Hubs and Hyperspheres: Reducing Hubness and Improving Transductive Few-shot Learning with Hyperspherical Embeddings

Daniel J. Trosten*†, Rwiddhi Chakraborty*†, Sigurd Løkse†, Kristoffer Knutsen Wickstrøm†,
Robert Jenssen†‡§¶, Michael C. Kampffmeyer†‡
Department of Physics and Technology, UiT The Arctic University of Norway
firstname[.middle initial].lastname@uit.no

Abstract

Distance-based classification is frequently used in transductive few-shot learning (FSL). However, due to the high-dimensionality of image representations, FSL classifiers are prone to suffer from the hubness problem, where a few points (hubs) occur frequently in multiple nearest neighbour lists of other points. Hubness negatively impacts distance-based classification when hubs from one class appear often among the nearest neighbors of points from another class, degrading the classifier's performance. To address the hubness problem in FSL, we first prove that hubness can be eliminated by distributing representations uniformly on the hypersphere. We then propose two new approaches to embed representations on the hypersphere, which we prove optimize a tradeoff between uniformity and local similarity preservation – reducing hubness while retaining class structure. Our experiments show that the proposed methods reduce hubness, and significantly improves transductive FSL accuracy for a wide range of classifiers[1].

1. Introduction

While supervised deep learning has made a significant impact in areas where large amounts of labeled data are available [6, 11], few-shot learning (FSL) has emerged as a promising alternative when labeled data is limited [3, 12, 14, 16, 21, 26, 28, 31, 33, 39, 40]. FSL aims to design classifiers that can discriminate between novel classes based on a few labeled instances, significantly reducing the cost of the labeling procedure.

In transductive FSL, one assumes access to the entire query set during evaluation. This allows transductive FSL classifiers to learn representations from a larger number of samples, resulting in better performing classifiers. However, many of these methods base their predictions on distances to prototypes for the novel classes [3, 16, 21, 28, 39, 40]. This makes these methods susceptible to the hubness problem [10, 22, 24, 25], where certain exemplar points (hubs) appear among the nearest neighbours of many other points. If a support sample is a hub, many query samples will be assigned to it regardless of their true label, resulting in low accuracy. If more training data is available, this effect can be reduced by increasing the number of labeled samples in the classification rule – but this is impossible in FSL.

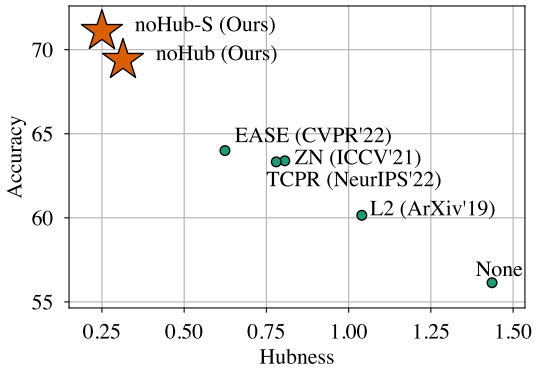

Figure 1. Few-shot accuracy increases when hubness decreases. The figure shows the 1-shot accuracy when classifying different embeddings with SimpleShot [33] on mini-ImageNet [29].

Several approaches have recently been proposed to embed samples in a space where the FSL classifier's performance is improved [4, 5, 7, 17, 33, 35, 39]. However, only one of these directly addresses the hubness problem. Fei *et al*. [7] show that embedding representations on a hypersphere with zero mean reduces hubness. They advocate the use of Z-score normalization (ZN) along the feature axis of each representation, and show empirically that ZN can reduce hubness in FSL. However, ZN does not guarantee a data mean of zero, meaning that hubness can still occur after ZN.

*Equal contributions.
†UiT Machine Learning group (machine-learning.uit.no) and Visual Intelligence Centre (visual-intelligence.no).
‡Norwegian Computing Center.
§Department of Computer Science, University of Copenhagen.
¶Pioneer Centre for AI (aicentre.dk).
[1]Code available at https://github.com/uitml/noHub.

In this paper we propose a principled approach to embed representations in FSL, which both reduces hubness and improves classification performance. First, we prove that hubness can be eliminated by embedding representations uniformly on the hypersphere. However, distributing representations uniformly on the hypersphere without any additional constraints will likely break the class structure which is present in the representation space – hurting the performance of the downstream classifier. Thus, in order to both reduce hubness and preserve the class structure in the representation space, we propose two new embedding methods for FSL. Our methods, Uniform Hyperspherical Structure-preserving Embeddings (noHub) and noHub with Support labels (noHub-S), leverage a decomposition of the Kullback-Leibler divergence between representation and embedding similarities, to optimize a tradeoff between Local Similarity Preservation (LSP) and uniformity on the hypersphere. The latter method, noHub-S, also leverages label information from the support samples to further increase the class separability in the embedding space.

Figure 1 illustrates the correspondence between hubness and accuracy in FSL. Our methods have both the *least hubness* and *highest accuracy* among several recent embedding techniques for FSL.

Our contributions are summarized as follows.

- We prove that the uniform distribution on the hypersphere has zero hubness and that embedding points uniformly on the hypersphere thus alleviates the hubness problem in distance-based classification for transductive FSL.
- We propose noHub and noHub-S to embed representations on the hypersphere, and prove that these methods optimize a tradeoff between LSP and uniformity. The resulting embeddings are therefore approximately uniform, while simultaneously preserving the class structure in the embedding space.
- Extensive experimental results demonstrate that noHub and noHub-S outperform current state-of-the-art embedding approaches, boosting the performance of a wide range of transductive FSL classifiers, for multiple datasets and feature extractors.

2. Related Work

The hubness problem. The hubness problem refers to the emergence of *hubs* in collections of points in high-dimensional vector spaces [22]. Hubs are points that appear among the nearest neighbors of many other points, and are therefore likely to have a significant influence on *e.g.* nearest neighbor-based classification. Radovanovic *et al.* [22] showed that points closer to the expected data mean are more likely be among the nearest neighbors of other points, indicating that these points are more likely to be hubs. Hubness can also be seen as a result of large density gradients [9], as points in high-density areas are more likely to be hubs. The hubness problem is thus an intrinsic property of data distributions in high-dimensional vector spaces, and not an artifact occurring in particular datasets. It is therefore important to take the hubness into account when designing classification systems in high-dimensional vector spaces.

Hubness in FSL. Many recent methods in FSL rely on distance-based classification in high-dimensional representation spaces [1, 3, 19, 33, 36, 38, 40], making them vulnerable to the hubness problem. Fei *et al.* [7] show that hyperspherical representations with zero mean reduce hubness. Motivated by this insight, they suggest that representations should have zero mean and unit standard deviation (ZN) *along the feature dimension*. This effectively projects samples onto the hyperplane orthogonal to the vector with all elements $= 1$, and pushes them to the hypersphere with radius $\sqrt{d}$, where d is the dimensionality of the representation space. Although ZN is empirically shown to reduce hubness, it does not guarantee that the data mean is zero. The normalized representations can therefore still suffer from hubness, potentially decreasing FSL performance.

