2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA) | 979-8-3503-8220-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/DICTA60407.2023.00063

Learning Stable Task-Level Manifold for Few-Shot Learning

Siyi Yu[1,2], Wei Luo[3], Gang Li[3], Bo Yang[1,2,*]
[1]*College of Computer Science and Technology, Jilin University, China*
[2]*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, China*
[3]*School of Information Technology, Deakin University, Geelong, Australia*
ysy21@mails.jlu.edu.cn, {wei.luo, gang.li}@deakin.edu.au, ybo@jlu.edu.cn

***Abstract*—Few-shot learning (FSL) aims to learn to new concepts based on very limited data. One of the main challenges in FSL is the use of pretrained embeddings whose dimension is too high for the small sample size. While the large dimensionality helps capture diverse features in the training example, not all the features are relevant to the task at hand. Such task-irrelevant features from pre-trained embeddings can negatively impact the performance of the FSL model. To address this challenge, we propose a simple yet effective way to learn task-level manifolds that explicitly eliminate the negative effect of nuisance features. One key novelty of our approach is to address class imbalance in the training data by injecting supervised information through key virtual samples, which we named Manifold Support Points. Empirically, we demonstrate that the stability of the learnt task-level manifolds and their effectiveness in challenging tasks in FSL. Our approach shows competitive performance on several benchmark datasets, including tieredImageNet and CUB.**

***Index Terms*—Few-shot Learning, Manifold**

I. Introduction

Earlier successes of deep learning rely on the availability of an enormous amount of labelled data. But in many data mining applications, labelled data is scarce and expensive to obtain. At the same time, humans (even children) have the ability to learn complex concepts from only a few examples. Therefore, few-shot learning (FSL) [19], which aims to recognise new objects or to learn new concepts with only a few labelled examples, occupies a very important position in the future of artificial intelligence.

FSL models are often evaluated under an N-*way* K-*shot* setting, where N is the number of classes in the task and K is the number of training examples for each class. A more realistic, at the same time more challenging, setting is N-*way any-shot*, where each class can have fewer than K number of training examples. There are different approaches to FSL [25]. In this paper, we focus on FSL based on class prototypes [4], [17], [24], [30]. In this approach, a generic pretrained backbone network is used to map images to feature vectors. Then a prototype vector is produced to represent the training examples of each label class. With N prototype vectors, a new image (a *query image*) is classified based on the nearest prototype vector.

Prototype-based few-shot learning models can achieve state-of-the-art classification performance on many standard N-way K-shot problems. However, in a problem with a small K or an N-way any-shot problem, class prototypes can become unreliable. Class prototypes are normally represented as vectors in a high-dimensional Euclidean space, which is known to suffer from the "the curse of dimensionality" [1], [8]. High-dimensional embeddings from a backbone model (e.g., trained with ImageNet) contain feature dimensions irrelevant to the N classes in an FSL task. For example, Figure 1 shows that humans often show up in images in FSL, but often they are irrelevant to the target class. Features representing humans are therefore *nuisance features* that will corrupt a prototype vector.

In this paper, we propose a novel but simple and effective solution to this problem which requires no additional parameters and training data. Our approach takes advantage of the observation that in a classification problem, we don't have to perfectly capture what an object is. It suffices to capture how an object is different from the others. Therefore, we can afford to lower the feature dimension for each task at hand. More specifically, we can find a task-specific manifold with a much lower dimension than the original embedding space. This task-level manifold provides a stable platform for reliable classification. In this paper, we make the following contributions:

- We point out a key hurdle to reliable few-shot learning (FSL), nuisance features in pretrained embeddings that are irrelevant to the current task.
- We propose a two-stage few-shot learning approach to address the reliability of FSL. The first stage learns a task-level manifold that captures essential features relevant to each task from limited labelled data. The next stage is a regular prototype-based classifier on the unrolled manifold.
- To learn a stable manifold, we propose a simple yet effective way to integrate supervised information in manifold learning. It is based on introducing virtual labelled samples, which we named *Manifold Support Points* (MSP), to construct coherent neighbourhoods from limited labelled data.
- We presented a new way to generate class-imbalance data based on controlling the variance of the number of labelled data belonging to different classes in the support set, which better simulate class imbalance in real-world

979-8-3503-8220-4/23/$31.00 ©2023 IEEE
DOI 10.1109/DICTA60407.2023.00063

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 06:48:10 UTC from IEEE Xplore. Restrictions apply.

