Z-Score Normalization, Hubness, and Few-Shot Learning

Nanyi Fei[2,3] Yizhao Gao[1,2] Zhiwu Lu[1,2,*] Tao Xiang[4]
[1]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
[2]Beijing Key Laboratory of Big Data Management and Analysis Methods
[3]School of Information, Renmin University of China, Beijing, China
[4]Department of Electrical and Electronic Engineering, University of Surrey, Guildford, UK
{feinanyi, luzhiwu}@ruc.edu.cn

Abstract

The goal of few-shot learning (FSL) is to recognize a set of novel classes with only few labeled samples by exploiting a large set of abundant base class samples. Adopting a meta-learning framework, most recent FSL methods meta-learn a deep feature embedding network, and during inference classify novel class samples using nearest neighbor in the learned high-dimensional embedding space. This means that these methods are prone to the hubness problem, that is, a certain class prototype becomes the nearest neighbor of many test instances regardless which classes they belong to. However, this problem is largely ignored in existing FSL studies. In this work, for the first time we show that many FSL methods indeed suffer from the hubness problem. To mitigate its negative effects, we further propose to employ z-score feature normalization, a simple yet effective transformation, during meta-training. A theoretical analysis is provided on why it helps. Extensive experiments are then conducted to show that with z-score normalization, the performance of many recent FSL methods can be boosted, resulting in new state-of-the-art on three benchmarks.

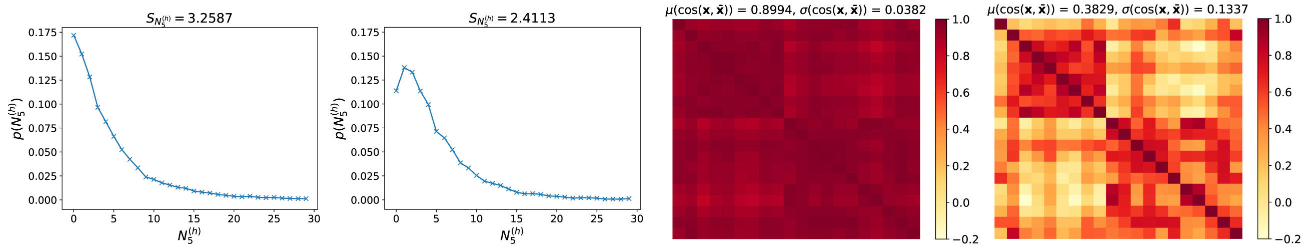
1. Introduction

In recent years, the advances of deep convolutional neural networks (CNNs) have had profound impacts on a variety of vision areas, such as object recognition [45, 38, 13], semantic segmentation [26, 4], and even image generation [33, 42]. To train an effective CNN model for visual recognition, a large number of manually labeled training samples are often required. However, obtaining sufficient training data is often expensive and sometimes even infeasible (e.g., for rare object categories). One solution to the data hungry nature of deep recognition models is few-shot learning (FSL) [20, 21], which aims to recognize a set of novel object classes with only few labeled samples by exploiting a set of base classes each containing ample samples.

Recent FSL methods typically follow the meta-learning framework and adopt episodic training [8, 47, 49, 2, 19, 9, 59, 7, 63]. That is, they train their models over a large number of meta-tasks/episodes sampled from the abundant base class images. This is to imitate the few-shot classification tasks for the novel classes. Specifically, each episode is constructed by sampling N base/novel classes with K labeled samples in each class as the support set and a set of query images to be classified. Existing meta-learning methods differ in which part of the recognition model, comprising a feature embedding network and a classifier, is meta-learned. It is noted that most recent FSL methods [47, 2, 59, 63] focus on meta learning the embedding network. Once the model learned, during inference, the support set samples are used to construct class prototypes in that embedding space, and the classification of query samples is done by the simple nearest neighbor (NN) search.

Using NN in a high dimensional embedding space makes these FSL methods prone to the *hubness problem* [34, 43, 50]. Specifically, in a high dimensional space, nearest neighbor suffers from the existence of *hubs*, i.e., the class prototypes which are the nearest neighbors of many test samples, regardless which classes they belong to. These hubs thus clearly harm the recognition performance. To illustrate the hubness problem, let us take a concrete example. Denote k-occurrence $N_k^{(h)}(x)$ as the number of times that a sample x occurs among the k nearest neighbors of all other points in a dataset. We visualize the distribution of $N_5^{(h)}$ on the test set of *mini*ImageNet [52] in Figure 1(a), where a four-block CNN Conv4-64 pre-trained on the training set is used. We also calculate the hubness measure skewness $S_{N_5^{(h)}}$ of the distribution (see the detailed definition in Section 4.3). It can be observed that the distribution is heavily skewed to the right, i.e., a large number of samples have low $N_5^{(h)}$ values while a small group of samples are frequently

[*] Corresponding author.



(a) Before ZN (b) After ZN (c) Before ZN (d) After ZN

Figure 1. Visualizations on the test set of *mini*ImageNet with a total of 12,000 samples using Conv4-64 pre-trained on the training set. (a) – (b) Visualizations of the distributions of $N_5^{(h)}$ using the original features and the z-score normalized (ZN) features, respectively. $S_{N_5^{(h)}}$ denotes the skewness of a distribution, whose absolute value is larger when the distribution is more skewed. (c) – (d) Visualizations of the cosine similarities among 20 class centers using the original features and the z-score normalized features, respectively. $\mu(\cos(\mathbf{x}, \bar{\mathbf{x}}))$ and $\sigma(\cos(\mathbf{x}, \bar{\mathbf{x}}))$ are respectively the mean and standard deviation of the cosine similarity between a sample and the dataset mean.

visited. The same observation holds when the pre-trained embedding network is meta-learned using recent FSL methods [47, 2, 63]. This provides direct evidence that the hubness problem indeed exists in FSL. However, as far as we know, this problem has been largely ignored.

In order to remedy the problem, we must first identify the potential causes for it. It is discovered that one cause for hubness is actually the widely used batch normalization (BN) [15] and non-negative activation functions (e.g., ReLU) in the deep embedding CNNs. In particular, we find that with BN and ReLU, the output feature vectors with non-negative elements often have similar directions in the feature space. To show this, in Figure 1(c), we visualize the cosine similarities among 20 class centers (i.e., feature mean of samples belonging to the same classes) also on the test set of *mini*ImageNet using the pre-trained Conv4-64. Besides, we calculate the cosine similarities of all samples to the dataset mean (i.e., the mean of all feature vectors) and obtain the statistical mean and standard deviation. We can clearly see from Figure 1(c) that these feature vectors are very much alike in terms of the direction, meaning samples of different classes can form clusters. This problem is supposed to be rectified by the subsequent classification layer. However, with NN search in metric-based FSL and no classification layer for the rescue, it must be addressed.

Our solution to the hubness problem in FSL is thus on deploying alternative normalization strategies during pre-training or episodic training. Particularly, we discover that *z-score normalization (ZN)*, a simple transforming operation at the feature level, can offer an effective solution. More concretely, with ZN, for each feature vector extracted by the embedding network, every component of it first subtracts the mean of all components and then is divided by the standard deviation of all components. Note that ZN is applied to each feature vector independently during both training and inference, and thus the inductive FSL setting is still strictly followed in this paper. We visualize the distribution of $N_5^{(h)}$ after applying ZN in Figure 1(b) and also the heat map of cosine similarities calculated with normalized features in Figure 1(d). From Figure 1(b), we can see that the distribution of the 5-occurrence is pulled back to the left and the value of skewness is much smaller. From Figure 1(d), we can also observe that the samples of different classes in the normalized embedding space are more separable. We show that this simple operation works during both pre-training and episodic training (see Table 1).

