu.cn hexiaofei@fabu.ai



(a) $\mathbf{Pr}_{\mathrm{DN4}}(c \mid \mathbf{q}, \mathbf{S}) \propto \exp\left(\sum_{q \in \mathbf{q}} \cos(q, \mathrm{NN}_{\mathbf{s}^c}(q))\right)$

(b) $\mathbf{Pr}_{\mathrm{DeepEMD}}(c \mid \mathbf{q}, \mathbf{S}) \propto \exp\left(-\sum_{q \in \mathbf{q}} \|q - \mathrm{EMD}_{\mathbf{s}^c}(q)\|^2\right)$

(c) $\mathbf{Pr}_{\mathrm{FRN}}(c \mid \mathbf{q}, \mathbf{S}) \propto \exp\left(-\sum_{q \in \mathbf{q}} \|q - \mathbf{s}^c \mathbf{w}\|^2\right)$

(d) $\mathbf{Pr}_{\mathrm{MCL}}(c \mid \mathbf{q}, \mathbf{S}) \propto \sum_{z \in \mathbf{q} \cup \mathbf{S}} \mathbb{E}\left[\sum_{t=1}^{\infty} \mathbb{1}[X_t \in \mathbf{s}^c] \,\middle|\, X_0 = z\right]$

Figure 1. Illustrations of four dense features based methods in a 3-way 1-shot scenario, where each colored pane indicates an image and particles within each pane indicate three feature vectors of that image. The arrow line indicates a direct connection between two feature vectors in inference. For example, (a) DN4 [12] accumulates cosine similarities between the nearest features; (b) DeepEMD [31] finds the optimal matching flow that has the minimum cost; (c) FRN [29] reconstructs query features with the linear combination of support features. (d) Our MCL lets query and support features bidirectionally random walk to the opposite dense features set and measures the expected number of visits in a Markov process $\{X_t\}$. Difference from them, MCL considers mutual affiliations between the dense features instead of following the unidirectional paradigm whose evident character is an accumulation over all query features followed by a softmax probability.

Abstract

Few-shot learning (FSL) aims to learn a classifier that can be easily adapted to accommodate new tasks, given only a few examples. To handle the limited-data in few-shot regimes, recent methods tend to collectively use a set of local features to densely represent an image instead of using a mixed global feature. They generally explore a unidirectional paradigm, e.g.*, finding the nearest support feature for every query feature and aggregating local matches for a joint classification. In this paper, we propose a novel Mutual Centralized Learning (MCL) to fully affiliate these two disjoint dense features sets in a bidirectional paradigm. We first associate each local feature with a particle that can bidirectionally random walk in discrete feature space. To estimate the class probability, we propose the dense features' accessibility that measures the expected number of visits to the dense features of that class in a Markov process. We relate our method to learning a centrality on an affiliation network and demonstrate its capability to be plugged in existing methods by highlighting centralized local features. Experiments show that our method achieves the new state-of-the-art.*

1. Introduction

Few-shot classification aims to learn a classifier that can be readily adapted to novel classes given just a small number of labeled instances. To address this problem, a line of previous literature adopts metric-based methods [23, 25, 26] that learn a global image representation in an appropriate feature space and use a distance metric to predict their labels.

Recent approaches [12, 14, 17, 31] have demonstrated that the significant intra-class variations would inevitably drive the image-level embedding from the same category far apart in a given metric space under low-data regimes. In contrast, densely representative local features can provide transferrable information across categories that have shown promising performances in the few-shot scenario. Among those methods illustrated in Figure 1, DN4 [12] finds the nearest neighbor support feature for each query feature and accumulates all the local matches in a Naive-Bayes way to represent an image-to-class similarity; DeepEMD [31] uses the earth mover distance to compare the complex structured representations composed of local features. FRN [29] reconstructs each of query dense features with a linear combination of support dense features in a latent space and use the reconstruction distance to measure the image-to-class

relevance. They all follow a unidirectional query-to-support paradigm, whose evident character is an accumulation over all query features followed by a softmax probability.

In this paper, we incorporate an extra support-to-query connection as a complement to thoroughly affiliate two disjoint sets of dense features. The potential offered by this bidirectional paradigm stems from the intuition that, except for using query features to find related support features, it is also plausible to estimate the task-relevance of query features according to the support features. Specifically, we associate dense features with particles that could bidirectionally random walk to the opposite dense features set in the discrete feature space. The prediction probability for each class is then estimated by the dense features' accessibility, *i.e*., the expected number of visits to the dense features of that class in a time-homogeneous Markov process.

The contributions are as follows: (1) We propose to learn mutual affiliations between the query and support dense features instead of following the unidirectional query-to-support paradigm in FSL. (2) We introduce the dense features' accessibility to FSL and demonstrate that traditional transductive methods could be easily adapted to the inductive setting if we treat dense features from a single query image as a set of unlabeled data. (3) We propose a novel bidirectional random walk based method in FSL and draw its connection to the single-mode eigenvector centrality of an affiliation network. (4) The underlying centrality investigated in this work can be plugged in existing global feature based methods like ProtoNet and RelationNet by highlighting task-centralized local features instead of global average pooling.

2. Related Work

Dense Feature based FSL. Those methods focus on learning image-to-image similarities by encoding each input into a set of dense feature vectors (or a feature map). Among them, DC [14] proposes to predict for each local features and average their probabilities. DeepEMD [31] adopts the earth mover's distance to compute an image-to-class distance. DN4 [12] uses the top-*k* nearest neighbors in a Naive-Bayes way to represent image-level similarities. ADM [11] and SAML [8] use extra network layers for dense classifications. FRN [29] reconstructs the dense query features from support features of a given class to predict their relevance. Lifchitz *et al*. [13] propagates labels from support sets to query features. MCL is also among the family of dense feature based frameworks. A major difference is that we consider the mutual affiliations between the query and support features instead of the unidirectional one in previous work.

Graphical models in FSL. Another branch of methods learns graphs on FSL. Among them, [7] build graph neural network (GNN) from a collection of images. [5] use the degree-centrality as part of graph features to adjust the weight of graph vertex in transductive FSL. Our proposed feature accessibility of bidirectional random walks is well-related to the eigenvector centrality. Differently, we use the single-mode centrality of bipartite data as a novel criterion for the inductive classification.

From Transductive to Inductive. TPN [16] first introduces the well-known label propagation [32] to the transductive FSL that transfers information from labeled data to unlabeled one and shows substantial improvements over inductive methods. Lifchitz *et al*. [13] first demonstrates it feasible to adopt label for inductive FSL by treating the query dense features as numerous unlabeled data. Different from unidirectional label propagation, our MCL further considers the set of query and support features as bipartite data where the self-reinforcements (*i.e*., query-to-query and support-to-support random walks) are avoided.