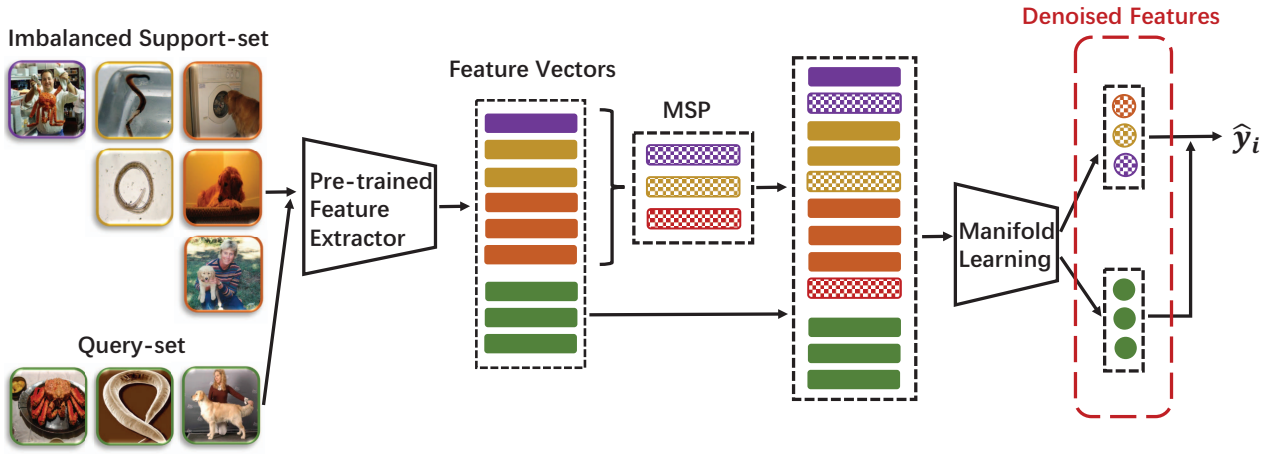


Fig. 1. Two-stage few-shot learning. The first stage learns a task-level manifold that captures essential features relevant to each task from limited labelled data. The next stage is a regular prototype-based classifier on the unrolled manifold. To learn a stable manifold, virtual labelled samples named *Manifold Support Points* (MSPs) were produced to capture the shared features for each class and added to the data mix. The projections of MSPs on the manifold are also used as stabilized prototypes for the classifier.

FSL scenarios.

II. Related Work

A. Few-Shot Learning

There are two main streams in few-shot learning, optimization-based and metric-based models. Optimization-based models like [5], [12], [19] use the support set to fine-tune the classifier, which is usually an MLP with carefully chosen initialization through a few epochs to gain task-specific embedding. However, since the publication of two important papers [24], [28], the metric-based classifier has received keen attention due to its excellent experimental results and theoretical guarantees, especially the idea of using prototypes as the centre of classes to construct a classifier. [30] introduces self-attention into the model to use prototypes of different classes in one task to modify each other and achieve reliable results. However, as the number of labelled images decreases or the level of imbalance increases, the impact of noise in the support set becomes even greater, which makes prototypes of each class unreliable and leads to poor performance.

B. Class Imbalance in Few-Shot Learning

There are two main methods to solve the problem of the imbalanced situation in few-shot learning from the perspective of over-sampling and meta-learning. [16] use data augmentation and oversampling to address the class imbalance in few-shot learning, which is not so effective and does not take into account the particularity of few-shot learning under imbalance scenarios. [10] is an optimization-based model that will produce a larger learning rate for tasks and classes with more labeled data to fine-tune the model since they believe that more information will direct the model in the right direction of change. However, it is more to improve the mean accuracy over many tasks, not trying to improve accuracy in each task.