Our main contributions are three-fold: (1) To the best of our knowledge, we are the first to bring to light the hubness problem in the context of FSL. (2) We propose to alleviate the negative effects of the hubness problem in FSL by employing the z-score feature normalization. We also provide theoretical analysis on why it works. (3) Comprehensive experiments are carried out to demonstrate that the simple ZN operation can boost a variety of embedding/metric-based FSL methods which dominated the state-of-the-art lately. The code and models will be released soon.

2. Related Work

Few-Shot Learning. Most recent few-shot learning (FSL) approaches [52, 35, 8, 47, 49, 30, 2, 59, 7, 63] are based on meta-learning using an episodic training strategy. They can be categorized into four groups: metric-based, model-based, optimization-based, and generation-based ones. (1) Metric-based methods try to learn suitable distance metrics for nearest neighbor search based classification. They either learn one embedding space for their chosen/designed metrics [52, 47, 49, 28] (e.g., Euclidean distance) or directly learn the metric [49, 56, 3, 40, 16, 58]. Instead of embedding all samples into a shared task-agnostic metric space, [60, 31, 59, 44] further learn task-adaptive metric spaces for FSL. (2) Model-based methods [8, 29, 39] aim to learn good model initialization using the base classes, in order to quickly fine-tune them with a limited number of gradient update steps on novel classes using only few labeled samples. (3) Optimization-based methods [35, 27, 24] meta-learn novel optimization algorithms instead of the standard gradient descent, again for quick adaptation from base to novel classes. (4) Generation-based methods meta-learn generators on base classes to either generate additional novel class samples [12, 55, 41, 22] or directly generate network parameters [32, 10, 11] based on the few shots of novel classes. In this paper, we mainly focus on embedding/metric-based FSL methods with nearest neighbor classifiers which suffer from the hubness problem. We show that a simple z-score feature normalization can improve their performance, often by considerable margins.

Feature Normalization for FSL. Normalization is universal and also essential in deep neural networks (e.g., Batch Normalization [15] in CNNs). In this paper, we focus on the effects of normalization at the final feature level for FSL. For meta-learning based FSL, Nguyen et al. [28] propose SEN which forces equal l_2 norms on all samples and modifies the Euclidean distance metric accordingly to learn features with similar norms. However, we find that applying z-score normalization (ZN) is simpler yet more effective for FSL. Importantly, we are motivated to solve the hubness problem while SEN is not. For non-meta-learning based FSL, several methods [25, 51] also employ l_2 normalization. They carefully design training algorithms to pre-train their models on base classes and then directly fine-tune them with few novel class samples without meta-learning. In contrast, we show that both pre-training and meta-learning can benefit from our ZN and having both steps leads to better performance. Additionally, the most related work to ours is SimpleShot [54], and we have the following differences: (1) We discover the existence of the hubness problem in FSL and provide a theoretical analysis while SimpleShot does neither. (2) We propose to address the hubness problem by adopting ZN based on the analysis while SimpleShot propose to adopt CL2N (i.e., centered l_2 normalization) with no reason. (3) Our choice ZN is performed on top of each feature vector independently, which is flexible enough to be applied over both pre-trained models (see Table 4) and meta-learning based methods; as for CL2N in SimpleShot, since it transforms *test sample features* by subtracting the mean of *whole training set features* before l_2 normalization, it can only be used over pre-trained/trained models rather than being integrated into the meta-training process. (4) ZN is insensitive to the train-test data distribution gap while SimpleShot is sensitive to it (see our analysis in Section 4.3).

Hubness Problem. The hubness problem is first studied in [34]. Following this prior work, hubness is then studied in the context of zero-shot learning (ZSL) [43, 6, 62] and natural language processing (NLP) [50, 46, 18]. In the field of FSL, although metric-based methods typically employ nearest neighbor classifiers in a high dimensional embedding space, the hubness problem has attracted little attention. In this paper, we not only show that hubness does exist in FSL, but also provide a z-score normalization based solution and theoretical analysis on how it works.

3. Methodology

3.1. Preliminary

We first give a formal definition of the few-shot learning (FSL) problem. Let $\mathcal{C}_b$ denote a set of base classes and $\mathcal{C}_n$ denote a set of novel classes, where $\mathcal{C}_b \cap \mathcal{C}_n = \emptyset$. We are then given a large sample set $\mathcal{D}_b$ from $\mathcal{C}_b$, a few-shot sample set $\mathcal{D}_n$ from $\mathcal{C}_n$, and a test set $\mathcal{T}$ also from $\mathcal{C}_n$, where $\mathcal{D}_n \cap \mathcal{T} = \emptyset$. Concretely, $\mathcal{D}_b = \{(I_i, y_i)|y_i \in \mathcal{C}_b; i = 1, 2, \cdots, N_b\}$, where I_i denotes the i-th image, y_i is the class label of I_i, and N_b denotes the number of images in $\mathcal{D}_b$. Similarly, the K-shot (i.e., each novel class only has K labeled images) sample set $\mathcal{D}_n = \{(I_i, y_i)|y_i \in \mathcal{C}_n; i = 1, 2, \cdots, N_n\}$, where $N_n = K|\mathcal{C}_n|$. The goal of FSL is thus to predict the labels of test images in $\mathcal{T}$ by exploiting the abundant base class sample set $\mathcal{D}_b$ and the few-shot novel class sample set $\mathcal{D}_n$.

Recent meta-learning based FSL methods mostly adopt an episodic training strategy, which means that their models are trained over multiple base class classification meta-tasks (or episodes) sampled from $\mathcal{C}_b$ (i.e., only the base class samples in $\mathcal{D}_b$ are used for meta-training). The learned models are then evaluated over novel class episodes randomly sampled from $\mathcal{C}_n$. Specifically, to form an N-way K-shot Q-query episode $e = \mathcal{S} \cup \mathcal{Q}$, a subset $\mathcal{C}$ containing N classes is first randomly sampled from $\mathcal{C}_b$ during meta-training (or from $\mathcal{C}_n$ during meta-test). A support set $\mathcal{S} = \{(I_i, y_i)|y_i \in \mathcal{C}; i = 1, 2, \cdots, N \times K\}$ and a query set $\mathcal{Q} = \{(I_i, y_i)|y_i \in \mathcal{C}; i = 1, 2, \cdots, N \times Q\}$ ($\mathcal{S} \cap \mathcal{Q} = \emptyset$) are then generated by sampling K support and Q query images from each class in the subset $\mathcal{C}$, respectively.