3. Method

3.1. Formulation of Dense Features based FSL

In a N-way K-shot episode, the goal is to predict a class label for a single query image given N support classes, each of which contains K support images for reference.

We first encode each input image into r number of d-dimensional dense features $f_\theta(\cdot) \in \mathbb{R}^{d\times r}$. We use $\mathbf{q} = \{q_1, ..., q_r\}$ to denote dense features from a query image. Inspired by prototype learning [23], we average feature vectors from the same spatial locations of K different images into support dense features for each $\mathbf{s}^c = \{s_1^c, ..., s_r^c\}$ category.

We use the bold font (*e.g*., $\mathbf{q}$ and $\mathbf{s}$) to denote a set of feature vectors and use the normal font (*e.g*., q and s) to denote a single feature vector. We use $\mathbf{S} = \bigcup_{c\in C} \mathbf{s}^c$ to represent the union set of dense features from all supporting classes. The cardinalities for the sets $\mathbf{q}$, $\mathbf{s}^c$ and $\mathbf{S}$ are r, r and Nr respectively.

The target of dense features based FSL is to find the label of a query image, given its dense query features $\mathbf{q}$ and the union set of dense support features $\mathbf{S}$ from all categories.

3.2. Bidirectional Random Walks on Dense Features

We start by using the *cosine similarity* $\phi(v_1, v_2) = \langle \frac{v_1}{\|v_1\|}, \frac{v_2}{\|v_2\|} \rangle$ to measure the closeness between the two feature vectors where $\langle \cdot, \cdot \rangle$ is the Frobenius inner product.

Given two dense features sets $\mathbf{q}$ and $\mathbf{S}$, we learn inter similarity matrix $\mathbf{\Phi} \in \mathbb{R}^{r\times Nr}$ between those two disjoint sets where the entry of $[\mathbf{\Phi}]_{qs} = \phi(q, s)$ indicates the similarity between vectors $q \in \mathbf{q}$ and $s \in \mathbf{S}$.

We build query-to-support affiliations from every query feature q to all support features $s \in \mathbf{S}$ by a random walk probability $p_{sq} = \frac{e^{\gamma\phi(s,q)}}{\sum_{s'\in\mathbf{S}} e^{\gamma\phi(s',q)}}$. The probability matrix from all query to support features can be written by

$$\mathbf{P}_{\mathbf{Sq}} = \exp(\gamma\mathbf{\Phi}^{\mathrm{T}})\mathbf{D}^{-1} \tag{1}$$

where $\exp(\cdot)$ is the element-wise exponential function, γ is a scaling temperature for a softmax like probability and $\mathbf{D}$ is a diagonal normalization matrix with its (j,j)-value to be the sum of the j-th column of $\exp(\gamma\mathbf{\Phi}^{\mathrm{T}})$. The subscript serves both as the matrix size and as an indication for the feature vector with which it is associated. For example, $\mathbf{P_{Sq}}$ is of size $|\mathbf{S}| \times |\mathbf{q}|$ and it includes random walk probabilities from query features $\mathbf{q}$ to support features $\mathbf{S}$.

Likewise, we connect every support feature s to all $q \in \mathbf{q}$ by a similar probability matrix:

$$\mathbf{P_{qS}} = \exp(\beta\mathbf{\Phi})\mathbf{W}^{-1} \tag{2}$$

where $\mathbf{W}$ is the analogous diagonal matrix with its (j,j)-value to be the sum of the j-th column of $\exp(\beta\mathbf{\Phi})$.

It should be noted that we use two different scaling parameters since $\mathbf{\Phi}$ is a non-square matrix due to the different cardinalities of dense feature sets $\mathbf{q}$ and $\mathbf{S}$. Although the matrices $\exp(\gamma\mathbf{\Phi}^{\mathrm{T}})$ and $\exp(\beta\mathbf{\Phi})$ would be symmetric if we ignore their different scaling parameters, the probability matrices $\mathbf{P_{qS}}, \mathbf{P_{Sq}}$ are both column-normalized by different $\mathbf{D}, \mathbf{W}$ respectively and are thus directional.

To learn the mutual affiliations between the two dense feature sets, we first associate each feature vector with a particle z that forms a discrete feature space $\mathbf{z} = \mathbf{q} \cup \mathbf{S}$. Next, we let particles bidirectionally random walk between particles in that space according to $\mathbf{P_{Sq}}$ and $\mathbf{P_{qS}}$ (*i.e.*, particles at query features set are only allowed to move to the support features set, and vice versa). We assume it time-homogeneous in a Markov process $\{X_t\}$ and the random walk probability from z_j to z_i for all z_i, z_j in $\mathbf{z}$ can be formulated by:

$$\mathbf{Pr}(X_{t+1} = z_i | X_t = z_j) = \begin{cases} \frac{e^{\gamma\phi(z_i,z_j)}}{\sum_{s\in\mathbf{S}} e^{\gamma\phi(s,z_j)}} & z_i \in \mathbf{S}, z_j \in \mathbf{q} \\ \frac{e^{\beta\phi(z_i,z_j)}}{\sum_{q\in\mathbf{q}} e^{\beta\phi(q,z_j)}} & z_i \in \mathbf{q}, z_j \in \mathbf{S} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

To write it in matrix, we can obtain the Markov transition matrix $\mathbf{P}$:

$$\mathbf{P} = \begin{pmatrix} \mathbf{0} & \mathbf{P_{Sq}} \\ \mathbf{P_{qS}} & \mathbf{0} \end{pmatrix} \tag{4}$$

that consists of the column-normalized $\mathbf{P_{Sq}}$ and $\mathbf{P_{qS}}$.