C. Dimensionality Reduction Based on Manifold Learning

Manifold learning is a commonly used dimensionality reduction technique that is based on the manifold hypothesis, which states that high-dimensional data typically lie on a low-dimensional manifold. This thought is consistent with our idea. The essence of manifold learning is to capture the important features of data by learning the geometry of the manifold, which is exactly what we need. UMAP [14] is currently the most popular and high-performance dimension reduction technique. It not only focuses on the global structures like other dimensionality reduction techniques but also captures a large number of local structures by constructing the KNN graph for each data point.

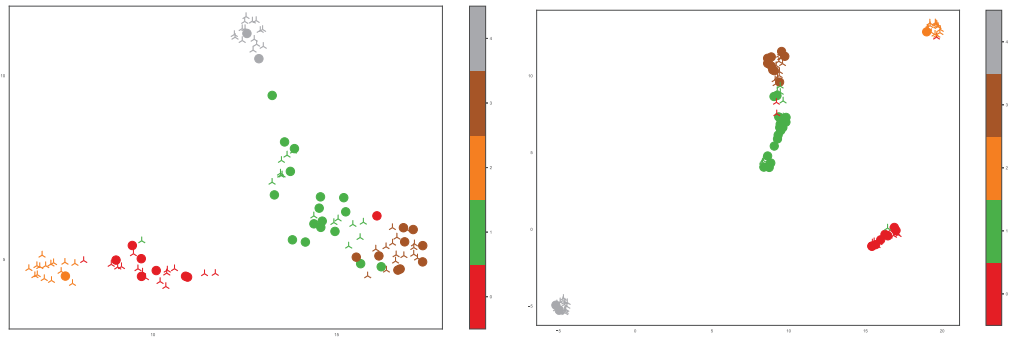
III. Two-Stage Manifold-Based Few-Shot-Learning

In this section, we introduce the problem setting and our two-stage manifold-based FSL method.

A. Problem Statement

In a typical few-shot learning setup, we assume a base data set $\mathcal{D}_{base}$, and a novel data set $\mathcal{D}_{novel}$, with $\mathcal{D}_{base} \bigcap \mathcal{D}_{novel} = \emptyset$. $\mathcal{D}_{base}$ contains a large number of labelled examples that are used to train the backbone of a model. We sampled few-shot learning tasks from $\mathcal{D}_{novel}$ which contains a small number of labelled examples (i.e. support set $\mathcal{S}$), along with some unlabeled ones (i.e. query set $\mathcal{Q}$). Classifying unlabeled examples in the query set based on the support set is called a

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 06:48:10 UTC from IEEE Xplore. Restrictions apply.



(a) Original feature vectors (b) Denoised feature vectors

Fig. 2. Feature vectors of a 5-way Any-shot task. The labeled data of each class in the support set are plotted as solid dots. In the global embedding space (a), classes have more dispersed distributions, most likely due to feature dimensions irrelevant to the task at hand. On the task-specific manifold (b), instances from the same class tend to have more consistent representation, as reflected in more collapsed class clusters. This mirrors the well-known Neural Collapse phenomenon [18], with small intra-class variances and large inter-class distances, which leads to more robust classification decisions.

task. The number of classes in a task is often denoted by N, and the size of each class in the support set is denoted K. If K is a constant across all N classes in a task, the scenario is named N*-way* K*-shot*.

As you can see, the typical setup of N-way K-shot is not in line with the real world. It is more realistic to consider the setting of N-way Any-shot, where the size of each class varies across N classes in the support set. Slightly different from the types of class imbalance described in [10], [16], we pay further attention to the variance of class imbalance and assume the size of each class in the support set is drawn from a statistical distribution of large variance. We believe that such a more flexible and more challenging setting better reflects reality. Furthermore, it becomes easier to evaluate how the traditional FSL methods may fail and the effectiveness of alternative solutions.