We employ Prototypical Network (ProtoNet) [47] to introduce the hubness problem and formulate our solution. ProtoNet is chosen because it is simple yet effective and underpins many recent SOTA FSL methods [2, 59, 63]. Concretely, ProtoNet first obtains a prototype for each class in an N-way K-shot episode by computing the mean representation of support samples from each class:

$$\mathbf{p}_c = \frac{1}{|\mathcal{S}_c|} \sum_{(I,y) \in \mathcal{S}_c} f_\phi(I), \tag{1}$$

where $\mathbf{p}_c$ denotes the prototype of class $c \in \mathcal{C}$, $\mathcal{S}_c = \{(I, y) \in \mathcal{S}|y = c\} \subseteq \mathcal{S}$ denotes the set of support samples from class c ($|\mathcal{S}_c| = K$), and f_ϕ is a feature extractor with learnable parameters ϕ whose output dimension is D (i.e., $\mathbf{x} = f_\phi(I) \in \mathbb{R}^D$). For each query image I in $\mathcal{Q}$, ProtoNet then computes the distances to all class prototypes and obtains the probabilities over N classes based on softmax:

$$\psi_c(\mathbf{x}) = \frac{\exp(-d(\mathbf{x}, \mathbf{p}_c))}{\sum_{c' \in \mathcal{C}} \exp(-d(\mathbf{x}, \mathbf{p}_{c'}))}, \ \mathbf{x} = f_\phi(I), \quad (2)$$

where $\psi_c(\mathbf{x})$ denotes the probability that $\mathbf{x}$ belongs to class c ($\sum_{c \in \mathcal{C}} \psi_c(\mathbf{x}) = 1$), and $d : \mathbb{R}^D \times \mathbb{R}^D \to [0, +\infty)$ is the squared Euclidean distance in the feature space. Specifically, we use the temperature scaling technique in all of our experiments when computing the distances in order to find the suitable scale for the metric:

$$d(\mathbf{x}, \mathbf{p}_c) = \|\mathbf{x} - \mathbf{p}_c\|^2 / T, \quad (3)$$

where T is the temperature hyper-parameter.

The classification loss of ProtoNet for each meta-training episode e is then defined as the negative log-probability of the true class of each query sample:

$$L_{fsl}(e) = \frac{1}{|\mathcal{Q}|} \sum_{(I,y) \in \mathcal{Q}} -\log \psi_y(f_\phi(I)). \quad (4)$$

3.2. Z-Score Normalization for FSL

As mentioned earlier, all embedding/metric learning based FSL methods are prone to the hubness problem and the BN used in the embedding network is one of the reasons for that. We thus propose to use z-score normalization to alleviate the hubness problem.

Concretely, let x_i ($i = 1, 2, \cdots, D$) denote the i-th component of each feature vector $\mathbf{x} \in \mathbb{R}^D$. We first compute the mean and the standard deviation of these D components:

$$\mu_{\mathbf{x}} = \frac{1}{D} \sum_{i=1}^{D} x_i, \quad \sigma_{\mathbf{x}} = \sqrt{\frac{1}{D} \sum_{i=1}^{D} (x_i - \mu_{\mathbf{x}})^2}. \quad (5)$$

Z-score normalization is then applied as

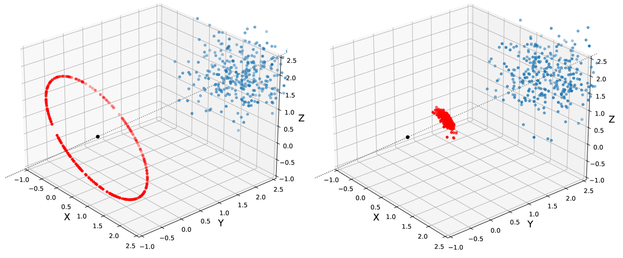
$$\mathbf{x}^{(zn)} = \mathrm{ZN}(\mathbf{x}) = \frac{\mathbf{x} - \mu_{\mathbf{x}} \mathbf{1}}{\sigma_{\mathbf{x}}} \in \mathbb{R}^D, \quad (6)$$

where $\mathbf{1} = [1, 1, \cdots, 1]^T$ is a D-dimensional vector with its components being all ones.

From the above definition of z-score feature normalization, we can obtain that

$$\begin{aligned} \|\mathbf{x}^{(zn)}\| &= \sqrt{\sum_{i=1}^{D} (\frac{x_i - \mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}})^2} \\ &= \sqrt{\frac{\sum_{i=1}^{D} (x_i - \mu_{\mathbf{x}})^2}{\frac{1}{D} \sum_{i=1}^{D} (x_i - \mu_{\mathbf{x}})^2}} = \sqrt{D}, \quad (7) \end{aligned}$$

$$\begin{aligned} < \mathbf{x}^{(zn)}, \mathbf{1} > &= \sum_{i=1}^{D} \frac{x_i - \mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}} \cdot 1 \\ &= \frac{\sum_{i=1}^{D} x_i - D\mu_{\mathbf{x}}}{\sigma_{\mathbf{x}}} = 0, \quad (8) \end{aligned}$$



(a) Z-score normalization (b) l_2 normalization

Figure 2. Examples of z-score and l_2 feature normalizations in 3D space. Blue dots are samples around the $\mathbf{1}$ vector before normalization, while red ones are those after normalization. The black dot is the origin of coordinates.

where $< \cdot, \cdot >$ denotes the dot product of two vectors. We can see from these calculations that z-score normalization first projects the original feature vectors along the $\mathbf{1}$ vector to a hyperplane which contains the origin and is perpendicular to $\mathbf{1}$. These vectors are then scaled to the same length of $\sqrt{D}$, i.e., the final normalized vectors lie on a hypersphere with the radius $\sqrt{D}$ (see Figure 2(a)).

Note that since we apply z-score normalization to each feature vector independently, the non-transductive FSL setting is still strictly followed. Once learned, with the optimal feature extractor f_ϕ found by ProtoNet+ZN (i.e., ProtoNet trained with ZN), we randomly sample multiple meta-test episodes from the set of novel classes $\mathcal{C}_n$ and then evaluate the learned model over these episodes also with ZN on top of the output features obtained by f_ϕ.

3.3. Analysis of Hubness Problem

To show the benefit of applying z-score feature normalization in addressing the hubness problem, we follow [34, 43] and explicitly study the effect of hubness with two data distributions: the normal distribution and the distribution on a hypersphere.

Normal Distribution. Let $\mathbf{a} \in \mathbb{R}^D$ denote a random vector in the D-dimensional space, with each of its channels a_i independently following a normal distribution with the expectation u_i and the variance v, i.e., $\mathbf{a} \sim \mathcal{N}(\mathbf{u}, v\mathbf{I})$, where $\mathbf{u} = [u_1, u_2, \cdots, u_D]^T$ and $\mathbf{I} \in \mathbb{R}^{D \times D}$ is the identity matrix. Let $s = \sqrt{\mathrm{Var}(\|\mathbf{a} - \mathbf{u}\|^2)}$ be the standard deviation of the squared norm of the difference between $\mathbf{a}$ and its expectation $\mathbf{u}$, where $\mathrm{Var}(\cdot)$ denotes the variation of a distribution. Consider two data points $\mathbf{a}_1$ and $\mathbf{a}_2$ randomly sampled from $\mathcal{N}(\mathbf{u}, v\mathbf{I})$, satisfying that

$$\|\mathbf{a}_1 - \mathbf{u}\|^2 - \|\mathbf{a}_2 - \mathbf{u}\|^2 = \gamma s, \quad (9)$$

where γ is a constant. We use the expected difference Δ between the squared Euclidean distances from $\mathbf{a}_1$ and $\mathbf{a}_2$ to $\hat{\mathbf{a}}$ to describe the effect of hubness problem, where $\hat{\mathbf{a}}$ is also sampled from $\mathbf{a}$:

$$\Delta = \mathbb{E}(\|\mathbf{a}_1 - \hat{\mathbf{a}}\|^2) - \mathbb{E}(\|\mathbf{a}_2 - \hat{\mathbf{a}}\|^2). \tag{10}$$