It can be proved (in supplementary) that, after infinity times of bidirectional random walks, the Markov chain with anti-diagonal $\mathbf{P}$ reaches the periodic stationary distribution:

$$\begin{aligned} \lim_{t\to\infty} \mathbf{P}^{2t} &= \begin{pmatrix} \boldsymbol{\pi}(\mathbf{S})\boldsymbol{e}_{Nr}^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\pi}(\mathbf{q})\boldsymbol{e}_{r}^{\mathrm{T}} \end{pmatrix} \\ \lim_{t\to\infty} \mathbf{P}^{2t-1} &= \begin{pmatrix} \mathbf{0} & \boldsymbol{\pi}(\mathbf{S})\boldsymbol{e}_{r}^{\mathrm{T}} \\ \boldsymbol{\pi}(\mathbf{q})\boldsymbol{e}_{Nr}^{\mathrm{T}} & \mathbf{0} \end{pmatrix} \end{aligned} \tag{5}$$

where $\boldsymbol{\pi}(\mathbf{S}) \in \mathbb{R}^{Nr}$, $\boldsymbol{\pi}(\mathbf{q}) \in \mathbb{R}^{r}$ are the stationary distributions with equations $\boldsymbol{\pi}(\mathbf{S}) = \mathbf{P_{Sq}}\mathbf{P_{qS}}\boldsymbol{\pi}(\mathbf{S})$ and $\boldsymbol{\pi}(\mathbf{q}) = \mathbf{P_{qS}}\mathbf{P_{Sq}}\boldsymbol{\pi}(\mathbf{q})$ respectively. $\boldsymbol{e}_{|\cdot|}$ is a vector of ones with different length indicated by its subscript.

The motivation of using these stationary distributions to encode mutual affiliations is straightforward: support features would be frequently visited in the long times of bidirectional random walk if they shared the same local characteristics with the query ones. In other words, the support class that owns the most mutual affiliations with querying image would be predicted due to their mutual closeness.

3.3. Classification by Dense Features' Accessibility

To formulate it in FSL, we first assume particles are uniformly distributed in the discrete finite space $\mathbf{z}$. Then, we use the expected number of visits from all particles $z \in \mathbf{z}$ to support features $\mathbf{s}^c \subset \mathbf{S}$ of class c in the long times of bidirectional random walks $\{X_t\}$ to measure the amount of local characteristics that class c owns for the query image:

$$\begin{aligned} &\mathbf{Pr}(\tilde{y} = c) \propto \lim_{t\to\infty} \sum_{z\in\mathbf{z}} \mathbb{E}\left[\sum_{k=1}^{t} \mathbb{1}[X_k \in \mathbf{s}^c] \middle| X_0 = z\right] \\ &= \lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} \left[\frac{1}{Nr+r} \sum_{s\in\mathbf{s}^c} \left(\sum_{s'\in\mathbf{S}} \left[\mathbf{P}^{2k}\right]_{ss'} + \sum_{q\in\mathbf{q}} \left[\mathbf{P}^{2k-1}\right]_{sq}\right)\right] \\ &= \frac{1}{Nr+r} \sum_{s\in\mathbf{s}^c} \left(\sum_{s'\in\mathbf{S}} \left[\lim_{t\to\infty} \mathbf{P}^{2t}\right]_{ss'} + \sum_{q\in\mathbf{q}} \left[\lim_{t\to\infty} \mathbf{P}^{2t-1}\right]_{sq}\right) \\ &= \sum_{s\in\mathbf{s}^c} [\boldsymbol{\pi}(\mathbf{S})]_s \end{aligned} \tag{6}$$

where $\mathbb{1}[\cdot]$ is an indicator function that equals 1 if its argument is true and zero otherwise. $[\cdot]_{ij}$ indicates the entry of the matrix from particle j to particle i and $[\cdot]_i$ indicates the entry of the vector that associated to feature i.

We give the derivation when the Markov chain length t is even in the first equality of Eqn.(6) and prove that the odd t reaches the same result in the supplementary material. The second equality is from the *absorbing* property of the periodic Markov chain where the power of matrix gets saturated to Eqn.(5) for the increasing t. Since $\boldsymbol{\pi}(\mathbf{S})$ is the stationary distribution of column-stochastic $\mathbf{P_{Sq}}\mathbf{P_{qS}}$ under the probability constraint $\boldsymbol{e}_{Nr}^{\mathrm{T}}\boldsymbol{\pi}(\mathbf{S}) = 1$, $\mathbf{Pr}(\tilde{y})$ is a valid distribution for classifications.

3.4. Reinterpretation as a Graph Centrality

If we interpret the random walk probability matrix $\mathbf{P}$ as an adjacency matrix $\{a_{v_1,v_2}\}$ of a directed bipartite graph $G := (V = \{\mathbf{q}, \mathbf{S}\}, E)$, we find that the dense features' accessibility in the time-homogeneous bidirectional random walk would be equivalent to learning a *single-mode eigenvector centrality* on the graph G. The bipartite graph is also called the *affiliation network* in social network analysis [1–3] that models two types of entities "*actors*" and "*society*" related by affiliation of the former in the latter. The concept of centrality in social network analysis is generally used to

investigate the acquaintanceships among people that often stem from one or more shared affiliations.

To see this in graph theory, we start with a brief overview of the eigenvector centrality that reflects score x of vertex v for both $q \in \mathbf{q}$ and $s \in \mathbf{S}$ in the affiliation network with adjacency $\{a_{v_1,v_2}\}$:

$$\begin{aligned} x_q &= \frac{1}{\lambda} \sum_{v \in \mathbb{V}(q)} a_{q,v} x_v = \frac{1}{\lambda} \sum_{v \in \mathbf{S}} a_{q,v} x_v \\ x_s &= \frac{1}{\lambda} \sum_{v \in \mathbb{V}(s)} a_{s,v} x_v = \frac{1}{\lambda} \sum_{v \in \mathbf{q}} a_{s,v} x_v \end{aligned} \tag{7}$$

where $\mathbb{V}(\cdot)$ is a set of neighbors for the given vertex and λ is a constant.

With a small rearrangement, Eqn.(7) can be rewritten in vector notation with an eigenvector equation $\mathbf{P}\mathbf{x} = \lambda \mathbf{x}$. The additional requirement that all the entries of the eigenvector be non-negative implies (by the Perron–Frobenius theorem [18]) that only the greatest eigenvalue results in the desired centrality measure. For the column-stochastic adjacency matrix $\mathbf{P}$ in our method, the largest eigenvalue λ is 1.

Single-mode centrality [3] is a special form of graph centrality that measures the extent to which nodes in one vertex set are relatively central only to other nodes in the same vertex set on bipartite graph. For example, the single-mode centrality for different s in $\mathbf{S}$ is defined by $\hat{\mathbf{x}}_{\mathbf{S}} = \mathbf{x}_{\mathbf{S}} / \sum_{s \in \mathbf{S}} x_s$. Lemma 1 (proved in supplementary material) shows that dense features' accessibility $\boldsymbol{\pi}(\mathbf{S})$ of bidirectional random walks in Eqn.(6) is equivalent to the single-mode eigenvector centrality of support set $\mathbf{S}$ on bipartite data.