B. Task-Level Manifold Learning

Figure 1 illustrates the proposed method. In a typical prototype-based FSL learning method (e.g., [4]), a pre-trained backbone model is used to map all support and query images into feature vectors. Then a prototype vector is generated in the same vector space to represent the corresponding class with labeled data in the support set. Finally, each query instance is classified into the class of its nearest prototype. To minimise the negative impact of nuisance features in high-dimensional embedding, we introduce an additional step to learn a task-level manifold. The manifold will remove task-irrelevant dimensions from the feature vector and hence improve the robustness of finding the nearest prototype.

1) Feature Extractor: Note that our proposed method can work with any pre-trained feature embedding, including generic backbone models and those pre-trained with meta-learning. Here we consider two pre-trained backbone networks: ResNet-12 from [4] and WideResNet28_10 from [13].

2) Why Reduce Dimension?: A pre-trained feature extractor converts an image into a high-dimension embedding. Why isn't it a good idea to use the same embedding mapping for all few-shot learning tasks?

First of all, high-dimension, low sample size data is notoriously difficult to work with. The "curse of dimensionality" tells us iid random vectors in R^n lie near the surface of the sphere S^{n-1} and the pair-wise distances (or angles) are nearly the same (see for example [3]). In other words, high-dimensional vectors do not form natural clusters. With a small number, say K, of random vectors, they tend to scatter and each occupies one vertex of a $(K-1)$-simplex [7].

Apart from the above generic difficulty, another challenge is specific to few-shot learning. A high-dimensional embedding vector from a pre-trained image model often contains features irrelevant to the N-way classification task at hand. This is because a deep learning model trained for a computer vision task needs to recognize different objects in an image. At the same time, not all objects are relevant for a few-shot learning task. As an example, in Figure 1, an image shows a person holding a dog. The person is a *nuisance feature* that is irrelevant to the target class—"dog" here. Its presence makes it difficult to differentiate the "crab" class, whose labelled data in the support set by coincidence also has a person holding something—a crab here.

Through learning a low-dimensional manifold, we can use topological structures from by within-class examples to factor out the nuisance features, eliminating their impact to the classifier.

3) How to Learn Low-Dimensional Manifold: Although numerous manifold learning algorithms exist, including Isomap [26], Locally Linear Embedding [6], [21], PCA [9], Spectral Embedding (SE) [2], and UMAP [14], they are unsupervised in nature. As we just saw in the Figure 1 example, it is important to learn the essential features in a class (e.g., fur colours in a dog image) and remove the nuisance features (a person holding a dog or a king crab).

There are two ways to achieve this. One is to modify the manifold learning algorithms. The other way is to ensure that

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 06:48:10 UTC from IEEE Xplore. Restrictions apply.

all support instances from the same class are clustered together in the produced manifold. We adopt the second approach due to its simplicity. To ensure that points from the same class in the support set are clustered, we propose the following data enhancement.

4) Inject Supervised Information: Dimensionality reduction techniques like UMAP map points from a high-dimensional space into many local neighbourhoods to capture the intrinsic global geometry. The construction of local neighbourhoods is based on k closest points in the ambient Euclidean distances. As we have seen before, the raw feature vectors in the ambient space may contain nuisance features, making two points from the same class artificially far away from each other. In Figure 2(a), we can see some green points are far away from other green points. In particular, there is one isolated green point that is closer to the grey cluster.

To ensure that points belonging to the same class are in the same neighbourhood so that essential features can be learnt, we propose to add an anchor point to connect all points from the same class in the support set. Let S_c denote the feature vectors in the support set of class c. The *Manifold Support Point* (MSP) of class c is defined as:

$$M_c = \frac{1}{|S_c|} \sum_{x \in S_c} x. \tag{1}$$

As M_c is the barycenter of class c in the support set, it is likely to be close to all data points of class c in the task and can help those points stay close within the manifold. (Note that in the 1-shot setting, the MSP is a replicate of the only feature vector; This can be regarded as balancing the Bayesian prior for all classes.)