For each term $\mathbb{E}(\|\mathbf{a}_i - \hat{\mathbf{a}}\|^2)$ $(i = 1, 2)$ in Eq. (10), we have

$$\begin{aligned}
&\mathbb{E}(\|\mathbf{a}_i - \hat{\mathbf{a}}\|^2)\\
=&\mathbb{E}(\|(\mathbf{a}_i - \mathbf{u}) - (\hat{\mathbf{a}} - \mathbf{u})\|^2)\\
=&\|\mathbf{a}_i - \mathbf{u}\|^2 + \mathbb{E}(\|\hat{\mathbf{a}} - \mathbf{u}\|^2) - 2(\mathbf{a}_i - \mathbf{u})^T\mathbb{E}(\hat{\mathbf{a}} - \mathbf{u})\\
=&\|\mathbf{a}_i - \mathbf{u}\|^2 + \mathbb{E}(\|\hat{\mathbf{a}} - \mathbf{u}\|^2).
\end{aligned} \tag{11}$$

We can then obtain that

$$\begin{aligned}
\Delta =& \left[\|\mathbf{a}_1 - \mathbf{u}\|^2 + \mathbb{E}(\|\hat{\mathbf{a}} - \mathbf{u}\|^2)\right]\\
&- \left[\|\mathbf{a}_2 - \mathbf{u}\|^2 + \mathbb{E}(\|\hat{\mathbf{a}} - \mathbf{u}\|^2)\right]\\
=&\|\mathbf{a}_1 - \mathbf{u}\|^2 - \|\mathbf{a}_2 - \mathbf{u}\|^2 = \gamma s.
\end{aligned} \tag{12}$$

Since $\mathbf{a} \sim \mathcal{N}(\mathbf{u}, v\mathbf{I})$, we have $\frac{\mathbf{a}-\mathbf{u}}{\sqrt{v}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and further $\frac{\|\mathbf{a}-\mathbf{u}\|^2}{v} \sim \chi^2(D)$, where $\chi^2(D)$ is the chi-squared distribution with the degree of freedom D. We then have

$$\begin{aligned}
\Delta = \gamma s &= \gamma\sqrt{v^2 \mathrm{Var}(\frac{\|\mathbf{a}-\mathbf{u}\|^2}{v})}\\
&= \gamma\sqrt{v^2 \cdot 2D} = \sqrt{2D}\gamma v.
\end{aligned} \tag{13}$$

We can infer from Eq. (13) that samples close to the data mean tend to be hubs since a randomly sampled data point is expected to be closer to $\mathbf{a}_2$ than $\mathbf{a}_1$ if $\gamma > 0$ (i.e., $\|\mathbf{a}_1 - \mathbf{u}\|^2 - \|\mathbf{a}_2 - \mathbf{u}\|^2 > 0$). This analysis also suggests that the effects of hubness problem in FSL can be alleviated by reducing the chance of sampling support samples whose distances to the data mean vary greatly (i.e., reducing the chance of getting a large γ when sampling $\mathbf{a}_1$ and $\mathbf{a}_2$ in the above example). A natural idea is thus to make the origin of coordinates be the data mean and then make the norms of all feature vectors identical. Next we discuss the hubness problem with data distribution on the unit hypersphere.

Distribution on Hypersphere. Without loss of generality, we consider the unit hypersphere since the radius does not change the relative positions of data points. Let $\mathbf{b} \in \mathbb{R}^D$ denote a random vector on the D-dimensional unit hypersphere with the expectation $\mathbb{E}(\mathbf{b})$. Note that $\mathbb{E}(\mathbf{b})$ not necessarily lies on the surface of the hypersphere, i.e., its norm $l = \|\mathbb{E}(\mathbf{b})\| \in [0, 1]$. Let $s' = \sqrt{\mathrm{Var}(\cos(\mathbf{b}, \mathbb{E}(\mathbf{b})))}$ be the standard deviation of the cosine similarity between $\mathbf{b}$ and its expectation. Consider two data points $\mathbf{b}_1$ and $\mathbf{b}_2$ randomly sampled from $\mathbf{b}$, satisfying that

$$\cos(\mathbf{b}_1, \mathbb{E}(\mathbf{b})) - \cos(\mathbf{b}_2, \mathbb{E}(\mathbf{b})) = \gamma' s', \tag{14}$$

where γ' is a constant. We still use the expected difference between the squared Euclidean distances from $\mathbf{b}_1$ and $\mathbf{b}_2$ to $\hat{\mathbf{b}}$ to describe the effect of hubness problem, where $\hat{\mathbf{b}}$ is also sampled from $\mathbf{b}$:

$$\Delta' = \mathbb{E}(\|\mathbf{b}_1 - \hat{\mathbf{b}}\|^2) - \mathbb{E}(\|\mathbf{b}_2 - \hat{\mathbf{b}}\|^2). \tag{15}$$

For each term $\mathbb{E}(\|\mathbf{b}_i - \hat{\mathbf{b}}\|^2)$ $(i = 1, 2)$ in Eq. (15), we have

$$\begin{aligned}
\mathbb{E}(\|\mathbf{b}_i - \hat{\mathbf{b}}\|^2) &= \mathbb{E}(\|\mathbf{b}_i\|^2 + \|\hat{\mathbf{b}}\|^2 - 2\mathbf{b}_i^T\hat{\mathbf{b}})\\
&= 2(1 - \mathbf{b}_i^T\mathbb{E}(\mathbf{b})).
\end{aligned} \tag{16}$$

Thus we can obtain that

$$\begin{aligned}
\Delta' =&2(1 - \mathbf{b}_1^T\mathbb{E}(\mathbf{b})) - 2(1 - \mathbf{b}_2^T\mathbb{E}(\mathbf{b}))\\
=&-2l[\cos(\mathbf{b}_1, \mathbb{E}(\mathbf{b})) - \cos(\mathbf{b}_2, \mathbb{E}(\mathbf{b}))]\\
=&-2l\gamma' s'.
\end{aligned} \tag{17}$$

We can draw a similar conclusion from Eq. (17) that data points having high cosine similarities with the distribution mean tend to be hubs. Moreover, when $l = 0$ (i.e., $\mathbb{E}(\mathbf{b})$ becomes the origin of coordinates), neither $\mathbf{b}_1$ nor $\mathbf{b}_2$ has a greater chance of being a hub. This validates the aforementioned idea that making feature vectors lie on a hypersphere with zero mean help address the hubness problem. Meanwhile, we choose z-score feature normalization instead of direct l_2 normalization because the former pulls the data mean closer to the origin than the latter (see Figure 2).

4. Experiments

4.1. Datasets and Settings

Datasets. We choose three widely-used benchmarks: (1) ***mini*ImageNet** [52]: It consists of 100 classes (with 600 images per class) from ILSVRC-12 [38]. We take the split setting of [35]: 64 base classes, 16 validation classes, and 20 novel classes. (2) ***tiered*ImageNet** [37]: This is a larger subset of ILSVRC-12, which contains 608 classes and 779,165 images in total. We split it into 351 base classes, 97 validation classes, and 160 novel classes as in [37]. (3) **CUB-200-2011 Birds (CUB)** [53]: CUB is a fine-grained dataset of birds, which has 200 bird classes and 11,788 images totally. We follow [59] and split the dataset into 100 base classes, 50 validation classes, and 50 novel classes. All images of the three datasets are resized to 84 × 84.