Lemma 1. *Assume G is the affiliation network of bipartite data $\mathbf{q}$, $\mathbf{S}$ with the adjacency matrix defined by the anti-diagonal Markov transition matrix $\mathbf{P}$ in Eqn.(4). The single-mode eigenvector centrality $\hat{\mathbf{x}}_{\mathbf{S}} = \mathbf{x}_{\mathbf{S}} / \sum_{s \in \mathbf{S}} x_s$ of vertex set $\mathbf{S}$ is equivalent to the dense features' accessibility $\boldsymbol{\pi}(\mathbf{S})$ on $\mathbf{S}$ in a time-homogeneous Markov process.*

Based on this reinterpretation, it is straightforward to introduce the attenuation (damping) factor α in Markov bidirectional random walks motivated by the Katz centrality [9], where connections made with distant neighbors are penalized by α. The dense features' accessibility with attenuation factor α for few-shot classifications is defined by

$$\begin{aligned} &\mathbf{Pr}_{\mathrm{Katz}}(\tilde{y} = c) \propto \sum_{z \in \mathbf{z}} \mathbb{E}\left[\sum_{t=1}^{\infty} \alpha^t \mathbb{1}[X_t \in \mathbf{s}^c] \middle| X_0 = z \right] \\ &= \frac{1}{\epsilon} \sum_{t=1}^{\infty} \sum_{s \in \mathbf{s}^c} \left(\sum_{s' \in \mathbf{S}} \left[\alpha^{2t} \mathbf{P}^{2t}\right]_{ss'} + \sum_{q \in \mathbf{q}} \left[\alpha^{2t-1} \mathbf{P}^{2t-1}\right]_{sq} \right) \end{aligned} \tag{8}$$

where $\epsilon = (Nr + r) \sum_{t=1}^{\infty} \alpha^t = (Nr + r)\alpha/(1 - \alpha)$ is a constant for a valid distribution.

Although we simply consider the single-mode centrality on $\mathbf{S}$ for an end-to-end classification purpose, it is also beneficial to learn its conjugate centrality $\boldsymbol{\pi}(\mathbf{q})$ on $\mathbf{q}$ in the affiliation network (by the analogous $\boldsymbol{\pi}(\mathbf{q}) = \mathbf{x}_{\mathbf{q}} / \sum_{q \in \mathbf{q}} x_q$ in Lemma 1). We will show that both single-mode centralities on two dense features sets could serve as plug-and-play for finding centralized local characteristics in existing methods (hence the term Mutual Centralized Learning, MCL).

3.5. End-to-End Training by Katz Approximation

The algorithm of Mutual Centralized Learning (MCL) in Eqn.(6) involves the computation of Markov stationary distribution $\boldsymbol{\pi}(\mathbf{S})$ with equation $\boldsymbol{\pi}(\mathbf{S}) = \mathbf{P}_{\mathbf{Sq}} \mathbf{P}_{\mathbf{qS}} \boldsymbol{\pi}(\mathbf{S})$. Theoretically, the $\boldsymbol{\pi}(\mathbf{S})$ is the eigenvector of $\mathbf{P}_{\mathbf{Sq}} \mathbf{P}_{\mathbf{qS}}$ with the eigenvalue 1 under a probability constraint $\boldsymbol{e}_{Nr}^{\mathrm{T}} \boldsymbol{\pi}(\mathbf{S}) = 1$.

The above constraints could lead to a solution of $\boldsymbol{\pi}(\mathbf{S})$ by solving the overdetermined linear system

$$\begin{pmatrix} \mathbf{P}_{\mathbf{Sq}} \mathbf{P}_{\mathbf{qS}} - \boldsymbol{I} \\ \boldsymbol{e}_{Nr}^{\mathrm{T}} \end{pmatrix} \boldsymbol{\pi}(\mathbf{S}) = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} \tag{9}$$

where $\boldsymbol{I}$ is an identity matrix and $\mathbf{0}$ is a vector of zeros. Although we can solve it by various methods like pseudo-inverse or QR decomposition and back substitution, it is empirically found time-consuming as these operators are either numerically unstable or not paralleled well in modern deep-learning packages.

To handle it, we present an alternative solution based on Lemma 1 as follows: We first calculate the Katz centrality by its closed-form solution [9]:

$$\mathbf{x}^{\mathrm{Katz}} = ((\boldsymbol{I} - \alpha \mathbf{P})^{-1} - \boldsymbol{I}) \boldsymbol{e}_{Nr+r} \tag{10}$$

and then solve single-mode Katz centrality in Eqn.(8) by the definition of the single-mode centrality:

$$\mathbf{Pr}_{\mathrm{Katz}}(\tilde{y} = c) = \frac{\sum_{s \in \mathbf{s}^c} \mathbf{x}_s^{\mathrm{Katz}}}{\sum_{s' \in \mathbf{S}} \mathbf{x}_{s'}^{\mathrm{Katz}}} \tag{11}$$

Since Katz centrality degrades to the eigenvector centrality when α approaches 1 [9] (indicates no attenuation), we can obtain the approximation of eigenvector centrality by a large $\alpha = 0.999$:

$$\begin{aligned} \mathbf{x}^{\mathrm{Eigen}} &= \lim_{\alpha \to 1} ((\boldsymbol{I} - \alpha \mathbf{P})^{-1} - \boldsymbol{I}) \boldsymbol{e}_{Nr+r} \\ &\approx ((\boldsymbol{I} - 0.999 \mathbf{P})^{-1} - \boldsymbol{I}) \boldsymbol{e}_{Nr+r} \end{aligned} \tag{12}$$

and the analogous single-mode eigenvector centrality in Eqn.(6) can then be approximated by:

$$\mathbf{Pr}_{\mathrm{MCL}}(\tilde{y} = c) = \frac{\sum_{s \in \mathbf{s}^c} \mathbf{x}_s^{\mathrm{Eigen}}}{\sum_{s' \in \mathbf{S}} \mathbf{x}_{s'}^{\mathrm{Eigen}}} \tag{13}$$

We use the negative log-likelihood loss to update parameters in different feature extractors and the whole picture of MCL for few-shot classifications is shown in Figure 2. We provide the detailed algorithm/pseudo-codes in the supplementary material for reference.