In Figure 2, we can see that MSPs help capture task-specific geometry information at high dimensions and regenerate task-specific embedding. From Figure 2(a) to Figure 2(b), embeddings belonging to the same class are more compact while embeddings belonging to different classes are farther away after nuisance feature dimensions were removed. The accuracy of this task has also increased from 81% to 90% due to better embedding.

C. Denoiesd Prototype Classifier

At this stage, we use the learnt manifold to perform prototype-based classification over denoised features (projected features on the manifold). We use $\widetilde{x}$ to denote the projection of x on the low-dimensional manifold. We use the cleaner features in the support set to produce denoised prototypes of each class which is defined by:

$$P_c = \frac{1}{|S_c| + 1} \sum_{x \in S_c \cup \{M_c\}} \widetilde{x}. \tag{2}$$

Let z_i be the task-specific embedding (also projected) of a query sample x_i. Its probability of belonging to class c is estimated as

$$p(y = c \mid x_i) = \frac{\exp(\tau \cdot d(z_i, P_c))}{\sum_{c'} \exp(\tau \cdot d(z_i, P_c))}, \tag{3}$$

where $d(,)$ is either Euclidean distance of inverse or cosine distance and τ is a temperature parameter.

IV. Empirical Evaluation

A. Datasets and Implementation Details

The miniImageNet dataset [28] is a benchmark for few-shot learning. It contains 100 classes sampled from ILSVRC-2012 [22], and there are 600 images of size 84×84 in each class. According to the standard spilt [28], the 100 classes are divided into 64 training classes, 16 validation classes, and 20 test classes respectively.

The tieredImageNet dataset [20] is also a subset of ILSVRC-2012, but with a much large scale compared to miniImageNet. Therefore, it is an easier dataset. It contains 608 classes from 34 super-categories with image size 84×84, which are split into 20, 6, and 8 super-categories, resulting in 351, 97, and 160 classes as training, validation, and testing set respectively.

The CUB dataset [29] contains 11,788 images from 200 classes, which was proposed for fine-grained bird classification. We follow the standard spilt to divide 200 classes into 100, 50, and 50 for training, validation, and testing. The difficulty of this dataset lies in the minor difference between its classes.

For miniImageNet, tieredImageNet and CUB, we follow the standard setting to construct $10,000$ 5-way 5-shot and $10,000$ 5-way 1-shot tasks to test the effectiveness of our method. Moreover, to further test the performance of our method as the noise in the support set increase, we construct $10,000$ 5-way Any-shot tasks. In particular, the size of classes in the support set for each task was determined by a random sample from a negative binomial distribution with an expectation of 4 and a variance of 20. We assigned one labelled image to each class in advance, the expectation of labelled images in each class is equal to 5, which guarantees at least one labelled image in each class. The query set of each task consists of 75 images, with 15 for each class.

For convenience, we uniformly reduce the dimensions of feature vectors obtained from the pre-trained model to 32, so as to obtain relatively cleaner feature vectors for small tasks.

We took two types of feature extractors, ResNet-12 and WRN-28-10, trained by [4] and [13] respectively. Both of which achieve the state of the art and are widely used. We report the performance of different models using the mean accuracy as well as the standard error on $10,000$ randomly generated tasks.

B. Comparison of Different Models under Different Datasets and Settings

We consider three baseline models: Meta-Baseline [4], FEAT [30], and S2M2_R [13]. Meta-Baseline [4] based on prototype adopts two training stages, one for large-scale image classification and the other for meta-learning training. FEAT uses transformer-style multi-head self-attention [27] to rectify the prototypes of each class in a task. S2M2_R is an

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 06:48:10 UTC from IEEE Xplore. Restrictions apply.

TABLE I
PERFORMANCE OF META-BASELINE [4]. THE FEATURE EXTRACTOR OF META-BASELINE IS RESNET-12.