Evaluation Protocols. We test the learned models under the 5-way 5-shot/1-shot settings. Concretely, each episode has 5 classes randomly sampled from the test split, where each class is composed of 5 shots (or 1 shot) and 15 queries. We thus have $N = 5$, $K = 5$ or 1, and $Q = 15$ for all meta-test episodes. When applying z-score normalization, although all feature vectors (i.e., those of both support and query samples) are transformed during both meta-training and meta-test, they are normalized independently. This means that the meta-test process still strictly follows the *non-transductive* setting. We report the mean 5-way few-shot classification accuracy (%, top-1) over 2,000 episodes from novel classes as well as the 95% confidence interval.

Backbones. We adopt Conv4-64 [52], Conv4-512 and ResNet-12 [14] as the feature extractors f_ϕ for fair comparison with published results. Conv4-64 and Conv4-512 both consist of four convolutional blocks, with each block containing a convolutional layer, a batch normalization layer, a ReLU activation layer, and a max pooling layer. The channel numbers of the four convolutional layers for Conv4-64 and Conv4-512 are 64-64-64-64 and 64-64-64-512, respectively. A global pooling layer is also adopted after four blocks, resulting in the output feature dimensions 64 and 512 for Conv4-64 and Conv4-512, respectively. ResNet-12 is also composed of four blocks, with three convolutional layers in each block and residual connections between blocks. The output dimension of ResNet-12 is 640.

Implementation Details. We pre-train all three backbones on the training split of each dataset to accelerate the training process as per common practice [61, 59, 44]. For Conv4-64 and Conv4-512, we employ the Adam optimizer [17] with the initial learning rate of 1e-4. For ResNet-12, the stochastic gradient descent (SGD) optimizer is employed with the initial learning rate of 1e-4, the weight decay of 5e-4, and the Nesterov momentum of 0.9. The learning rate is halved every 20 epochs in all experiments. The scaling hyper-parameter T in Eq. (3) is selected from $\{16, 32, 64, 128, 256\}$ according to the validation performances. We also adopt the element-wise affine transformation when applying ZN during episodic training in the experiments, i.e., $\mathbf{x}^{(zn)} \leftarrow \mathbf{x}^{(zn)} \odot \omega + \beta$, where $\odot$ denotes the element-wise multiplication, and $\omega \in \mathbb{R}^D$ and $\beta \in \mathbb{R}^D$ are the learnable weight and bias parameters, respectively.

Method	Backbone	5-way 1-shot	5-way 5-shot
MatchingNet [52]	Conv4-64	43.56 ± 0.84	55.31 ± 0.73
MAML [8]	Conv4-64	48.70 ± 1.84	63.10 ± 0.92
RelationNet [49]	Conv4-64	50.40 ± 0.80	65.30 ± 0.70
Baseline++ [5]	Conv4-64	48.24 ± 0.75	66.43 ± 0.63
DN4 [23]	Conv4-64	51.24 ± 0.74	71.02 ± 0.64
PARN [56]	Conv4-64	55.22 ± 0.84	71.55 ± 0.66
Centroid [1]	Conv4-64	53.14 ± 1.06	71.45 ± 0.72
Neg-Cosine [25]	Conv4-64	52.84 ± 0.76	70.41 ± 0.66
FEAT [59]	Conv4-64	55.15 ± 0.20	71.61 ± 0.16
ProtoNet [47]	Conv4-64	53.01 ± 0.45	71.10 ± 0.36
IMP [2]	Conv4-64	51.80 ± 0.44	70.09 ± 0.36
IEPT† [63]	Conv4-64	54.87 ± 0.44	73.76 ± 0.34
ProtoNet+ZN	Conv4-64	55.16 ± 0.44	71.78 ± 0.36
IMP+ZN	Conv4-64	54.74 ± 0.44	70.66 ± 0.36
IEPT†+ZN	Conv4-64	$\mathbf{57.83 \pm 0.45}$	$\mathbf{74.88 \pm 0.34}$
MAML [8]	Conv4-512	49.33 ± 0.60	65.17 ± 0.49
Relation Net [49]	Conv4-512	50.86 ± 0.57	67.32 ± 0.44
PN+rot [9]	Conv4-512	56.02 ± 0.46	74.00 ± 0.35
CC+rot [9]	Conv4-512	56.27 ± 0.43	74.30 ± 0.33
ProtoNet [47]	Conv4-512	54.73 ± 0.45	73.06 ± 0.36
IMP [2]	Conv4-512	52.58 ± 0.45	72.29 ± 0.36
IEPT† [63]	Conv4-512	55.40 ± 0.45	74.29 ± 0.35
ProtoNet+ZN	Conv4-512	56.63 ± 0.45	73.90 ± 0.35
IMP+ZN	Conv4-512	54.76 ± 0.45	72.47 ± 0.36
IEPT†+ZN	Conv4-512	$\mathbf{57.76 \pm 0.45}$	$\mathbf{75.11 \pm 0.35}$
TADAM [30]	ResNet-12	58.50 ± 0.30	76.70 ± 0.38
MetaOptNet [19]	ResNet-12	62.64 ± 0.61	78.63 ± 0.46
MTL [48]	ResNet-12	61.20 ± 1.80	75.50 ± 0.80
AM3 [57]	ResNet-12	65.21 ± 0.49	75.20 ± 0.36
Shot-Free [36]	ResNet-12	59.04 ± 0.43	77.64 ± 0.39
Neg-Cosine [25]	ResNet-12	63.85 ± 0.81	81.57 ± 0.56
Distill [51]	ResNet-12	64.82 ± 0.60	82.14 ± 0.43
DSN-MR [44]	ResNet-12	64.60 ± 0.72	79.51 ± 0.50
DeepEMD [61]	ResNet-12	65.91 ± 0.82	82.41 ± 0.56
FEAT [59]	ResNet-12	66.78 ± 0.20	82.05 ± 0.14
ProtoNet [47]	ResNet-12	63.38 ± 0.45	81.22 ± 0.30
IMP [2]	ResNet-12	63.70 ± 0.47	80.55 ± 0.30
IEPT† [63]	ResNet-12	64.05 ± 0.44	82.73 ± 0.29
ProtoNet+ZN	ResNet-12	66.06 ± 0.44	81.73 ± 0.30
IMP+ZN	ResNet-12	65.01 ± 0.43	81.72 ± 0.30
IEPT†+ZN	ResNet-12	$\mathbf{67.35 \pm 0.43}$	$\mathbf{83.04 \pm 0.29}$

Table 1. Comparative results of standard FSL on *mini*ImageNet. The average 5-way few-shot classification accuracies (%, top-1) along with the 95% confidence intervals are reported.

4.2. Main Results

Note that we can employ any metric-based FSL method as the baseline. Without loss of generality, we apply the z-score feature normalization to three classic/state-of-the-art FSL approaches: ProtoNet [47], IMP [2], and IEPT [63]. Particularly, for simple implementation, we only use the main module (i.e., the concatenation of four feature vectors that come from the original image and three augmented ones, denoted with †) of IEPT instead of the whole model. After adopting ZN, each FSL model is thus named with the suffix '+ZN'. For fair comparison, we re-implement ProtoNet, IMP, and IEPT† by also adopting the temperature scaling technique with different backbones.