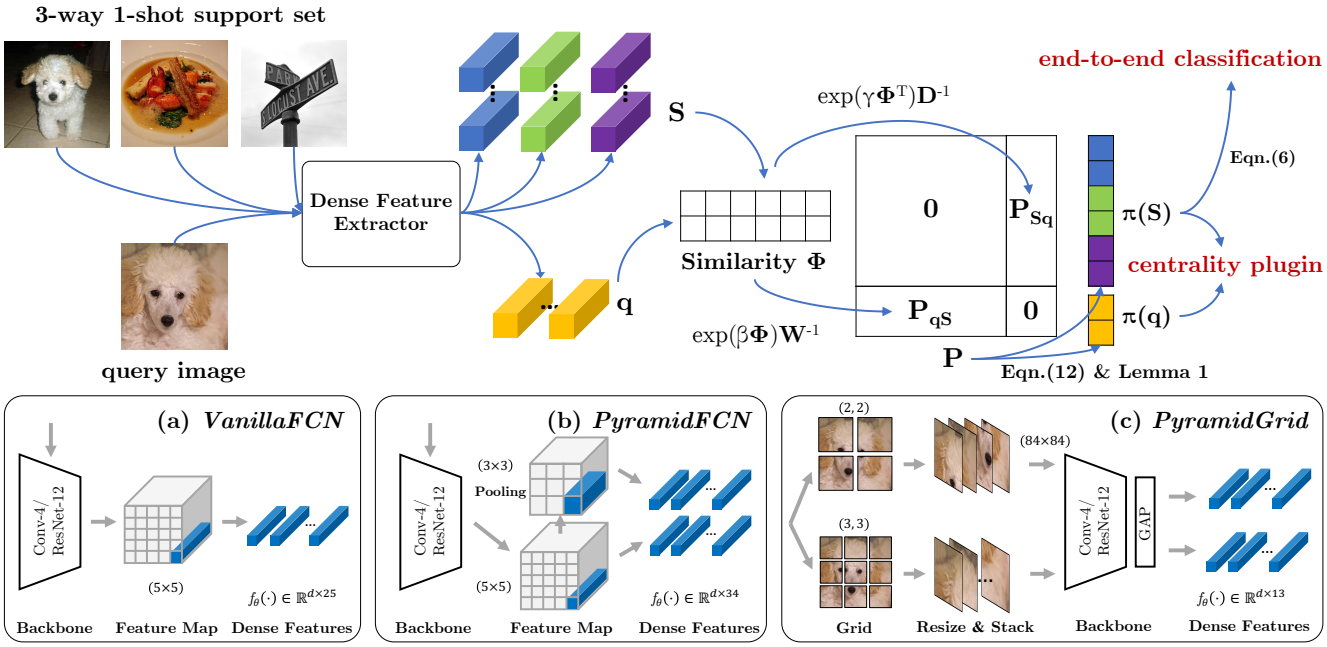


Figure 2. Our framework for 3-way 1-shot classification. Given the query and support images, we first extract their dense features $\mathbf{q}$ and $\mathbf{S}$ which are two sets of feature vectors. Then we calculate their similarity matrix $\mathbf{\Phi}$. After that, we build probability matrix $\mathbf{P}$ based on the scaled and column-normalized similarity matrix. We obtain the stationary distributions $\boldsymbol{\pi}(\mathbf{S})$ and $\boldsymbol{\pi}(\mathbf{q})$ by the Katz approximation in Eqn.(12) and the Lemma 1. Finally, we use them for both end-to-end FSL and as a centrality plugin in global feature based FSL. We explore three dense feature extractors (*i.e.*, *VanillaFCN*, *PyramidFCN*, *PyramidGrid*) inspired by [31] to extract dense feature representations.

3.6. Plugin for Global Feature based FSL.

Since the underlying centrality learned by MCL reveals the different importance of local features on an affiliation network, it is thus plausible to plug it into global feature based methods by replacing their native global average pooling (GAP) with the centrality weighted pooling as follows:

a. Extract dense features for each input image.

b. Compute the eigenvector centrality $\mathbf{x}^{\text{Eigen}}$ by Eqn.(12).

c. Compute $\boldsymbol{\pi}(\mathbf{S})$ and $\boldsymbol{\pi}(\mathbf{q})$ according to definition of the single-mode centrality in Lemma 1.

d. Use $\boldsymbol{\pi}(\mathbf{q})$ as weights on query dense features and weighted accumulate them to a single feature vector.

e. Use the class-wise normalized centrality from $\boldsymbol{\pi}(\mathbf{S})$ to accumulate dense features for each supporting class.

Once we get the centrality weighted features for the query image and support classes, it is straightforward to perform traditional global feature based methods like before.

4. Experiments

We conduct experiments on two widely-used FSL datasets and three fine-grained datasets: (1) ***mini*ImageNet** [26] contains 600 images per class over 100 classes. We follow the split used by [20] that takes 64, 16 and 20 classes for train/val/test respectively; (2) ***tiered*ImageNet** [19] is much larger compared to *mini*ImageNet with 608 classes. The 351, 97 and 160 classes are used for train/val/test respectively. (3) **CUB** [27] consists 11,788 images from 200 bird classes. 100/50/50 classes are used for train/val/test and each image is first cropped to a human-annotated bounding box. (4) **meta-iNat** [28] is a benchmark of animal species in the wild. We follow the same class split proposed by [28] that uses 908/227 classes for training/evaluation respectively. (5) ***tiered*-meta-iNat** [28] is a more difficult version of meta-iNat where a large domain gap is introduced. The 354 test classes are populated by insects and arachnids, while the remaining 781 classes form the training set.

Backbone networks. We conduct experiments with both widely-used four layer convolutional Conv-4 [26] and deep ResNet-12 [24] backbones. As is commonly implemented in the state-of-the-art FSL literature, we adopt a pre-training stage for the ResNet-12 before the episode meta-training while directly meta-train from scratch for the simple Conv-4.

Dense feature extractor $f_\theta(\cdot)$. We explore three dense feature extractors in our experiments as shown in Figure 2: (1) *VanillaFCN* simply treats the feature map output of fully convolutional network as dense features. (2) *PyramidFCN* uses an extra adaptative average pooling layer to obtain 34 dense features for each image. (3) *PyramidGrid* crops the image evenly into the grid of size $2 \times 2 + 3 \times 3$ and encodes each grid cell to a feature vector individually. The feature vectors from all the cells constitute the set of dense features.