Embeddings in FSL. FSL classifiers often operate on embeddings of representations instead of the representations themselves, to improve the classifier's ability to generalize to novel classes [5, 33, 35, 39]. Earlier works use the L2 normalization and Centered L2 normalization to embed representations on the hypersphere [33]. Among more recent embedding techniques, ReRep [5] performs a two-step fusing operation on both the support and query features with an attention mechanism. EASE [39] combines both support and query samples into a single sample set, and jointly learns a similarity and dissimilarity matrix, encouraging similar features to be embedded closer, and dissimilar features to be embedded far away. TCPR [35] computes the top-k neighbours of each test sample from the base data, computes the centroid, and removes the feature components in the direction of the centroid. Although these methods generally lead to a reduction in hubness and an increase in performance (see Figure 1), they are not explicitly designed to address the hubness problem resulting in suboptimal hubness reduction and performance. In contrast, our proposed noHub and noHub-S directly leverage our theoretic insights to target the root of the hubness problem.

Hyperspherical uniformity. Benefits of uniform hyperspherical representations have previously been studied for contrastive self-supervised learning (SSL) [32]. Our work differs from [32] on several key points. First, we study a non-parametric embedding of support and query samples for FSL, which is a fundamentally different task from contrastive SSL. Second, the contrastive loss studied in [32] is a

combination of different cross-entropies, making it different from our KL-loss. Finally, we introduce a tradeoff-parameter between uniformity and LSP, and connect our theoretical results to hubness and Laplacian Eigenmaps.

3. Hyperspherical Uniform Eliminates Hubness

We will now show that hubness can be eliminated completely by embedding representations *uniformly* on the hypersphere[2].

Definition 1 (Uniform PDF on the hypersphere.)**.** *The uniform probability density function (PDF) on the unit hypersphere* $\mathbb{S}_d = \{\boldsymbol{x} \in \mathbb{R}^d \mid ||\boldsymbol{x}|| = 1\} \subset \mathbb{R}^d$ *is*

$$u_{\mathbb{S}_d}(\boldsymbol{x}) = A_d^{-1}\delta(||\boldsymbol{x}|| - 1) \tag{1}$$

where $A_d = \frac{2\pi^{d/2}}{\Gamma(d/2)}$ *is the surface area of* $\mathbb{S}_d$*, and* $\delta(\cdot)$ *is the Dirac delta distribution.*

We then have the following propositions[3] for random vectors with this PDF.

Proposition 1. *Suppose* $\boldsymbol{X}$ *has PDF* $u_{\mathbb{S}_d}(\boldsymbol{x})$*. Then*

$$\mathbb{E}(\boldsymbol{X}) = 0 \tag{2}$$

Proposition 2. *Let* $\Pi_{\boldsymbol{p}}$ *be the tangent plane of* $\mathbb{S}_d$ *at an arbitrary point* $\boldsymbol{p} \in \mathbb{S}_d$*. Then, for any direction* $\boldsymbol{\theta}^*$ *in* $\Pi_{\boldsymbol{p}}$ *the directional derivative of* $u_{\mathbb{S}_d}$ *along* $\boldsymbol{\theta}^*$ *is*

$$\nabla_{\boldsymbol{\theta}^*} u_{\mathbb{S}_d} = 0 \tag{3}$$

These two propositions show that the hyperspherical uniform has (i) zero mean; and (ii) zero density gradient along all directions tangent to the hypersphere's surface, at all points on the hypersphere. The hyperspherical uniform thus provably eliminates hubness, both in the sense of having a zero data mean, and having zero density gradient everywhere. We note that the latter property is un-attainable in Euclidean space, as it is impossible to define a uniform distribution over the whole space. It is therefore necessary to embed points on a non-Euclidean sub-manifold in order to eliminate hubness.

4. Method

In the preceding section, we proved that uniform embeddings on the hypersphere eliminate hubness. However, naïvely placing points uniformly on the hypersphere does not incorporate the inherent class structure in the data, leading to poor FSL performance. Thus, there exists a tradeoff between uniformity on the hypersphere and the preservation of local similarities. To address this tradeoff, we introduce two novel embedding approaches for FSL, namely noHub and noHub-S. noHub (Sec. 4.1) incorporates a novel loss function for embeddings on the hypersphere, while noHub-S (Sec. 4.2), guides noHub with additional label information, which should act as a supervisory signal for a class-aware embedding that leads to improved classification performance. Figure 2 provides an overview of the proposed noHub method. We also note that, since our approach generates embeddings, they are compatible with most transductive FSL classifier.

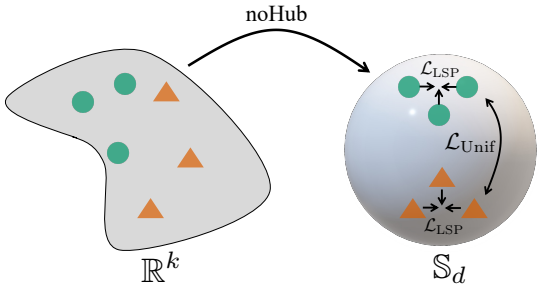

Figure 2. Illustration of the noHub embedding. Given representations $\in \mathbb{R}^k$, $\mathcal{L}_{\text{LSP}}$ preserves local similarities. $\mathcal{L}_{\text{Unif}}$ simultaneously encourages uniformity in the embedding space $\mathbb{S}_d$. This feature embedding framework helps reduce hubness while improving classification performance.

Few-shot Preliminaries. Assume we have a large labeled *base* dataset $\mathcal{X}_{\text{Base}} = \{(\boldsymbol{x}_i, y_i) \mid y_i \in \mathcal{C}_{\text{Base}};\ i = 1, \ldots, n_{\text{Base}}\}$, where $\boldsymbol{x}_i$ and y_i denotes the raw features and labels, respectively. Let $\mathcal{C}_{\text{Base}}$ denote the set of classes for the base dataset. In the few–shot scenario, we assume that we are given another labeled dataset $\mathcal{X}_{\text{Novel}} = \{(\boldsymbol{x}_i, y_i) \mid y_i \in \mathcal{C}_{\text{Novel}};\ i = 1, \ldots, n_{\text{Novel}}\}$ from *novel*, previously unseen classes $\mathcal{C}_{\text{Novel}}$, satisfying $\mathcal{C}_{\text{Base}} \cap \mathcal{C}_{\text{Novel}} = \emptyset$. In addition, we have a test set $\mathcal{T}$, $\mathcal{T} \cap \mathcal{X}_{\text{Novel}} = \emptyset$, also from $\mathcal{C}_{\text{Novel}}$.