The comparative results on the three datasets are shown in Table 1, Table 2, and Table 3, respectively. Models using the same backbones are placed together. We can make the following observations: (1) The z-score feature normalization boosts a variety of metric-based FSL methods. Specifically, the improvements achieved by methods trained with ZN over their original versions without ZN range from 0.2% – 5.3%. This clearly validates the general applicability of ZN for metric-based FSL. (2) The improvements obtained by employing ZN under the 1-shot setting (1.3% – 5.3%) are significantly larger than those under the 5-shot setting (0.2% – 2.5%). One plausible explanation is that: classification tasks with less support samples are more likely to suffer from the hubness problem, and ZN is designed to alleviate such negative effects and thus results in better performance when the problem is more acute. (3) Methods boosted by ZN achieve the best results on all three datasets under all settings. Particularly, a method as simple as ProtoNet+ZN is already comparable to the state-of-the-art, further demonstrating the effectiveness of ZN. (4) Baseline++ [5], Neg-Cosine [25], and Distill [51] claim that only pre-training on the base classes (i.e., meta-training is not needed) is re-

Method	Backbone	5-way 1-shot	5-way 5-shot
MAML [8]	Conv4-64	51.67 ± 1.81	70.30 ± 0.08
RelationNet [49]	Conv4-64	54.48 ± 0.93	71.32 ± 0.78
ProtoNet [47]	Conv4-64	53.56 ± 0.48	72.52 ± 0.41
IMP [2]	Conv4-64	52.33 ± 0.48	72.62 ± 0.41
IEPT† [63]	Conv4-64	54.76 ± 0.48	74.12 ± 0.40
ProtoNet+ZN	Conv4-64	56.70 ± 0.49	73.34 ± 0.40
IMP+ZN	Conv4-64	56.65 ± 0.49	73.61 ± 0.41
IEPT†+ZN	Conv4-64	$\mathbf{57.76 \pm 0.49}$	$\mathbf{74.90 \pm 0.40}$
MAML [8]	Conv4-512	52.84 ± 0.56	70.91 ± 0.46
Relation Net [49]	Conv4-512	54.69 ± 0.59	72.71 ± 0.43
ProtoNet [47]	Conv4-512	55.12 ± 0.48	75.27 ± 0.39
IMP [2]	Conv4-512	55.62 ± 0.49	73.94 ± 0.40
IEPT† [63]	Conv4-512	55.29 ± 0.47	75.76 ± 0.38
ProtoNet+ZN	Conv4-512	58.93 ± 0.48	76.27 ± 0.38
IMP+ZN	Conv4-512	57.30 ± 0.49	74.81 ± 0.39
IEPT†+ZN	Conv4-512	$\mathbf{59.28 \pm 0.48}$	$\mathbf{77.34 \pm 0.38}$
MetaOptNet [19]	ResNet-12	65.99 ± 0.72	81.56 ± 0.63
MTL [48]	ResNet-12	65.62 ± 1.80	80.61 ± 0.90
AM3 [57]	ResNet-12	67.23 ± 0.34	78.95 ± 0.22
Shot-Free [36]	ResNet-12	66.87 ± 0.43	82.64 ± 0.43
Distill [51]	ResNet-12	71.52 ± 0.69	86.03 ± 0.49
DSN-MR [44]	ResNet-12	67.39 ± 0.82	82.85 ± 0.56
DeepEMD [61]	ResNet-12	71.16 ± 0.87	86.03 ± 0.58
FEAT [59]	ResNet-12	70.80 ± 0.23	84.79 ± 0.16
ProtoNet [47]	ResNet-12	69.36 ± 0.52	85.80 ± 0.35
IMP [2]	ResNet-12	65.65 ± 0.51	83.28 ± 0.35
IEPT† [63]	ResNet-12	69.66 ± 0.52	86.41 ± 0.34
ProtoNet+ZN	ResNet-12	71.98 ± 0.51	86.42 ± 0.34
IMP+ZN	ResNet-12	67.58 ± 0.51	83.94 ± 0.35
IEPT†+ZN	ResNet-12	$\mathbf{72.28 \pm 0.51}$	$\mathbf{87.20 \pm 0.34}$

Table 2. Comparative results of standard FSL on *tiered*ImageNet. The average 5-way few-shot classification accuracies (%, top-1) along with the 95% confidence intervals are reported.

Method	Backbone	5-way 1-shot	5-way 5-shot
MatchingNet [52]	Conv4-64	61.16 ± 0.89	72.86 ± 0.70
MAML [8]	Conv4-64	55.92 ± 0.95	72.09 ± 0.76
Relation Net [49]	Conv4-64	62.45 ± 0.98	76.11 ± 0.69
FEAT [59]	Conv4-64	68.87 ± 0.22	82.90 ± 0.15
ProtoNet [47]	Conv4-64	68.00 ± 0.51	84.41 ± 0.32
IMP [2]	Conv4-64	67.90 ± 0.49	84.81 ± 0.31
IEPT† [63]	Conv4-64	68.27 ± 0.51	85.30 ± 0.32
ProtoNet+ZN	Conv4-64	71.30 ± 0.49	85.35 ± 0.32
IMP+ZN	Conv4-64	71.22 ± 0.48	85.51 ± 0.31
IEPT†+ZN	Conv4-64	$\mathbf{73.54 \pm 0.48}$	$\mathbf{87.82 \pm 0.30}$

Table 3. Comparative results of fine-grained FSL on CUB. The average 5-way few-shot classification accuracies (%, top-1) along with the 95% confidence intervals are reported.

Method	1-shot Acc.	5-shot Acc.	$\mathbf{S_{N_1^{(h)}}}$	$\mathbf{S_{N_5^{(h)}}}$
***mini*ImageNet:**				
None	49.89 ± 0.44	69.54 ± 0.37	3.219	3.259
l_2	49.91 ± 0.44	69.73 ± 0.36	3.087	3.141
CS_l_2	52.07 ± 0.44	69.74 ± 0.36	/	/
SimpleShot [54]	$\mathbf{53.21 \pm 0.44}$	69.84 ± 0.36	**2.322**	**2.337**
ZN	52.80 ± 0.44	$\mathbf{70.19 \pm 0.36}$	2.360	2.411
***tiered*ImageNet:**				
None	51.12 ± 0.47	69.26 ± 0.42	2.702	2.812
l_2	51.92 ± 0.47	70.44 ± 0.42	2.415	2.588
CS_l_2	52.21 ± 0.46	69.53 ± 0.42	/	/
SimpleShot [54]	52.89 ± 0.47	69.61 ± 0.42	2.241	2.250
ZN	$\mathbf{53.76 \pm 0.47}$	$\mathbf{70.82 \pm 0.42}$	**2.072**	**2.093**

Table 4. Comparative results of alternative normalization operations over the *pre-trained* models on two benchmarks. The average 5-way few-shot classification accuracies (%, top-1) along with the 95% confidence intervals are reported. To show the effect of hubness, the skewnesses $S_{N_1^{(h)}}$ and $S_{N_5^{(h)}}$ are reported.

