RankDNN: Learning to Rank for Few-Shot Learning

Qianyu Guo[1,2], Gong Haotong[1], Xujun Wei[1,3], Yanwei Fu[2], Yizhou Yu[4], Wenqiang Zhang[2,3], Weifeng Ge[1,2*]

[1]Nebula AI Group, School of Computer Science, Fudan University,Shanghai,China
[2]Shanghai Key Laboratory of Intelligent Information Processing,Shanghai,China
[3]Academy for Engineering & Technology, Fudan University,Shanghai,China
[4]Department of Computer Science, The University of Hong Kong,Hong Kong,China
wfge@fudan.edu.cn

Abstract

This paper introduces a new few-shot learning pipeline that casts relevance ranking for image retrieval as binary ranking relation classification. In comparison to image classification, ranking relation classification is sample efficient and domain agnostic. Besides, it provides a new perspective on few-shot learning and is complementary to state-of-the-art methods. The core component of our deep neural network is a simple MLP, which takes as input an image triplet encoded as the difference between two vector-Kronecker products, and outputs a binary relevance ranking order. The proposed RankMLP can be built on top of any state-of-the-art feature extractors, and our entire deep neural network is called the ranking deep neural network, or RankDNN. Meanwhile, RankDNN can be flexibly fused with other post-processing methods. During the meta test, RankDNN ranks support images according to their similarity with the query samples, and each query sample is assigned the class label of its nearest neighbor. Experiments demonstrate that RankDNN can effectively improve the performance of its baselines based on a variety of backbones and it outperforms previous state-of-the-art algorithms on multiple few-shot learning benchmarks, including miniImageNet, tieredImageNet, Caltech-UCSD Birds, and CIFAR-FS. Furthermore, experiments on the cross-domain challenge demonstrate the superior transferability of RankDNN.The code is available at: https://github.com/guoqianyu-alberta/RankDNN.

Introduction

Contrary to the normal practice of using a large amount of labeled data (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016; Zhou et al. 2022), few-shot learning (Lu et al. 2020; Afrasiyabi, Lalonde, and Gagne 2021) refers to learning new concepts from a very small amount of data by leveraging the learning "skills" gained from many similar learning tasks. State-of-the-art methods (Hong et al. 2021; Tang et al. 2021; Rizve et al. 2021) attempt to learn meta knowledge that can be transferred to new tasks. Although much progress has been made, recently proposed methods still bear the risk of unsatisfactory generalization performance on new tasks. The reason is two-fold. First, the meta knowledge they learn may have limited transferability and may not be well applicable to certain new tasks. Second, there is simply too little data available for new tasks and overfitting is very hard to avoid. Modern deep neural networks are so flexible and overparameterized that they can even perfectly fit image data with random labels, as stated in (Zhang et al. 2021; Arpit et al. 2017; Ge and Yu 2017). Overfitting creates a large gap between training and testing performance.

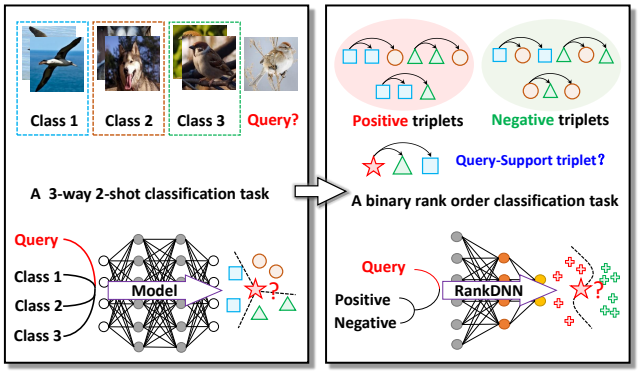

Figure 1: Ranking deep neural network (RankDNN) establishes a learning framework that can cast a N-way-K-shot learning task into a binary relevance ranking problem, which is sample efficient and domain agnostic.

Inspired by relevance ranking in information retrieval (Shashua, Levin et al. 2003; Joachims 2002), we consider binary relevance ranking for few-shot learning, as shown in Fig 1. Given a query image and two support images in a given order, binary relevance ranking ranks the relevance of these support images concerning the query image. There are only two possible outcomes of this ranking problem. That is, the first support image is more relevant than the second one or vice versa. Relevance ranking among images focuses on the similarity of images, especially the similarity of foreground objects, but not the specific categories of images or foreground objects. Thus a relevance ranking model should be able to grasp the meta knowledge about the assessment of image or object similarity regardless of their specific contents. Such a capability endows the learned model a strong transferability to a wide range of new tasks where specific image contents may vary. Meanwhile, as we need three images for each binary relevance ranking instance, the