FSL Setting	Meta-Baseline	Denoised-Version
miniImageNet		
1-Shot	63.03± 0.21	**69.48 ± 0.25**
5-Shot	79.02± 0.15	**79.61 ± 0.17**
ND5-Shot	72.28 ± 0.19	**76.98 ± 0.19**
tieredImageNet		
1-Shot	68.80± 0.23	**74.53 ± 0.27**
5-Shot	83.40± 0.16	**84.42 ± 0.17**
ND5-Shot	74.84 ± 0.21	**82.22 ± 0.20**
CUB		
1-Shot	71.65± 0.22	**79.09 ± 0.24**
5-Shot	88.31± 0.12	**89.15 ± 0.13**
ND5-Shot	82.74 ± 0.18	**87.51 ± 0.16**

optimization-based model, whose classifier is an MLP fine-tuned with the support set for each task.

The results are shown in Table I, II and III. The second and third columns of these tables represent the performance change of the model before and after integrating our method. We can see that S2M2_R has the best performance on balanced data when the noise level in the support set is very low, but its performance drops significantly as the level of noise in the support set increases. FEAT is most resilient to noise in the support, however, it introduces additional models and computational complexity. As our method works directly on the feature space, it can be combined with any baseline method, without requiring changes to the original network. The last column of the three tables shows how our method works on imbalanced tasks. "ND5-Shot" means our task is sampled from a negative binomial distribution with a variance of 20 and an expected value of 5, which means the average number of labelled samples for each class in the support set is 5 under 10000 tasks. In all cases, our method can significantly improve the classification performance on imbalanced tasks.

C. Performance of Our Method under Different Levels of Noise

To verify the effectiveness of our method in handling noise, We can gradually increase the variance of the size of classes in the support set in an imbalanced scenario and view a balanced setting as a special distribution with a variance of 0. We use negative binomials to construct tasks with certain expectations equal to 5 which means the average labelled data of each class is 5. We vary the variance of each class to increase the level of noise in the support set. We use Meta-baseline [4] as the model and miniImageNet as the dataset to test the performance of our method against different levels of noise. We follow the experiment settings in [4], only making the support set in each task imbalanced. The result is shown in Figure 3. As the variance increases, tasks become more imbalanced. The accuracy of the model incorporating our method drops more slowly than the original model.

TABLE II
PERFORMANCE OF FEAT [30]. THE FEATURE EXTRACTOR FOR MINIIMAGENET AND TIEREDIMAGENET IS RESNET-12, WHILE FOR CUB IS CONVNET WHOSE SETTINGS ARE CONSISTENT WITH THE ORIGINAL PAPER.

FSL Setting	FEAT	Denoised-Version
miniImageNet		
1-Shot	66.97 ± 0.20	**72.59 ± 0.23**
5-Shot	**82.20 ± 0.14**	82.11 ± 0.15
ND5-Shot	77.73 ± 0.17	**80.01 ± 0.18**
tieredImageNet		
1-Shot	70.50± 0.22	**77.53 ± 0.25**
5-Shot	84.10± 0.16	**85.55 ± 0.16**
ND5-Shot	79.91 ± 0.19	**83.68 ± 0.19**
CUB		
1-Shot	68.76± 0.22	**71.93 ± 0.26**
5-Shot	**82.26± 0.15**	81.14 ± 0.17
ND5-Shot	77.91 ± 0.19	**78.69 ± 0.20**

TABLE III
PERFORMANCE OF S2M2_R [13]. THE FEATURE EXTRACTOR OF S2M2_R IS WIDERESNET28_10.