quired. Our results demonstrate that with ZN, even ProtoNet+ZN outperforms all of them (except Distill under the 5-shot on *mini*ImageNet). This suggests that *meta-training is still of great usefulness for FSL.*

4.3. Comparison to Alternative Normalizations

We compare different normalization strategies on two datasets in Table 4 and Table 5 based on pre-trained models and episodic training, respectively. To show the effect of hubness, we additionally report the skewness $S_{N_k^{(h)}}$ of the distribution $N_k^{(h)}$ ($k = 1, 5$) on the whole test set of each dataset, defined as:

$$S_{N_k^{(h)}} = \frac{\mathbb{E}(N_k^{(h)} - \mu_{N_k^{(h)}})^3}{\sigma^3_{N_k^{(h)}}}, \tag{18}$$

where $N_k^{(h)}$ is the distribution of k-occurrence, $\mu_{N_k^{(h)}}$ is the mean of $N_k^{(h)}$, $\sigma_{N_k^{(h)}}$ is the standard deviation of $N_k^{(h)}$, and the skewness $S_{N_k^{(h)}}$ is the third moment of $N_k^{(h)}$.

For experiments in Table 4, we use the same pre-trained model for all methods on each dataset, which is trained on the training set as a conventional classification task (e.g., 64-class classification on the training set of *mini*ImageNet) and is validated on the validation set as a multi-way few-shot classification (e.g., 16-way few-shot classification for *mini*ImageNet) based on the nearest neighbor. We compare ZN with directly using the pre-trained model (denoted as 'None') and three alternative ways of normalization: (1) l_2: l_2 normalization. (2) CS_l_2 (centering with support mean before l_2): for each feature vector in an episode, it is centered by subtracting the mean of all support features before applying l_2 normalization. Since it can be only employed within each episode, we cannot compute $S_{N_k^{(h)}}$ based on the whole test set. (3) SimpleShot [54]: each test feature vector is normalized by subtracting the mean of the whole training set before l_2 normalization. We can observe from Table 4 that: (1) Compared to 'None', all normalizations help improve the FSL results and alleviate the effect of the hubness problem since $S_{N_1^{(h)}}$ and $S_{N_5^{(h)}}$ become smaller. This indicates that mitigating the hubness problem is helpful for FSL. (2) ZN achieves the best results on *tiered*ImageNet in terms of all evaluation metrics and is very close to SimpleShot on *mini*ImageNet. SimpleShot performs better on *mini*ImageNet because it centers the test features with training set mean and the train-test gap

Method	1-shot Acc.	5-shot Acc.	$S_{N_1^{(h)}}$	$S_{N_5^{(h)}}$
***mini*ImageNet:**				
ProtoNet+Ring [28]	53.22 ± 0.44	71.53 ± 0.35	2.856	2.916
SEN [28]	53.38 ± 0.45	71.06 ± 0.35	2.698	2.809
ProtoNet [47]	53.01 ± 0.45	71.10 ± 0.36	2.914	2.884
ProtoNet+l_2	53.99 ± 0.45	71.50 ± 0.35	2.743	2.877
ProtoNet+CS_l_2	53.93 ± 0.43	71.32 ± 0.35	/	/
ProtoNet+ZN	$\mathbf{55.16 \pm 0.44}$	$\mathbf{71.78 \pm 0.36}$	**2.572**	**2.565**
***tiered*ImageNet:**				
ProtoNet+Ring [28]	53.47 ± 0.48	72.66 ± 0.41	3.003	3.134
SEN [28]	53.10 ± 0.48	71.86 ± 0.41	3.035	3.220
ProtoNet [47]	53.56 ± 0.48	72.52 ± 0.41	2.937	3.045
ProtoNet+l_2	55.13 ± 0.49	72.98 ± 0.40	2.787	2.931
ProtoNet+CS_l_2	55.39 ± 0.47	72.92 ± 0.40	/	/
ProtoNet+ZN	$\mathbf{56.70 \pm 0.49}$	$\mathbf{73.34 \pm 0.40}$	**2.203**	**2.312**

Table 5. Comparative results of alternative normalization operations with *episodic training* on two benchmarks. The average 5-way few-shot classification accuracies (%, top-1) along with the 95% confidence intervals are reported. To show the effect of hubness, the skewnesses $S_{N_1^{(h)}}$ and $S_{N_5^{(h)}}$ are reported.

is smaller for *mini*ImageNet than that of *tiered*ImageNet, while ZN is insensitive to the train-test gap. This also validates our idea in Section 3.3 that when the dataset mean can be estimated accurately, centering before normalization can be helpful. (3) CS_l_2 also performs centering but the results are not as good as ZN. This is because the support mean is not reliable enough especially with less shots.

For experiments in Table 5, we conduct episodic training with different normalizations. SimpleShot is not adopted here since it cannot be integrated into the meta-training process. We compare two additional methods: (1) ProtoNet+Ring [28] adopts an extra Ring loss [64] to explicitly constrain the norms of feature vectors besides the standard ProtoNet FSL loss; (2) SEN [28] modifies the Euclidean distance metric to learn features with similar norms. It can be observed from Table 5 that ZN is also able to help the episodic training process and ProtoNet+ZN consistently beats all compared methods.

4.4. Visualization Results

To observe the effects of hubness on data with multiple classes/clusters, we visualize the kNN relations with $k = 5$ among samples in Figure 3. Concretely, we first randomly sample two subsets from the test split of *mini*ImageNet (corresponding to two rows in Figure 3), with each subset containing 5 classes and 250 samples in total. For each subset, we then visualize two kNN matrices with ProtoNet (left) and ProtoNet+ZN (right), respectively. In each matrix, both two axes represent the ID of the samples. If the i-th sample is among the 5 nearest neighbors of the j-th sample, a point is then plotted at coordinates (i, j) (i.e., there are exactly 5 points in each row). The color of points indicates the $N_5^{(h)}$ value of the i-th sample, and deeper color represents higher value. Since we sort the samples by class, the ideal case would be that the k nearest neighbors of every sample are only those from the same class, i.e., only the diagonal blocks would be colored. We can see from the left column that there exist deep colored vertical lines, which correspond to hubs that appear among the 5 nearest neighbors of many other samples, even of those not belonging to the same class as the hubs. The presence of such hubs clearly harms the classification performance. On the contrary, kNN matrices of ProtoNet+ZN (in the right column) are cleaner in the off-diagonal areas, and the long vertical lines disappear. This demonstrates the ability of ZN to mitigate the hubness problem for FSL.

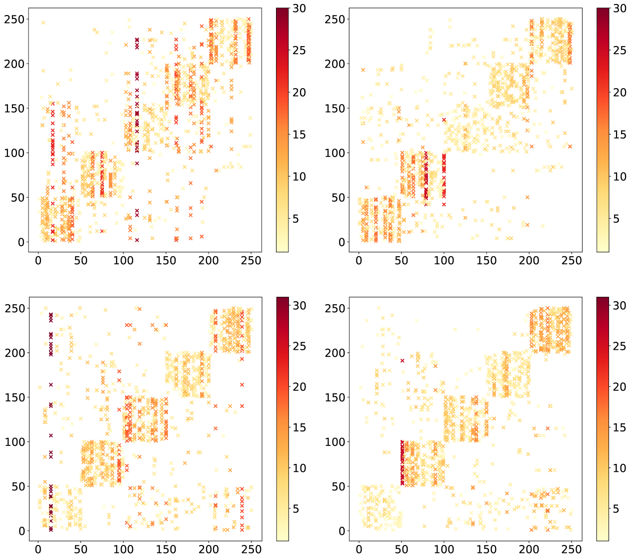


Figure 3. Visualizations of kNN matrices on two subsets (corresponding to two rows) of the *mini*ImageNet test set. Each subset contains 5 randomly sampled classes with 50 samples in each class. Subfigures in the left column are results of ProtoNet, while right ones are results of ProtoNet+ZN. Conv4-64 is adopted.