Method		Conv-4				ResNet-12			
		*mini*ImageNet		*tiered*ImageNet		*mini*ImageNet		*tiered*ImageNet	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Global features	DSN [22]	51.78	68.99	53.22	71.06	62.64	78.83	67.39	82.85
	MetaOptNet [10]	52.87	68.76	54.71	71.76	62.64	78.63	65.99	81.56
	Negative Margin [15]	52.84	70.41	-	-	63.85	81.57	-	-
	FEAT [30]	55.15	71.61	-	-	66.78	82.05	70.80	84.79
	Meta-Baseline [4]	-	-	-	-	63.17	79.26	68.62	83.29
	RelationNet‡ [25]	52.12	66.90	54.33	69.95	60.97	75.12	64.71	78.41
	RelationNet+MCL	*54.50*	*70.63*	*57.73*	*74.46*	*61.70*	*75.53*	*65.93*	*80.27*
	ProtoNet‡ [23]	52.32	69.74	53.19	72.28	62.67	77.88	68.48	83.46
	ProtoNet+MCL	*54.31*	*69.84*	*56.67*	*74.36*	*64.40*	*78.60*	*70.62*	*83.84*
Dense features (*VanillaFCN*)	DN4‡ [12]	54.66	71.26	56.86	72.16	65.35	81.10	69.60	83.41
	DeepEMD† [31]	52.15	65.52	50.89	66.12	65.91	82.41	71.16	83.95
	FRN† [29]	54.87	71.56	55.54	74.68	66.45	82.83	71.16	86.01
	Label Propagation	52.24	67.68	54.56	70.08	65.00	80.07	71.12	83.89
	MCL (ours)	55.38	70.02	57.63	74.25	67.36	83.63	71.76	86.01
	MCL-Katz (ours)	**55.55**	**71.74**	**57.78**	**74.77**	**67.51**	**83.99**	**72.01**	**86.02**
Dense features (*PyramidFCN*)	DN4‡ [12]	54.54	70.94	57.05	72.90	63.54	79.04	71.10	84.22
	DeepEMD‡ [31]	50.67	64.94	51.26	65.64	66.27	82.41	70.76	84.20
	FRN‡ [29]	54.40	70.75	57.30	**75.58**	65.94	81.97	70.56	85.44
	Label Propagation	53.38	68.69	55.21	71.22	65.71	75.78	71.00	78.01
	MCL (ours)	55.13	70.77	57.93	74.36	67.45	84.36	72.01	86.31
	MCL-Katz (ours)	**55.77**	**71.24**	**58.20**	74.73	**67.85**	**84.47**	**72.13**	**86.32**
Dense features (*PyramidGrid*)	DN4‡ [12]	57.17	70.91	56.71	70.92	67.86	80.08	71.29	82.60
	DeepEMD‡ [31]	55.68	70.75	55.88	70.06	67.83	81.32	73.13	84.18
	FRN‡ [29]	55.80	71.52	55.68	72.87	67.00	82.20	71.42	85.58
	Label Propagation	55.12	68.43	56.05	72.39	67.18	81.07	73.18	85.19
	MCL (ours)	57.50	73.03	57.57	73.81	**69.31**	**85.11**	**73.62**	**86.29**
	MCL-Katz (ours)	**57.88**	**74.03**	**57.63**	**73.96**	69.25	84.71	73.38	86.21

Table 1. Few-shot classification accuracy (%) on *mini*ImageNet and *tiered*ImageNet. The 95% confidence intervals are all below 0.2 for the 10,000 episodes evaluation. Results of *italic* font indicates the performance of MCL as a plugin. Results of **bold** fonts are the best results for different dense feature extractors, respectively. We re-implement our plugin baseline methods (*i.e.*, ProtoNet and RelationNet) as well as the competitive dense features based methods (*i.e.*, DN4, DeepEMD and FRN) in our unified framework in case their performances are not available under specific settings. †: re-implemented results on Conv-4 backbone, ‡: re-implemented results on both backbones.

4.1. General Few-shot Classification Results

Table 1 details the comparisons of MCL with global feature based methods [4, 10, 15, 22, 30] as well as dense feature based methods [12, 29, 31] on *mini-*/*tiered*ImageNet.

Besides these methods, we also exploit label propagation [32] in dense feature based inductive FSL like in [13] to demonstrate that traditional transductive methods could be easily adapted to inductive settings if we treat dense feature vectors as a set of unlabeled data. As shown, our MCL is highly competitive with recent state-of-the-art results. In particular, our MCL-Katz achieves **67.51**% (1-shot) and **83.99**% (5-shot) on *mini*ImageNet with the simplest *VanillaFCN*.

Comparisons with dense feature based methods. The last three panes of Table 1 illustrate that proposed bidirectional MCL outperforms query-to-support DN4's nearest neighboring, DeepEMD's optimal matching and FRN's latent feature reconstructions in various tasks. Although the optimal matching flow is unidirectional in DeepEMD, they adopt a cross-reference attention that encodes the bidirectional relevance to a certain extent. If we treat dense features as *nodes* in the graph and use their similarities as *edges*, a major difference is that they treat nodes equally to derive different weights on edges while we use the fixed edges to derive the centrality of nodes directly for classifications.

Comparisons with global feature based methods. The centrality investigated in this work is also capable to be plugged into global feature based methods as local features are not equally important before global average pooling. As shown in the first pane of Table 1, our proposed centrality weighted pooling could provide at most **3.4**% and **4.5**% performance gains for ProtoNet and RelationNet respectively. We show in Table 4 that the improvements are also consistent if we solely apply it on the query and support dense features.

Method	CUB				meta-iNat				*tiered*-meta-iNat			
	Conv-4		ResNet-12		Conv-4		ResNet-12		Conv-4		ResNet-12	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ProtoNet [23]	71.64	81.50	82.98	91.38	53.78	73.80	76.47	90.20	35.47	54.85	47.61	71.06
ProtoNet + MCL	*76.87*	*86.95*	*84.59*	*91.71*	*57.52*	*74.33*	*78.75*	*90.53*	*39.09*	*59.16*	*51.22*	*72.50*
DSN [22]	66.01	85.41	80.80	91.19	58.08	77.38	78.80	89.77	36.82	60.11	46.61	72.79
CTX [6]	69.64	87.31	78.47	90.90	60.03	78.80	69.04	88.39	36.83	60.84	42.91	69.88
DN4 [12]	78.31	88.43	85.44	92.51	62.32	79.76	79.58	91.37	43.82	64.17	48.99	72.29
DeepEMD [31]	75.34	85.68	83.35	91.60	54.48	68.36	76.05	86.82	36.05	48.55	48.14	66.27
FRN [29]	73.48	88.43	83.16	92.59	62.42	80.45	73.52	91.83	43.91	63.36	48.86	76.62
MCL	77.80	88.71	83.64	92.18	**64.66**	**81.31**	**80.17**	91.59	**44.08**	**64.61**	**51.35**	**76.87**
MCL-Katz	**79.61**	**90.56**	**85.63**	**93.18**	63.92	81.09	79.34	**91.84**	44.00	64.24	49.68	76.05

Table 2. Few-shot classification (%) results on fine-grained datasets. The 95% confidence intervals are below 0.19 within 10,000 episodes.