In a K–way N_S–shot FSL problem, we create randomly sampled *tasks* (or episodes), with data from K randomly chosen novel classes. Each task consists of a *support* set $\mathcal{S} \subset \mathcal{X}_{\text{Novel}}$ and a *query* set $\mathcal{Q} \subset \mathcal{T}$. The support set contains $|\mathcal{S}| = N_S \cdot K$ random examples (N_S random examples from each of the K classes). The query set contains $|\mathcal{Q}| = N_Q \cdot K$ random examples, sampled from the same K classes. The goal of FSL is then to predict the class of samples $\boldsymbol{x} \in \mathcal{Q}$ by exploiting the labeled support set $\mathcal{S}$, using a model trained on the base classes $\mathcal{C}_{\text{Base}}$. We assume a fixed feature extractor, trained on the base classes, which maps the raw input data to the representations $\boldsymbol{x}_i$.

4.1. noHub: Uniform Hyperspherical Structure-preserving Embeddings

We design an embedding method that encourages uniformity on the hypersphere, and simultaneously preserves local similarity structure. Given the support and query rep-

[2]Our results assume hyperspheres with unit radius, but can easily be extended to hyperspheres with arbitrary radii.

[3]The proofs for all propositions are included in the supplementary.

resentations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^k$, $n = K(N_S + N_Q)$, we wish to find suitable embeddings $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n \in \mathbb{S}_d$, where local similarities are preserved. For both representations and embeddings, we quantify similarities using a softmax over pairwise cosine similarities

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2}, \quad p_{i|j} = \frac{\exp(\kappa_i \frac{\boldsymbol{x}_i^\top \boldsymbol{x}_j}{||\boldsymbol{x}_i|| \cdot ||\boldsymbol{x}_j||})}{\sum_{l,m} \exp(\kappa_i \frac{\boldsymbol{x}_l^\top \boldsymbol{x}_m}{||\boldsymbol{x}_l|| \cdot ||\boldsymbol{x}_m||})} \tag{4}$$

and

$$q_{ij} = \frac{\exp(\kappa \boldsymbol{z}_i^\top \boldsymbol{z}_j)}{\sum_{l,m} \exp(\kappa \boldsymbol{z}_l^\top \boldsymbol{z}_m)}, \tag{5}$$

where κ_i is chosen such that the effective number of neighbours of $\boldsymbol{x}_i$ equals a pre-defined perplexity[4]. As in [27, 30], local similarity preservation can now be achieved by minimizing the Kullback-Leibler (KL) divergence between the p_{ij} and the q_{ij}

$$KL(P||Q) = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{6}$$

However, instead of directly minimizing $KL(P||Q)$, we find that the minimization problem is equivalent to minimizing the sum of two loss functions[5]

$$\underset{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n \in \mathbb{S}_d}{\arg\min}\ KL(P||Q) = \underset{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n \in \mathbb{S}_d}{\arg\min}\ \mathcal{L}_{\text{LSP}} + \mathcal{L}_{\text{Unif}} \tag{7}$$

where

$$\mathcal{L}_{\text{LSP}} = -\kappa \sum_{i,j} p_{ij} \boldsymbol{z}_i^\top \boldsymbol{z}_j, \tag{8}$$

$$\mathcal{L}_{\text{Unif}} = \log \sum_{l,m} \exp(\kappa \boldsymbol{z}_l^\top \boldsymbol{z}_m). \tag{9}$$

In Sec. 5 we provide a thorough theoretical analysis of these losses, and how they relate to LSP and uniformity on the hypersphere. Essentially, $\mathcal{L}_{\text{LSP}}$ is responsible for the local similarity preservation by ensuring that the embedding similarities ($\boldsymbol{z}_i^\top \boldsymbol{z}_j$) are high whenever the representation similarities (p_{ij}) are high. $\mathcal{L}_{\text{Unif}}$ on the other hand, can be interpreted as a negative entropy on $\mathbb{S}_d$, and is thus minimized when the embeddings are uniformly distributed on $\mathbb{S}_d$. This is discussed in more detail in Sec. 5.

Based on the decomposition of the KL divergence, and the subsequent interpretation of the two terms, we formulate the loss in noHub as the following tradeoff between LSP and uniformity

$$\mathcal{L}_{\text{noHub}} = \alpha \mathcal{L}_{\text{LSP}} + (1 - \alpha) \mathcal{L}_{\text{Unif}} \tag{10}$$

[4]Details on the computation of the κ_i are provided in the supplementary.
[5]Intermediate steps are provided in the supplementary.

Input: Features $\in \mathbb{R}^k$, $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$; perplexity, P; number of iterations, T ; learning rate, η.
Output: Embeddings $\in \mathbb{S}_d$, $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$
Compute p_{ij} from Eq (4)
Initialize solution $\boldsymbol{Z}^0 = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ with PCA
for $i \leftarrow 1$ ***to*** T **do**
  Compute q_{ij} from Eq. (5)
  Compute gradients $\frac{\mathrm{d}\mathcal{L}_{\text{noHub}}}{\mathrm{d}\boldsymbol{Z}}$, using loss from Eq. (10)
  Update $\boldsymbol{Z}^t$ using the ADAM optimizer with learning rate η [15]
  Re-normalize elements of $\boldsymbol{Z}^t$ using L_2 normalization
end
return $\boldsymbol{Z}^T$

Algorithm 1: noHub algorithm for embeddings on the hypersphere

where α is a weight parameter quantifying the tradeoff. $\mathcal{L}_{\text{noHub}}$ can then be optimized directly with gradient descent. The entire procedure is outlined in Algorithm 1.

4.2. noHub-S: noHub with Support labels

In order to strengthen the class structure in the embedding space, we modify $\mathcal{L}_{\text{LSP}}$ and $\mathcal{L}_{\text{Unif}}$ by exploiting the additional information provided by the support labels. For $\mathcal{L}_{\text{LSP}}$, we change the similarity function in p_{ij} such that

$$p_{i|j} = \frac{\exp(\kappa_i s_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{x}_j))}{\sum_{l,m} \exp(\kappa_i s_{\boldsymbol{x}}(\boldsymbol{x}_l, \boldsymbol{x}_m))} \tag{11}$$

where

$$s_{\boldsymbol{x}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} 1 & \text{if } \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{S}, \text{ and } y_i = y_j \\ -1 & \text{if } \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{S}, \text{ and } y_i \neq y_j \\ \boldsymbol{x}_i^\top \boldsymbol{x}_j & \text{otherwise} \end{cases}. \tag{12}$$