FSL Setting	S2M2_R	Denoised-Version
miniImageNet		
1-Shot	65.06 ± 0.20	**74.57 ± 0.24**
5-Shot	83.00 ± 0.13	**84.49 ± 0.14**
ND5-Shot	73.22 ± 0.20	**81.77 ± 0.17**
tieredImageNet		
1-Shot	67.40± 0.22	**78.82 ± 0.25**
5-Shot	85.00 ± 0.15	**86.98 ± 0.15**
ND5-Shot	72.06 ± 0.23	**84.44 ± 0.19**
CUB		
1-Shot	80.33± 0.20	**86.05 ± 0.21**
5-Shot	90.82± 0.11	**91.01 ± 0.12**
ND5-Shot	86.98 ± 0.15	**89.93 ± 0.15**

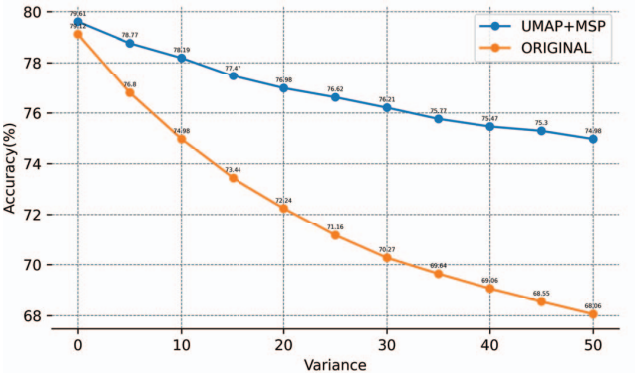


Fig. 3. Classification accuracy under increasing variance of the shot number distribution. When the variance increases, the task becomes more imbalanced. The baseline model's performance (orange line) drops quickly with more imbalance. In comparison, our method is more robust.

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 06:48:10 UTC from IEEE Xplore. Restrictions apply.

D. Ablation Study

We use Meta-baseline [4] to evaluate the contribution to performance improvement of the manifold-learning component and the MSP component. The results are shown in Table IV. It shows that MSP contributes to the performance of our method.

TABLE IV
5-WAY ANY-SHOT ACCURACY (%) WITH DIFFERENT MANIFOLD LEARNING METHODS. THE MEAN AND STANDARD ERROR WERE CALCULATED WITH 10000 RANDOM TASKS WITH THE SHOT NUMBERS SAMPLED FROM $NB(n = 1, p = 0.2)$.

Method	Dataset	
	miniImageNet	**tieredImageNet**
no manifold	72.24 ± 0.19	74.92 ± 0.21
Isomap [26]	73.03 ± 0.18	78.06 ± 0.19
LLE_H [6]	73.86 ± 0.20	79.03 ± 0.21
PCA [9]	74.82 ± 0.17	79.89 ± 0.18
SE [2]	75.53 ± 0.21	81.23 ± 0.21
UMAP [14]	76.19 ± 0.20	81.55 ± 0.21
UMAP+MSP	**76.98 ± 0.19**	**82.10 ± 0.20**

V. CONCLUSION

We propose a two-stage few-shot learning approach based on learning a stable task-level manifold spanned by essential features. The method is effective in countering class imbalance in FSL and can be plugged into a wide collection of existing few-shot learning methods. Our work opens several lines of future research. In particular, it would be interesting to investigate how the recently discovered phenomenon of neural collapse [18] can be explicitly incorporated into the learning of a task-level manifold. In future work, we intend to apply the proposed techniques to other areas, such as graph representation learning [23], [31] and recommender systems [11], [15].

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China under Grant Nos. 2021ZD0112501 and 2021ZD0112502; the National Natural Science Foundation of China under Grant Nos. U22A2098, 62172185, 62202200, and 62206105.