	$f_\theta(\cdot)$	*mini*ImageNet		*tiered*ImageNet	
		1-shot	5-shot	1-shot	5-shot
unidirectional	*VanillaFCN*	54.88	69.98	55.31	68.87
MCL		55.38	70.02	57.63	74.25
MCL-Katz		**55.55**	**71.74**	**57.78**	**74.77**
unidirectional	*PyramidFCN*	54.92	69.17	55.69	69.26
MCL		55.13	70.77	57.93	74.36
MCL-Katz		**55.77**	**71.24**	**58.20**	**74.73**
unidirectional	*PyramidGrid*	56.61	71.40	55.12	69.19
MCL		57.50	73.03	57.57	73.81
MCL-Katz		**57.88**	**74.03**	**57.63**	**73.96**

Table 3. Comparisons between the *unidirectional* random walks and *bidirectional* MCL on *mini*-/*tiered*ImageNet with Conv-4.

	+MCL		*mini*ImageNet		*tiered*ImageNet	
	$\mathbf{q}$	$\mathbf{s}$	1-shot	5-shot	1-shot	5-shot
ProtoNet [23]			52.32	69.74	53.19	72.28
	✓		53.74	69.61	55.36	73.17
		✓	53.97	69.74	56.60	74.18
	✓	✓	**54.31**	**69.84**	**56.67**	**74.36**
RelationNet [25]			52.12	66.90	54.33	69.95
	✓		53.25	66.79	54.46	70.71
		✓	54.37	70.57	57.68	74.24
	✓	✓	**54.50**	**70.63**	**57.73**	**74.46**

Table 4. Ablation of MCL as plug-and-play that individually applied on query features $\mathbf{q}$ and support features $\mathbf{s}$. The experiments are conducted with Conv-4 and *VanillaFCN*

4.2. Fine-grained Few-shot Classification Results

We follow the same setting (shown in supplementary) as in FRN [29] to train all the models from scratch on those fine-grained datasets. We re-implement DN4 [12], DeepEMD [31] to thoroughly compare with dense features based methods. Apart from the basic comparisons (both backbones on CUB and Conv-4 backbone on meta-iNat/*tiered*-meta-iNat) in [29], we conduct extra ResNet-12 experiments for all the comparing methods on meta-iNat and *tiered*-meta-iNat in our unified framework for a thorough comparison.

Results in Table 2 demonstrate that both our end-to-end MCL (*VanillaFCN*) and the centrality plugin are broadly effective in various fine-grained few-shot classification tasks.

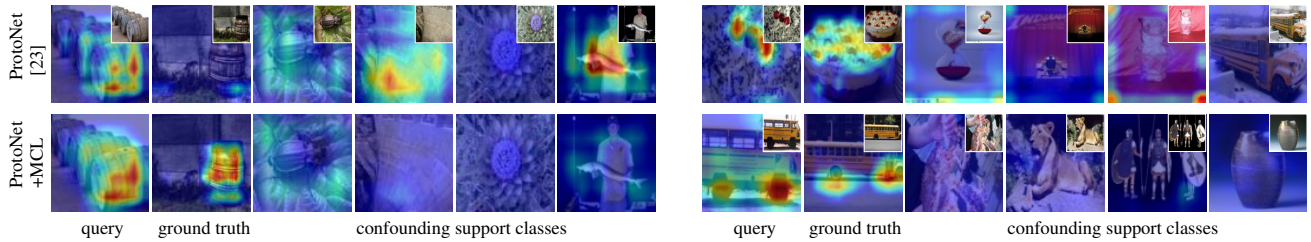
5. Analysis

5.1. Ablation Study

Different choices of γ,β. Since $\mathbf{\Phi}$ is non-square due to the different cardinalities of set $\mathbf{S}$ and $\mathbf{q}$, we use two different scaling parameters in Eqn.(1) and Eqn.(2). By the column-wise normalization, γ and β can be interpreted as the reciprocal of temperatures in softmax random walk probability. Large γ, β will have a hard random walk probability that leads to a concentrated centrality in the affiliation network. However, extremely large γ, β (*e.g.*, one-hot probability when they approaching infinity) would inevitably bias the episodic training due to the potential gradient explosion.

Since the cardinality of set $\mathbf{S}$ is larger than that of $\mathbf{q}$, we empirically use a larger γ than β as we need a harder probability in random walks from query to the large set of support features. In the experiments, we carefully select γ and β (generally γ=20 and β=10 for 1-shot models and γ=40 and β=20 for 5-shot models) from several combinations of parameters by their validation performance.

Influence of Katz attenuation factor α. When α is small, the contribution given by paths longer than one rapidly declines, and the centrality will be determined by short paths (mostly in-degrees). When α is large, long paths are devalued smoothly, and the centrality would be more influenced by endogenous topology. When α approaching 0, the classification will be only determined by the unidirectional query-to-support random walk where the contributions from paths longer than one just vanished. In the experiments, we simply use $\alpha = 0.5$ for MCL-Katz methods, and the performances could be further improved by selecting different α.



(a) MCL as a plugin for feature centralization

(b) MCL as an end-to-end classification method

Table 5. Grad-CAM visualizations of query and support images in 5-way 1-shot tasks. The left pane (a) illustrates MCL as a plugin on the ProtoNet. The right pane (b) illustrates MCL as an end-to-end classification method. Images (from *mini*ImageNet) at the second column of each pane are from the ground truth class while the four on the far right are the confounding support classes.

5.2. Computational Speed

Intuitively, bidirectional methods would be slower than unidirectional ones since their back-and-forth computations. However, our MCL would definitely reach to the periodic equilibrium state as shown in Eqn.(5). Thus, we could directly solve it by the Katz approximation.