With this, we encourage embeddings for support samples in the *same class* to be maximally similar, and support samples in *different classes* to be maximally dissimilar. Similarly, for $\mathcal{L}_{\text{Unif}}$

$$\mathcal{L}_{\text{Unif}} = \log \sum_{l,m} \exp(\kappa s_{\boldsymbol{z}}(\boldsymbol{z}_i, \boldsymbol{z}_j)) \tag{13}$$

where

$$s_{\boldsymbol{z}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \begin{cases} -\infty, & \text{if } \boldsymbol{z}_i, \boldsymbol{z}_j \in \mathcal{S}, \text{ and } y_i = y_j \\ \varepsilon\, \boldsymbol{z}_i^\top \boldsymbol{z}_j, & \text{if } \boldsymbol{z}_i, \boldsymbol{z}_j \in \mathcal{S}, \text{ and } y_i \neq y_j \\ \boldsymbol{z}_i^\top, \boldsymbol{z}_j & \text{otherwise} \end{cases} \tag{14}$$

where ε is a hyperparameter. This puts more emphasis on between-class uniformity by weighting the similarity higher

for embeddings belonging to different classes ($\varepsilon > 1$), and ignoring the similarity between embeddings belonging to the same class[6]. The final loss function is the same as Eq. (10), but with the additional label-informed similarities in Eqs. (11)–(14).

5. Theoretical Results

In this section we provide a theoretical analysis of $\mathcal{L}_{\text{LSP}}$ and $\mathcal{L}_{\text{Unif}}$. Based on our analysis, we interpret these losses with regards to the Laplacian Eigenmaps algorithm and Rényi entropy, respectively.

Proposition 3. *Let $W_{ij} = \frac{1}{2}\kappa p_{ij}$, where $\sum_{i,j} p_{ij} = 1$, and let $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n \in \mathbb{S}_d$. Then we have*

$$\mathcal{L}_{\text{LSP}} = \sum_{i,j} \|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2 W_{ij} - \kappa. \tag{15}$$

Proposition 4 (Minimizing $\mathcal{L}_{\text{Unif}}$ maximizes entropy)**.** *Let $H_2(\cdot)$ be the 2-order Rényi entropy, estimated with a kernel density estimator using a Gaussian kernel. Then*

$$\underset{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n \in \mathbb{S}_d}{\arg\min}\ \mathcal{L}_{\text{Unif}} = \underset{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n \in \mathbb{S}_d}{\arg\max}\ H_2(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n). \tag{16}$$

Definition 2 (Normalized counting measure)**.** *The normalized counting measure associated with a set B on A is*

$$\nu_B(A) = \frac{|B \cap A|}{|B|} \tag{17}$$

Definition 3 (Normalized surface area measure on $\mathbb{S}_d$)**.** *The normalized surface area measure on the hyperspehere $\mathbb{S}_d \subset \mathbb{R}^d$, of a subset $S' \subset \mathbb{S}_d$ is*

$$\sigma_d(S') = \frac{\int_{S'} \mathrm{d}S}{\int_{\mathbb{S}_d} \mathrm{d}S} = A_d^{-1} \int_{S'} \mathrm{d}S \tag{18}$$

where A_d is defined as in Eq. (1), and $\int \mathrm{d}S$ denotes the surface integral on $\mathbb{S}_d$.

Definition 4 (Weak* convergence of measures [32])**.** *A sequence of Borel measures $\{\mu_n\}_{n=1}^{\infty}$ in $\mathbb{R}^d$ converges weak* to a Borel measure μ, if for all continuous functions $f : \mathbb{R}^d \to \mathbb{R}$,*

$$\lim_{n\to\infty} \int f(x)\mathrm{d}\mu_n(x) = \int f(x)\mathrm{d}\mu(x) \tag{19}$$

Proposition 5 (Minimizer of $\mathcal{L}_{\text{Unif}}$)**.** *For each $n > 0$, the n point minimizer of $\mathcal{L}_{\text{Unif}}$ is*

$$\boldsymbol{z}_1^\star, \ldots, \boldsymbol{z}_n^\star = \underset{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n \in \mathbb{S}_d}{\arg\min}\ \mathcal{L}_{\text{Unif}}. \tag{20}$$

Then $\nu_{\{\boldsymbol{z}_1^\star,\ldots,\boldsymbol{z}_n^\star\}}$ converge weak* to σ_d as $n \to \infty$.

[6] Although any constant value would achieve the same result, we set the similarity to $-\infty$ in this case to remove the contribution to the final loss.

Interpretation of Proposition 3–5. Proposition 3 states an alternative formulation of $\mathcal{L}_{\text{LSP}}$, under the hyperspherical assumption. We recognize this formulation as the loss function in Laplacian Eigenmaps [2], which is known to produce *local similarity-preserving* embeddings from graph data. When unconstrained, this loss has a trivial solution where the embeddings for all representations are equal. This is avoided in our case since $\mathcal{L}_{\text{noHub}}$ (Eq. (10)) can be interpreted as the Lagrangian of minimizing $\mathcal{L}_{\text{LSP}}$ subject to a specified level of *entropy*, by Proposition 4.

Finally, Proposition 5 states that the normalized counting measure associated with the set of points that minimize $\mathcal{L}_{\text{Unif}}$, converges to the normalized surface area measure on the sphere. Since $u_{\mathbb{S}_d}$ is the density function associated with this measure, the points that minimize $\mathcal{L}_{\text{Unif}}$ will tend to be uniform on the sphere. Consequently, minimizing $\mathcal{L}_{\text{LSP}}$ also minimizes hubness, by Propositions 1 and 2.

6. Experiments

6.1. Setup

Implementation details. Our implementation is in PyTorch [20]. We optimize noHub and noHub-S for $T = 150$ iterations, using the Adam optimizer [15] with learning rate $\eta = 0.1$. The other hyperparameters were chosen based on validation performance on the respective datasets[7]. We analyze the effect of α in Sec. 6.2. Analyses of the κ and ε hyperparameters are provided in the supplementary.

Initialization. Since noHub and noHub-S reduce the embedding dimensionality ($d = 400$), we initialize embeddings with Principal Component Analysis (PCA) [13], instead of a naïve, random initialization. The PCA initialization is computationally efficient, and approximately preserves global structure. It also resulted in faster convergence and better performance, compared to random initialization.

Base feature extractors. We use the standard networks ResNet-18 [11] and Wide-Res28-10 [37] as the base feature extractors with pretrained weights from [28] and [18], respectively.

Datasets. Following common practice, we evaluate FSL performance on the *mini-ImageNet (mini)* [29], *tiered-ImageNet (tiered)* [23], and *CUB-200 (CUB)* [34] datasets.

Classifiers. We evaluate the baseline embeddings and our proposed methods using both established and recent FSL classifiers: *SimpleShot* [33], *LaplacianShot* [40], *α−TIM* [28], *Oblique Manifold (OM)* [21], *iLPC* [16], and *SIAMESE* [39].