REFERENCES

[1] Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: International conference on database theory. pp. 420–434. Springer (2001)
[2] Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation **15**, 1373–1396 (2003)
[3] Blum, A., Hopcroft, J., Kannan, R.: Foundations of data science. Cambridge University Press (2020)
[4] Chen, Y., Liu, Z., Xu, H., Darrell, T., Wang, X.: Meta-baseline: Exploring simple meta-learning for few-shot learning. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV. pp. 9042–9051 (2021)
[5] Dhillon, G.S., Chaudhari, P., Ravichandran, A., Soatto, S.: A baseline for few-shot image classification. ArXiv **abs/1909.02729** (2020)
[6] Donoho, D.L., Grimes, C.: Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. Proceedings of the National Academy of Sciences of the United States of America **100**, 5591 – 5596 (2003)
[7] Hall, P., Marron, J.S., Neeman, A.: Geometric representation of high dimension, low sample size data. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **67**(3), 427–444 (2005)
[8] Hinneburg, A., Aggarwal, C.C., Keim, D.A.: What is the nearest neighbor in high dimensional spaces? In: 26th Internat. Conference on Very Large Databases. pp. 506–515 (2000)
[9] Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology **24**, 498–520 (1933)
[10] Lee, H.B., Lee, H., Na, D., Kim, S., Park, M., Yang, E., Hwang, S.J.: Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks (2019), arXiv preprint
[11] Li, A., Yang, B.: Gsirec: Learning with graph side information for recommendation. World Wide Web **24**, 1411–1437 (2021)
[12] Li, X., Sun, Q., Liu, Y., Zhou, Q., Zheng, S., Chua, T.S., Schiele, B.: Learning to self-train for semi-supervised few-shot classification. Advances in Neural Information Processing Systems **32** (2019)
[13] Mangla, P., Kumari, N., Sinha, A., Singh, M., Krishnamurthy, B., Balasubramanian, V.N.: Charting the right manifold: Manifold mixup for few- shot learning. In: Proceedings of the IEEE/CVF winter conference on ap- plications of computer vision. pp. 2218–2227 (2020)
[14] McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction, arXiv preprint
[15] Miao, H., Li, A., Yang, B.: Meta-path enhanced lightweight graph neural network for social recommendation. In: International Conference on Database Systems for Advanced Applications. pp. 134–149. Springer (2022)
[16] Ochal, M., Patacchiola, M., Storkey, A.J., Vazquez, J., Wang, S.: Few-shot learning with class imbalance. ArXiv **abs/2101.02523** (2021)
[17] Oreshkin, B., Rodríguez López, P., Lacoste, A.: Tadam: Task dependent adaptive metric for improved few-shot learning. Advances in neural information processing systems **31** (2018)
[18] Papyan, V., Han, X., Donoho, D.L.: Prevalence of neural collapse during the terminal phase of deep learning training. Proceedings of the National Academy of Sciences **117**(40), 24652–24663 (2020)
[19] Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. ICLR (2017)
[20] Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J.B., Larochelle, H., Zemel, R.S.: Meta-learning for semi-supervised fewshot classification. arxiv (2018), preprint
[21] Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. science **290**(5500), 2323–2326 (2000)
[22] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
[23] Shi, D., Li, A., Yang, B.: Counterfactual-guided and curiosity-driven multi-hop reasoning over knowledge graph. In: International Conference on Database Systems for Advanced Applications. pp. 171–179. Springer (2022)
[24] Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. Advances in neural information processing systems **30** (2017)
[25] Song, Y., Wang, T., Mondal, S.K., Sahoo, J.P.: A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. arXiv preprint arXiv:2205.06743 (2022)
[26] Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290 5500**, 2319–23 (2000)
[27] Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. ArXiv **abs/1706.03762** (2017)
[28] Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In Advances in neural information processing systems **1**(2), 3630–3638 (2016)
[29] Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-ucsd birds 200 (2010)
[30] Ye, H.J., Hu, H., Zhan, D.C., Sha, F.: Few-shot learning via embedding adaptation with set-to-set functions. in Proceedings of the IEEE/CVF Con- ference on Computer Vision and Pattern Recognition pp pp. 8808–8817 (2020)

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 06:48:10 UTC from IEEE Xplore. Restrictions apply.

[31] Zang, X., Yang, B., Liu, X., Li, A.: Dnea: Dynamic network embedding method for anomaly detection. In: International Conference on Knowledge Science, Engineering and Management. pp. 236–248. Springer (2021)

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 06:48:10 UTC from IEEE Xplore. Restrictions apply.