At first glance, the most expensive step in Eqn.(10) is inverting $\boldsymbol{I}-\alpha\mathbf{P}$ that costs $O(N^3r^3)$ where N is the number of supporting classes and r is the number of dense features for each image input. Fortunately, $\boldsymbol{I}-\alpha\mathbf{P}$ is a very special matrix that equals a diagonal matrix minus an anti-diagonal matrix, and we can reformulate it by block-wise inversion:

$$
\begin{aligned}
(\boldsymbol{I}-\alpha\mathbf{P})^{-1} &= \begin{pmatrix} \boldsymbol{I} & -\alpha\mathbf{P}_{\mathbf{Sq}} \\ -\alpha\mathbf{P}_{\mathbf{Sq}} & \boldsymbol{I} \end{pmatrix}^{-1} \\
&= \begin{pmatrix} \boldsymbol{I}+\alpha^2\mathbf{P}_{\mathbf{Sq}}\boldsymbol{\Delta}^{-1}\mathbf{P}_{\mathbf{qS}} & \alpha\mathbf{P}_{\mathbf{Sq}}\boldsymbol{\Delta}^{-1} \\ \alpha\boldsymbol{\Delta}^{-1}\mathbf{P}_{\mathbf{qS}} & \boldsymbol{\Delta}^{-1} \end{pmatrix} \qquad (14)
\end{aligned}
$$

where $\boldsymbol{\Delta}$ is defined by $\boldsymbol{I}-\alpha^2\mathbf{P}_{\mathbf{Sq}}\mathbf{P}_{\mathbf{qS}}$. The most expensive step then becomes a inversion of $\boldsymbol{\Delta}\in\mathbb{R}^{r\times r}$ and the time complexity is reduced to $O(r^3)$.

As shown in Table 6, we compare the inference speed for different image resolutions on 5-way 1-shot FSL tasks with ResNet-12. It can be observed that our bidirectional MCL did not introduce many computational overheads compared to unidirectional methods, and Katz approximation does accelerate computing the stationary distribution.

5.3. Qualitative Feature Centralization

To better understand our centrality based methods, we visualize the Grad-CAM [21] of the last convolutional layer in ResNet-12 for both plugin and end-to-end methods. We show in Table 5(a) that our MCL helps ProtoNet concentrate on the most relevant regions of interest. Table 5(b) demonstrates that the most centralized support features in MCL are not only from the ground truth but also mutually affiliated with task-relevant objects in the query images. We consider it qualitatively validates our underlying idea that support features would be frequently visited in the long times of bidirectional random walks if they shared the same local characteristics with the query image.

Model	*VanillaFCN* resolution r			
	5×5	8×8	10×10	12×12
DN4 [12]	11.16	13.41	16.82	23.45
DeepEMD [31]	262.0	814.6	2781.9	8035.4
FRN [29] (Ridge regression)	14.55	20.42	25.81	30.65
FRN [29] (Woodbury identity)	36.74	42.33	47.31	55.84
MCL (`torch.pinverse`)	55.91	119.0	292.2	464.8
MCL (QR decomposition)	600.0	658.6	719.4	802.6
MCL (Katz approximation)	12.59	14.59	18.92	29.10

Table 6. Time per episode (ms) for different feature map resolutions r on 5-way 1-shot tasks where the feature extraction time is excluded for all comparing methods.

6. Conclusions

We present a novel dense feature based framework: Mutual Centralized Learning (MCL) to highlight the mutual affiliations of bipartite dense features in FSL. We introduce a novel dense features' accessibility and demonstrate classic transductive methods like label propagation could be easily adapted to the inductive setting if we treat dense features from a single image as the set of unlabeled data. We use bidirectional random walk to learn mutual affiliations in FSL and prove its features' accessibility in the time-homogeneous Markov process equivalent to the single-mode eigenvector centrality of an affiliation network. We show that such centrality could not only serve as a noval end-to-end classification criterion but also as a plugin in existing methods.

Acknowledgements

This work was supported in part by The National Key Research and Development Program of China (Grant Nos: 2018AAA0101400), in part by The National Nature Science Foundation of China (Grant Nos: 62036009, U1909203, 61936006, 62133013), in part by Innovation Capability Support Program of Shaanxi (Program No. 2021TD-05).

References

[1] Phillip Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of mathematical sociology*, 2(1):113–120, 1972.

[2] Phillip Bonacich. Simultaneous group and individual centralities. *Social networks*, 13(2):155–168, 1991.

[3] Stephen P Borgatti and Martin G Everett. Network analysis of 2-mode data. *Social networks*, 19(3):243–269, 1997.

[4] Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. A new meta-baseline for few-shot learning. *arXiv preprint arXiv:2003.04390*, 2020.

[5] Kaize Ding, Jianling Wang, Jundong Li, Kai Shu, Chenghao Liu, and Huan Liu. Graph prototypical networks for few-shot learning on attributed networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 295–304, 2020.

[6] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. *arXiv preprint arXiv:2007.11498*, 2020.

[7] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

[8] Fusheng Hao, Fengxiang He, Jun Cheng, Lei Wang, Jianzhong Cao, and Dacheng Tao. Collect and select: Semantic alignment metric learning for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8460–8469, 2019.

[9] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[10] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019.

[11] Wenbin Li, Lei Wang, Jing Huo, Yinghuan Shi, Yang Gao, and Jiebo Luo. Asymmetric distribution measure for few-shot learning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 2957–2963, 2021.

[12] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7260–7268, 2019.

[13] Yann Lifchitz, Yannis Avrithis, and Sylvaine Picard. Local propagation for few-shot learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10457–10464. IEEE, 2021.

[14] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc. Dense classification and implanting for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9258–9267, 2019.

[15] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. In *European Conference on Computer Vision*, pages 438–455. Springer, 2020.

[16] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *International Conference on Learning Representations*, 2019.

[17] Yang Liu, Tu Zheng, Jie Song, Deng Cai, and Xiaofei He. Dmn4: Few-shot learning via discriminative mutual nearest neighbor neural network. *arXiv preprint arXiv:2103.08160*, 2021.

[18] Oskar Perron. Zur theorie der matrices. *Mathematische Annalen*, 64(2):248–263, 1907.

[19] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018.

[20] Ravi Sachin and Larochell Hugo. Optimization as a model for few-shot learning. *ICLR*, 2017.

[21] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[22] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *CVPR*, pages 4136–4145, 2020.

[23] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, pages 4077–4087, 2017.

[24] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019.

[25] 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*, pages 1199–1208, 2018.

[26] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3637–3645, 2016.

[27] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

[28] Davis Wertheimer and Bharath Hariharan. Few-shot learning with localization in realistic settings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6558–6567, 2019.

[29] Davis Wertheimer, Luming Tang, and Bharath Hariharan. Few-shot classification with feature map reconstruction networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8012–8021, 2021.

[30] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *CVPR*, pages 8808–8817, 2020.

[31] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Differentiable earth mover's distance for few-shot learning. *arXiv e-prints*, pages arXiv–2003, 2020.

[32] Dengyong Zhou, Olivier Bousquet, Thomas N. Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, volume 16, pages 321–328, 2003.