Baseline Embeddings. We compare our proposed method with a wide range of techniques for embedding the base features: *None* (No embedding of base features), *L2* [33], *Centered L2* [33], *ZN* [7], *ReRep* [5], *EASE* [39], and *TCPR* [35].

[7] Hyperparameter configurations for all experiments are included in the supplementary.

Embedding	Feature Extractor	mini		tiered		CUB	
		1-shot ↑	5-shot ↑	1-shot ↑	5-shot ↑	1-shot ↑	5-shot ↑
None	ResNet-18	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*
L2 (ArXiv'19 [33])	ResNet-18	73.77 (0.24)	83.14 (0.14)	80.46 (0.26)	87.04 (0.16)	83.1 (0.23)	89.48 (0.12)
CL2 (ArXiv'19 [33])	ResNet-18	75.56 (0.26)	84.04 (0.15)	82.1 (0.26)	87.9 (0.16)	84.35 (0.24)	90.14 (0.12)
ZN (ICCV'21 [7])	ResNet-18	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*
ReRep (ICML'21 [5])	ResNet-18	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*
EASE (CVPR'22 [39])	ResNet-18	76.05 (0.27)	84.61 (0.15)	82.57 (0.27)	88.33 (0.16)	85.24 (0.24)	90.42 (0.12)
TCPR (NeurIPS'22 [35])	ResNet-18	75.99 (0.26)	84.39 (0.15)	82.65 (0.26)	88.26 (0.16)	85.34 (0.23)	90.5 (0.11)
noHub (Ours)	ResNet-18	76.65 (0.28)	84.05 (0.16)	82.94 (0.27)	87.87 (0.17)	**85.88 (0.24)**	90.34 (0.12)
noHub-S (Ours)	ResNet-18	**76.68 (0.28)**	**84.67 (0.15)**	**83.09 (0.27)**	**88.43 (0.16)**	85.81 (0.24)	**90.52 (0.12)**
None	WideRes28-10	45.69 (0.31)	58.82 (0.31)	75.29 (0.28)	82.56 (0.22)	61.36 (0.55)	82.22 (0.37)
L2 (ArXiv'19 [33])	WideRes28-10	80.2 (0.23)	87.11 (0.13)	80.89 (0.26)	87.34 (0.15)	91.98 (0.18)	94.15 (0.1)
CL2 (ArXiv'19 [33])	WideRes28-10	75.23 (0.27)	83.99 (0.16)	79.59 (0.27)	86.71 (0.16)	92.17 (0.18)	94.48 (0.09)
ZN (ICCV'21 [7])	WideRes28-10	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*	20.0 (0.0)*
ReRep (ICML'21 [5])	WideRes28-10	36.69 (0.28)	36.41 (0.3)	67.41 (0.29)	76.49 (0.24)	57.62 (0.56)	60.36 (0.6)
EASE (CVPR'22 [39])	WideRes28-10	81.19 (0.25)	87.82 (0.13)	82.04 (0.26)	88.06 (0.16)	91.99 (0.19)	94.36 (0.09)
TCPR (NeurIPS'22 [35])	WideRes28-10	81.27 (0.24)	87.8 (0.13)	81.89 (0.26)	87.95 (0.16)	91.91 (0.18)	94.25 (0.1)
noHub (Ours)	WideRes28-10	81.97 (0.25)	87.78 (0.14)	82.8 (0.27)	87.99 (0.17)	92.53 (0.18)	94.56 (0.09)
noHub-S (Ours)	WideRes28-10	**82.0 (0.26)**	**88.03 (0.13)**	**82.85 (0.27)**	**88.31 (0.16)**	**92.63 (0.18)**	**94.69 (0.09)**

Table 1. Accuracies (Confidence interval) with the SIAMESE [39] classifier for different embedding approaches. Best and second best performance are denoted in **bold** and underlined, respectively. *The SIAMESE classifier is sensitive to the norm of the embedding, thus leading to detrimental performance for some of the embedding approaches.

Evaluation protocol. We follow the standard evaluation protocol in FSL and calculate the accuracy for 1-shot and 5-shot classification with 15 images per class in the query set. We evaluate on 10000 episodes, as is standard practice in FSL. Additionally, we evaluate the hubness of the representations after embedding using two common hubness metrics, namely the skewness (Sk) of the k-occurrence distribution [22] and the hub occurrence (HO) [8], which measures the percentage of hubs in the nearest neighbour lists of all points.

6.2. Results

Comparison to the state-of-the-art. To illustrate the effectiveness of noHub and noHub-S as an embedding approach for FSL, we consider the current state-of-the-art FSL method, which leverages the EASE embedding and obtains query predictions with SIAMESE [39]. We replace EASE with our proposed embedding approaches noHub and noHub-S, as well as other baseline embeddings, and evaluate performance on all datasets in the 1 and 5-shot setting. As shown in Table 1, noHub and noHub-S outperform all baseline approaches in both settings across all datasets, illustrating noHub's and noHub-S' ability to provide useful FSL embeddings, and updating the state-of-the-art in transductive FSL.

Aggregated FSL performance. To further evaluate the general applicability of noHub and noHub-S as embedding approaches, we perform extensive experiments for all classifiers and all baseline embeddings on all datasets. Tables 2a and 2b provide the results averaged over classifiers[8]. To clearly present the results, we aggregate the accuracy and a ranking *score* for each embedding method across all classifiers. The ranking score is calculated by performing a paired Student's t-test between all pairwise embedding methods for each classifier. We then average the ranking scores across all classifiers. A high ranking score then indicates that a method often significantly outperforms the competing embedding methods. We set the significance level to 5%. noHub and noHub-S consistently outperform previous embedding approaches – sometimes by a large margin. Overall, we further observe that noHub-S outperforms noHub in most settings and is particular beneficial in the 1-shot setting, which is more challenging, given that fewer samples are likely to generate noisy embeddings.

Hubness metrics. To further validate noHub's and noHub-S' ability to reduce hubness, we follow the same procedure of aggregating results for the hubness metrics and average over classifiers. Compared to the current state-of-the-art embedding approaches, Table 3 illustrates that noHub and noHub-S consistently result in embeddings with lower hubness.

Visualization of similarity matrices. As discussed in Sec. 4, completely eliminating hubness by distributing points uniformly on the hypersphere is not sufficient to obtain good FSL performance. Instead, representations need to also capture the inherent class structure of the data. To further evaluate the embedding approaches, we therefore compute the pairwise inner products for the embeddings of a random 5-shot episode on tiered-ImageNet with ResNet-18 features in Figure 3. It can be observed that the block structure is considerably more distinct for noHub and noHub-S, with

[8]The detailed results for all classifiers are provided in the supplementary.