5. Conclusion

In this paper, we have discovered the existence of the hubness problem in FSL, identified the cause and also proposed a solution on how to alleviate the effect of hubness based on a theoretical analysis. Specifically, we propose to apply z-score feature normalization, which is simple yet effective in mitigating the hubness problem. We validate its effectiveness with extensive experiments and visualizations. It is also shown to be generally applicable, boosting the performance of a variety of metric-based FSL methods.

Acknowledgments This work was supported by National Natural Science Foundation of China (61976220 and 61832017), Beijing Outstanding Young Scientist Program (BJJWZYJH012019100020098), and Open Project Program Foundation of Key Laboratory of Opto-Electronics Information Processing, CAS (OEIP-O-202006).

References

[1] Gagné Christian Afrasiyabi Arman, Lalonde Jean-François. Associative alignment for few-shot image classification. *ECCV*, 2020. 6

[2] Kelsey R. Allen, Evan Shelhamer, Hanul Shin, and Joshua B. Tenenbaum. Infinite mixture prototypes for few-shot learning. In *ICML*, pages 232–241, 2019. 1, 2, 3, 6, 7

[3] Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2019. 2

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2018. 1

[5] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019. 6

[6] Georgiana Dinu and Marco Baroni. Improving zero-shot learning by mitigating the hubness problem. In *ICLR Workshop*, 2015. 3

[7] Nanyi Fei, Zhiwu Lu, Tao Xiang, and Songfang Huang. MELR: Meta-learning via modeling episode-level relationships for few-shot learning. In *ICLR*, 2021. 1, 2

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017. 1, 2, 6, 7

[9] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Perez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *ICCV*, pages 8059–8068, 2019. 1, 6

[10] Spyros Gidaris and Nikos Komodakis. Generating classification weights with GNN denoising autoencoders for few-shot learning. In *CVPR*, pages 21–30, 2019. 3

[11] Yiluan Guo and Ngai-Man Cheung. Attentive weights generation for few shot learning via information maximization. In *CVPR*, pages 13496–13505, 2020. 3

[12] Bharath Hariharan and Ross B. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, pages 3037–3046, 2017. 3

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 2, 3

[16] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D. Yoo. Edge-labeling graph neural network for few-shot learning. In *CVPR*, pages 11–20, 2019. 2

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[18] Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *ICLR*, 2018. 3

[19] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, pages 10657–10665, 2019. 1, 6, 7

[20] Fei-Fei Li, Robert Fergus, and Pietro Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, pages 1134–1141, 2003. 1

[21] Fei-Fei Li, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(4):594–611, 2006. 1

[22] Kai Li, Yulun Zhang, Kunpeng Li, and Yun Fu. Adversarial feature hallucination networks for few-shot learning. In *CVPR*, pages 13467–13476, 2020. 3

[23] 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 *CVPR*, pages 7260–7268, 2019. 6

[24] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few shot learning. *CoRR*, abs/1707.09835, 2017. 3

[25] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. *ECCV*, 2020. 3, 6

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1

[27] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *ICML*, pages 2554–2563, 2017. 3

[28] Van Nhan Nguyen, Sigurd Løkse, Kristoffer Wickstróm, Michael Kampffmeyer, Davide Roverso, and Robert Jenssen. Sen: A novel feature normalization dissimilarity measure for prototypical few-shot learning networks. *ECCV*, 2020. 2, 3, 8

[29] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. 2

[30] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pages 721–731, 2018. 2, 6

[31] Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *ICCV*, pages 3603–3612, 2019. 2

[32] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L. Yuille. Few-shot image recognition by predicting parameters from activations. In *CVPR*, pages 7229–7238, 2018. 3

[33] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 1

[34] Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. Hubs in space: Popular nearest neighbors in high-dimensional data. *JMLR*, 11:2487–2531, 2010. 1, 3, 4

[35] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 2, 3, 5

[36] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *ICCV*, pages 331–339, 2019. 6, 7

[37] 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 *ICLR*, 2018. 5

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 1, 5

[39] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019. 2

[40] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *ICLR*, 2018. 2

[41] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogério Schmidt Feris, Raja Giryes, and Alexander M. Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *Advances in Neural Information Processing Systems*, pages 2850–2860, 2018. 3

[42] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *ICCV*, pages 4569–4579, 2019. 1

[43] Yutaro Shigeto, Ikumi Suzuki, Kazuo Hara, Masashi Shimbo, and Yuji Matsumoto. Ridge regression, hubness, and zero-shot learning. In *ECML-PKDD*, pages 135–151, 2015. 1, 3, 4

[44] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *CVPR*, pages 4136–4145, 2020. 2, 6, 7

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[46] Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *ICLR*, 2017. 3

[47] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4080–4090, 2017. 1, 2, 3, 6, 7, 8

[48] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, pages 403–412, 2019. 6, 7

[49] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, pages 1199–1208, 2018. 1, 2, 6, 7

[50] Ikumi Suzuki, Kazuo Hara, Masashi Shimbo, Marco Saerens, and Kenji Fukumizu. Centering similarity measures to reduce hubs. In *EMNLP*, pages 613–623, 2013. 1, 3

[51] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *ECCV*, 2020. 3, 6, 7

[52] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016. 1, 2, 5, 6, 7

[53] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5

[54] Yan Wang, Wei-Lun Chao, Kilian Q. Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *CoRR*, abs/1911.04623, 2019. 3, 7

[55] Yu-Xiong Wang, Ross B. Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *CVPR*, pages 7278–7286, 2018. 3

[56] Ziyang Wu, Yuwei Li, Lihua Guo, and Kui Jia. Parn: Position-aware relation networks for few-shot learning. In *ICCV*, pages 6659–6667, 2019. 2, 6

[57] Chen Xing, Negar Rostamzadeh, Boris Oreshkin, and Pedro O O. Pinheiro. Adaptive cross-modal few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4847–4857, 2019. 6, 7

[58] Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu. DPGN: distribution propagation graph network for few-shot learning. In *CVPR*, pages 13387–13396, 2020. 2

[59] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *CVPR*, 2020. 1, 2, 3, 5, 6, 7

[60] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *ICML*, pages 7115–7123, 2019. 2

[61] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, pages 12203–12213, 2020. 6, 7

[62] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, pages 3010–3019, 2017. 3

[63] Manli Zhang, Jianhong Zhang, Zhiwu Lu, Tao Xiang, Mingyu Ding, and Songfang Huang. IEPT: Instance-level and episode-level pretext tasks for few-shot learning. In *ICLR*, 2021. 1, 2, 3, 6, 7

[64] Yutong Zheng, Dipan K. Pal, and Marios Savvides. Ring loss: Convex feature normalization for face recognition. In *CVPR*, pages 5089–5097, 2018. 8