	Embedding	mini Acc↑	mini Score↑	tiered Acc↑	tiered Score↑	CUB Acc↑	CUB Score↑
ResNet18	None	55.74	0.17	62.61	0.0	63.78	0.17
	L2 (ArXiv'19 [33])	68.22	2.33	75.94	2.17	78.09	2.33
	CL2 (ArXiv'19 [33])	69.56	2.83	76.97	3.0	78.26	2.83
	ZN (ICCV'21 [7])	60.0	2.33	66.21	2.5	67.43	2.67
	ReRep (ICML'21 [5])	60.76	4.0	67.07	3.67	69.6	4.17
	EASE (CVPR'22 [39])	69.63	3.67	77.05	4.0	78.84	3.67
	TCPR (NeurIPS'22 [35])	69.97	4.0	77.18	3.33	78.83	4.0
	noHub (Ours)	72.58	6.83	79.77	6.83	81.91	6.83
	noHub-S (Ours)	**73.64**	**7.67**	**80.6**	**7.67**	**83.1**	**7.67**
WideRes28-10	None	63.59	1.0	71.29	0.83	79.23	1.17
	L2 (ArXiv'19 [33])	74.3	3.0	76.19	2.67	88.61	3.5
	CL2 (ArXiv'19 [33])	71.32	1.33	75.17	2.0	88.52	3.33
	ZN (ICCV'21 [7])	64.27	2.5	65.64	2.5	76.0	1.5
	ReRep (ICML'21 [5])	65.51	3.0	71.83	3.17	83.1	3.5
	EASE (CVPR'22 [39])	74.95	4.33	76.59	3.67	88.51	3.5
	TCPR (NeurIPS'22 [35])	75.64	4.83	76.51	4.0	88.22	2.5
	noHub (Ours)	78.22	7.0	79.76	7.0	90.25	5.67
	noHub-S (Ours)	**79.24**	**7.67**	**80.46**	**7.67**	**90.82**	**7.67**

(a) 1-shot

	Embedding	mini Acc↑	mini Score↑	tiered Acc↑	tiered Score↑	CUB Acc↑	CUB Score↑
ResNet18	None	69.83	0.83	74.38	0.67	76.01	1.17
	L2 (ArXiv'19 [33])	81.58	2.33	86.05	1.83	88.43	2.83
	CL2 (ArXiv'19 [33])	81.95	2.67	86.43	3.0	88.49	2.5
	ZN (ICCV'21 [7])	71.49	4.0	75.32	3.83	76.92	3.5
	ReRep (ICML'21 [5])	70.25	2.5	74.52	1.83	76.43	2.5
	EASE (CVPR'22 [39])	81.84	3.5	86.4	3.17	88.57	3.5
	TCPR (NeurIPS'22 [35])	82.1	4.0	86.54	3.83	88.79	4.33
	noHub (Ours)	82.58	5.5	86.9	4.5	**89.13**	**6.0**
	noHub-S (Ours)	**82.61**	**6.5**	**87.13**	**6.67**	88.93	5.33
WideRes28-10	None	78.77	1.5	84.1	1.67	89.49	1.67
	L2 (ArXiv'19 [33])	85.65	4.0	86.29	3.83	93.47	3.67
	CL2 (ArXiv'19 [33])	83.14	1.33	85.47	1.5	93.49	4.0
	ZN (ICCV'21 [7])	74.61	4.33	75.34	5.0	81.02	3.17
	ReRep (ICML'21 [5])	73.86	1.83	81.51	1.67	87.2	2.0
	EASE (CVPR'22 [39])	85.51	3.5	86.29	3.33	93.34	3.5
	TCPR (NeurIPS'22 [35])	86.03	**6.0**	86.37	4.0	93.3	3.0
	noHub (Ours)	**86.44**	5.67	**87.07**	5.5	93.65	4.17
	noHub-S (Ours)	85.95	5.5	87.05	**5.83**	**93.76**	**5.0**

(b) 5-shot

Table 2. Aggregated FSL performance for all embedding approaches on the mini-ImageNet, tiered-ImageNet, and CUB-200 datasets. Results are averaged over FSL classifiers. Best and second best performance are denoted in **bold** and underlined, respectively.

	Embedding	mini Sk↓	mini HO↓	tiered Sk↓	tiered HO↓	CUB Sk↓	CUB HO↓
ResNet18	None	1.349	0.407	1.211	0.408	0.887	0.341
	L2 (ArXiv'19 [33])	0.937	0.301	0.812	0.265	0.691	0.236
	CL2 (ArXiv'19 [33])	0.667	0.233	0.679	0.249	0.549	0.201
	ZN (ICCV'21 [7])	0.68	0.231	0.698	0.264	0.564	0.216
	ReRep (ICML'21 [5])	3.655	0.548	3.604	0.549	3.565	0.513
	EASE (CVPR'22 [39])	0.521	0.16	0.479	0.158	0.466	0.153
	TCPR (NeurIPS'22 [35])	0.651	0.228	0.65	0.25	0.532	0.204
	noHub (Ours)	0.315	**0.095**	0.303	**0.102**	0.32	**0.112**
	noHub-S (Ours)	**0.276**	0.13	**0.283**	0.127	**0.296**	0.162
WideRes28-10	None	1.6	0.459	1.81	0.494	1.073	0.369
	L2 (ArXiv'19 [33])	0.781	0.296	0.737	0.275	0.475	0.228
	CL2 (ArXiv'19 [33])	0.981	0.288	0.817	0.307	0.52	0.267
	ZN (ICCV'21 [7])	0.73	0.287	0.769	0.302	0.517	0.263
	ReRep (ICML'21 [5])	3.56	0.704	3.55	0.777	3.026	0.47
	EASE (CVPR'22 [39])	0.47	0.177	0.477	0.175	0.437	0.213
	TCPR (NeurIPS'22 [35])	0.589	0.236	0.685	0.264	0.477	0.231
	noHub (Ours)	0.29	**0.111**	0.301	**0.111**	0.188	**0.108**
	noHub-S (Ours)	**0.258**	0.148	**0.274**	0.135	**0.162**	0.13

(a) 1-shot

	Embedding	mini Sk↓	mini HO↓	tiered Sk↓	tiered HO↓	CUB Sk↓	CUB HO↓
ResNet18	None	1.436	0.422	1.339	0.432	0.987	0.364
	L2 (ArXiv'19 [33])	1.04	0.318	0.914	0.287	0.812	0.263
	CL2 (ArXiv'19 [33])	0.786	0.264	0.821	0.28	0.698	0.236
	ZN (ICCV'21 [7])	0.806	0.264	0.839	0.296	0.716	0.25
	ReRep (ICML'21 [5])	1.631	0.863	1.721	0.872	1.432	0.869
	EASE (CVPR'22 [39])	0.624	0.186	0.598	0.183	0.607	0.186
	TCPR (NeurIPS'22 [35])	0.78	0.259	0.796	0.283	0.687	0.235
	noHub (Ours)	0.286	0.096	0.289	0.104	**0.329**	0.12
	noHub-S (Ours)	**0.25**	**0.074**	**0.213**	**0.078**	0.433	**0.097**
WideRes28-10	None	1.709	0.473	1.937	0.51	1.16	0.395
	L2 (ArXiv'19 [33])	0.887	0.322	0.86	0.305	0.632	0.266
	CL2 (ArXiv'19 [33])	1.12	0.318	0.956	0.337	0.701	0.31
	ZN (ICCV'21 [7])	0.858	0.32	0.912	0.335	0.699	0.305
	ReRep (ICML'21 [5])	1.597	0.819	1.617	0.846	1.299	0.549
	EASE (CVPR'22 [39])	0.579	0.199	0.585	0.193	0.572	0.241
	TCPR (NeurIPS'22 [35])	0.717	0.27	0.815	0.294	0.634	0.264
	noHub (Ours)	**0.294**	0.115	**0.298**	**0.115**	**0.195**	**0.1**
	noHub-S (Ours)	0.494	**0.103**	0.407	0.12	0.421	0.127

(b) 5-shot

Table 3. Aggregated hubness metrics for all embedding approaches on the Mini-ImageNet, Tiered-ImageNet and CUB-200 dataset. Results are averaged over FSL classifiers. Best and second best performance are denoted in **bold** and underlined, respectively.

noHub-S slightly improving upon noHub. These results indicate that (i) samples are more uniform, indicating the reduced hubness; and (ii) classes are better separated, due to the local similarity preservation.

Tradeoff between uniformity and similarity preservation. We analyze the effect of α on the tradeoff between LSP and Uniformity in the loss function in Eq. (10), on tiered-ImageNet with ResNet-18 features in the 5-shot setting and with the SIAMESE [39] classifier. The results are visualized in Figure 4. We notice a sharp increase in performance when we have a high emphasis on uniformity. This demonstrates the impact of hubness on accuracy in FSL performance. As we keep increasing the emphasis on LSP, however, after a certain point we notice a sharp drop off in performance. This is due to the fact that the classifier does not take into account the uniformity constraint on the features, resulting in a large number of misclassifications. In general, we observe that noHub-S is slightly more robust compared to noHub.

Increasing number of classes. We analyze the behavior of noHub and noHub-S for an increasing number of classes (ways) on the tiered-ImageNet dataset with SIAMESE [39]

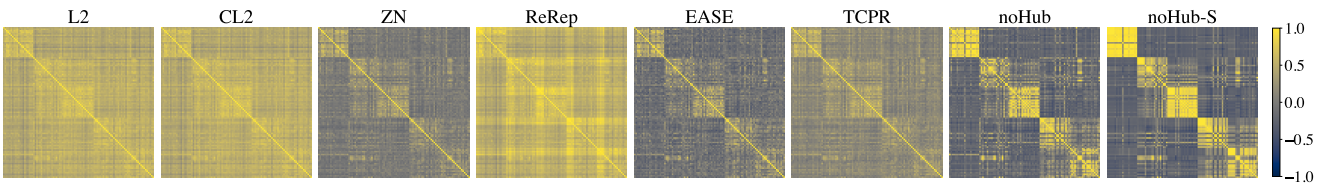

Figure 3. Inner product matrices between features for a random episode for all embedding approaches.

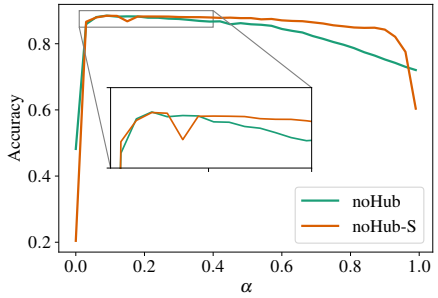

Figure 4. Accuracies for different values of the weighting parameter, α, which quantifies the tradeoff between $\mathcal{L}_{\text{LSP}}$ and $\mathcal{L}_{\text{Unif}}$.

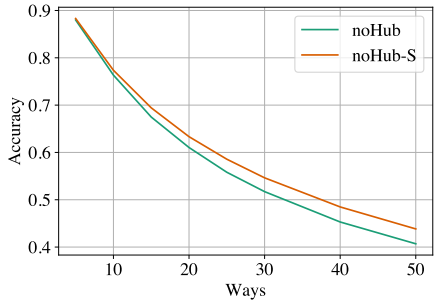

Figure 5. Accuracies for an increasing number of classes (ways) for noHub and noHub-S.

as classifier. While classification accuracy generally decreases with an increasing number of classes, which is expected, we observe from Figure 5 that noHub-S has a slower decay and is able to leverage the label guidance to obtain better performance for a larger number of classes.

Effect of label information in $\mathcal{L}_{\text{LSP}}$ and $\mathcal{L}_{\text{Unif}}$. To validate the effectiveness of using label guidance in noHub-S, we study the result of including label information in $\mathcal{L}_{\text{LSP}}$ and $\mathcal{L}_{\text{Unif}}$ (Eqs. (11)–(14)). We note that the default setting of noHub is that none of the two losses include label information. Ablation experiments are performed on tiered-ImageNet with the ResNet-18 feature extractor and the SimpleShot and SIAMESE classifier [39]. In Table 4, we generally see improvements of noHub-S when *both* the loss terms are label-informed, indicating the usefulness of label guidance.

We further observe that incorporating label information in $\mathcal{L}_{\text{Unif}}$ tends to have a larger contribution than doing the same for $\mathcal{L}_{\text{LSP}}$. This aligns with our observations in Figure 4, where a small α yielded the best performance.

	Label-informed		SimpleShot [33]		SIAMESE [39]	
	$\mathcal{L}_{\text{LSP}}$	$\mathcal{L}_{\text{Unif}}$	1-shot↑	5-shot↑	1-shot↑	5-shot↑
noHub	–	–	76.72 (0.23)	**86.31 (0.16)**	82.94 (0.27)	87.87 (0.17)
noHub-S	✓	–	78.25 (0.24)	85.46 (0.16)	82.56 (0.28)	88.07 (0.17)
noHub-S	–	✓	78.33 (0.23)	86.15 (0.15)	82.81 (0.27)	**88.43 (0.16)**
noHub-S	✓	✓	**78.35 (0.23)**	86.22 (0.15)	**83.09 (0.27)**	**88.43 (0.16)**

Table 4. Ablation study with the label-informed losses in noHub-S. Check marks (✓) indicate that the loss uses information from the support labels.

7. Conclusion

In this paper we have addressed the hubness problem in FSL. We have shown that hubness is eliminated by embedding representations uniformly on the hypersphere. The hyperspherical uniform distribution has zero mean and zero density gradient at all points along all directions tangent to the hypersphere – both of which are identified as causes of hubness in previous work [9, 22]. Based on our theoretical findings about hubness and hyperspheres, we proposed two new methods to embed representations on the hypersphere for FSL. The proposed noHub and noHub-S leverage a decomposition of the KL divergence between similarity distributions, and optimize a tradeoff between LSP and uniformity on the hypersphere – thus reducing hubness while maintaining the class structure in the representation space. We have provided theoretical analyses and interpretations of the LSP and uniformity losses, proving that they optimize LSP and uniformity, respectively. We comprehensively evaluate the proposed methods on several datasets, features extractors, and classifiers, and compare to a number of recent state-of-the-art baselines. Our results illustrate the effectiveness of our proposed methods and show that we achieve state-of-the-art performance in transductive FSL.

Acknowledgements

This work was financially supported by the Research Council of Norway (RCN), through its Centre for Research-based Innovation funding scheme (Visual Intelligence, grant no. 309439), and Consortium Partners. It was further funded by RCN FRIPRO grant no. 315029, RCN IKTPLUSS grant no. 303514, and the UiT Thematic Initiative "Data-Driven Health Technology".

References

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

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 2003. 5

[3] Malik Boudiaf, Ziko Imtiaz Masud, Jérôme Rony, José Dolz, Pablo Piantanida, and Ismail Ben Ayed. Transductive Information Maximization For Few-Shot Learning. In *NeurIPS*, 2020. 1, 2

[4] Philip Chikontwe, Soopil Kim, and Sang Hyun Park. CAD: Co-Adapting Discriminative Features for Improved Few-Shot Classification. In *CVPR*, 2022. 1

[5] Wentao Cui and Yuhong Guo. Parameterless Transductive Feature Re-representation for Few-Shot Learning. In *ICML*, 2021. 1, 2, 5, 6, 7

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1

[7] Nanyi Fei, Yizhao Gao, Zhiwu Lu, and Tao Xiang. Z-Score Normalization, Hubness, and Few-Shot Learning. In *ICCV*, 2021. 1, 2, 5, 6, 7

[8] Arthur Flexer and Dominik Schnitzer. Choosing ℓ^p norms in high-dimensional spaces based on hub analysis. *Neurocomputing*, 2015. 6

[9] Kazuo Hara, Ikumi Suzuki, Kei Kobayashi, Kenji Fukumizu, and Milos Radovanovic. Flattening the Density Gradient for Eliminating Spatial Centrality to Reduce Hubness. In *AAAI*, 2016. 2, 8

[10] Kazuo Hara, Ikumi Suzuki, Masashi Shimbo, Kei Kobayashi, Kenji Fukumizu, and Milos Radovanovic. Localized Centering: Reducing Hubness in Large-Sample Data. In *AAAI*, 2015. 1

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 5

[12] Shell Xu Hu, Da Li, Jan Stuhmer, Minyoung Kim, and Timothy M Hospedales. Pushing the Limits of Simple Pipelines for Few-Shot Learning: External Data and Fine-Tuning Make a Difference. In *CVPR*, 2022. 1

[13] Ian T Jolliffe. *Principal component analysis for special types of data*. Springer, 2002. 5

[14] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *CVPR*, 2019. 1

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4, 5

[16] Michalis Lazarou, Tania Stathaki, and Yannis Avrithis. Iterative Label Cleaning for Transductive and Semi-Supervised Few-Shot Learning. In *ICCV*, 2021. 1, 5

[17] Duong H Le, Khoi D Nguyen, and Khoi Nguyen. POODLE: Improving Few-shot Learning via Penalizing Out-of-Distribution Samples. In *NeurIPS*, 2021. 1

[18] Puneet Mangla, Mayank Singh, Abhishek Sinha, Nupur Kumari, Vineeth N Balasubramanian, and Balaji Krishnamurthy. Charting the Right Manifold: Manifold Mixup for Few-shot Learning. In *WACV*, 2020. 5

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

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5

[21] Guodong Qi, Huimin Yu, Zhaohui Lu, and Shuzhao Li. Transductive Few-Shot Classification on the Oblique Manifold. In *ICCV*, 2021. 1, 5

[22] Miloš Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. Hubs in Space: Popular Nearest Neighbors in High-Dimensional Data. *JMLR*, 2010. 1, 2, 6, 8

[23] 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

[24] Yutaro Shigeto, Ikumi Suzuki, Kazuo Hara, Masashi Shimbo, and Yuji Matsumoto. Ridge Regression, Hubness, and Zero-Shot Learning. In *ECML-PKDD*, 2015. 1

[25] Ikumi Suzuki, Kazuo Hara, Masashi Shimbo, Marco Saerens, and Kenji Fukumizu. Centering Similarity Measures to Reduce Hubs. In *EMNLP*, 2013. 1

[26] Ran Tao, Han Zhang, Yutong Zheng, and Marios Savvides. Powering Finetuning in Few-Shot Learning: Domain-Agnostic Bias Reduction with Selected Sampling. In *AAAI*, 2022. 1

[27] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *JMLR*, 2008. 4

[28] Olivier Veilleux, Malik Boudiaf, Pablo Piantanida, and Ismail Ben Ayed. Realistic evaluation of transductive few-shot learning. In *NeurIPS*, 2021. 1, 5

[29] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. In *NeurIPS*, 2016. 1,

5

[30] Mian Wang and Dong Wang. VMF-SNE: embedding for spherical data. *arxiv:1507.08379 [cs]*, 2015. 4

[31] Ruohan Wang, Massimiliano Pontil, and Carlo Ciliberto. The Role of Global Labels in Few-Shot Classification and How to Infer Them. In *NeurIPS*, 2021. 1

[32] Tongzhou Wang and Phillip Isola. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *ICML*, 2020. 2, 5

[33] Yan Wang, Wei-Lun Chao, Kilian Q. Weinberger, and Laurens van der Maaten. SimpleShot: Revisiting Nearest-Neighbor Classification for Few-Shot Learning. *arXiv:1911.04623 [cs]*, 2019. 1, 2, 5, 6, 7, 8

[34] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. Technical Report CNS-TR-201, Caltech, 2010. 5

[35] Jing Xu, Xu Luo, Xinglin Pan, Wenjie Pei, Yanan Li, and Zenglin Xu. Alleviating the Sample Selection Bias in Few-shot Learning by Removing Projection to the Centroid. In *NeurIPS*, 2022. 1, 2, 5, 6, 7

[36] 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. 2

[37] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 5

[38] 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. 2

[39] Hao Zhu and Piotr Koniusz. EASE: Unsupervised Discriminant Subspace Learning for Transductive Few-Shot Learning. In *CVPR*, 2022. 1, 2, 5, 6, 7, 8

[40] Imtiaz Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed. Laplacian Regularized Few-Shot Learning. In *ICML*, 2020. 1, 2, 5