*Corresponding author

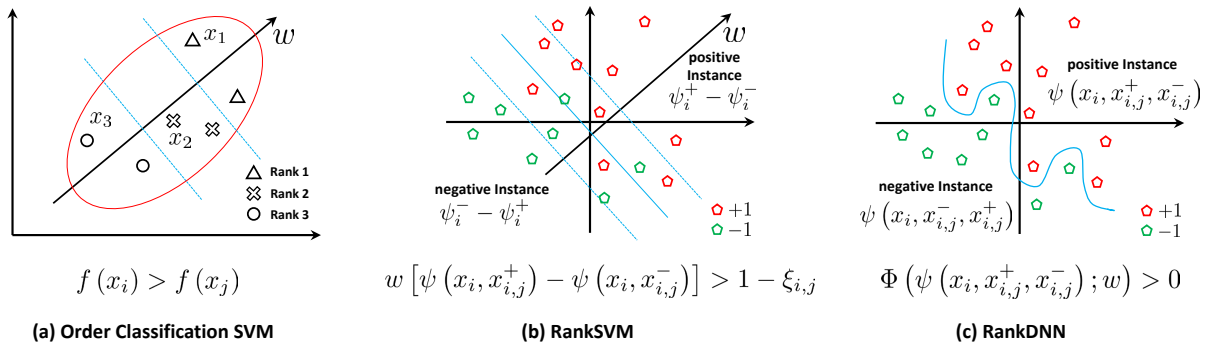

Figure 2: Comparison of ranking learning frameworks. (a) Pointwise ranking algorithms, such as McRank (Li, Wu, and Burges 2007; Shashua, Levin et al. 2003) and OC SVM, cast the ranking learning problem as a regression or classification task on single objects. (b) Pairwise ranking algorithms, such as RankSVM (Joachims 2002; Prosser et al. 2010), models binary relevance ranking within object pairs. (c) The proposed RankDNN generalizes binary relevance ranking in RankSVM into binary triplet classification with deep neural networks.

number of training samples for ranking is greatly expanded by constructing triplets, which significantly alleviates training data shortage in few-shot learning.

We develop a ranking learning framework for few-shot image classification, which is formulated as a query-support relevance ranking problem. We generalize the SVM based ranking learning idea in OC SVM (Shashua, Levin et al. 2003) and RankSVM (Joachims 2002) to deep neural networks, and call the proposed framework RankDNN. Meanwhile, we provide a new perspective on image classification by converting a multiclass classification task into a binary classification task.

To implement binary relevance ranking of two support images to a query image, there are two key issues: The first issue is how to simultaneously encode the semantic features of an image triplet as well as the roles and order of the three images in the triplet; the second issue is which machine learning algorithm should be used to produce the binary ranking result. Encoding an image triplet while preserving all necessary information is very challenging. We use the query image in the triplet to form two query-support pairs with the two support images, and encode each query-support pair using the outer product, also called vector-Kronecker product, of their feature vectors. The entire image triplet is finally represented using the difference between these two vector outer products. Our intuition behind this encoding scheme is that the vector-Kronecker product of two feature vectors can not only preserve all information in the original feature vectors but also model the correlation between any two dimensions of these features. In addition, the sign of the final difference encodes the order of the two support images in the triplet.

In summary, this paper has the following contributions:

• We introduce a new ranking learning framework for few-shot learning called ranking deep neural network (or RankDNN), which decomposes a few-shot image classification task into multiple binary relevance ranking problems. By constructing image triplets for such binary ranking problems, our framework generates a large amount of training data across different image classes.

• We propose a novel image triplet encoding scheme based on vector-Kronecker products. It preserves all necessary information in an image triplet and can also model the correlation between any two feature dimensions respectively of the query image and one of the support images.

• Experiments demonstrate that our RankDNN outperforms its baselines in many different scenarios and achieves state-of-the-art performance on multiple few-shot learning benchmarks. Besides, the fusion experiments, cross-domain challenges and different backbones experiments prove the generalization, flexibility and robustness of RankDNN.

Related Work

Few-Shot Learning. Various few-shot learning algorithms have been proposed, such as Matching Net (Vinyals et al. 2016), Relation Net (Sung et al. 2018), MAML (Finn, Abbeel, and Levine 2017), and TAML (Jamal and Qi 2019). However, according to (Chen et al. 2019a; Gidaris and Komodakis 2018; Qiao et al. 2018), since there are two few annotated samples, state-of-the-art algorithms (Mangla et al. 2020; Rizve et al. 2021) focus on solving the overfitting problem by self-supervised learning. There are also many other methods that try to learn meta knowledge with the strong generalization ability (Hong et al. 2021; Tang et al. 2021).In this paper, we provide a new perspective to solve the few-shot learning problem by converting it into a relevance ranking task. Then our learning task is built upon image triplets and becomes domain agnostic.
Learning to Rank. Applying existing and effective machine learning methods to rank is called learning to rank, such as RankSVM(Joachims 2002) and RankNet(Burges et al. 2005). Learning to rank is not only widely used in major tasks of natural language processing(Masuda 2003; Briakou and Carpuat 2020; Zhang and van Genabith 2020), but also has attracted the attention of metric learning in recent years(Liu et al. 2021; Cakir et al. 2019; Wang et al.

2019). (Wang et al. 2019) proposed ranked-based list loss with structured information of multiple samples to improve the training efficiency of metric learning. Although DeepRank (Pang et al. 2017) utilizes deep neural networks to conduct similarity ranking, its learning objective is a pairwise contrastive loss which is widely used in metric learning. While in RankDNN, we convert the rank problem into a binary classification problem and solve it with gradient-based methods.

A Ranking Learning Framework for Few-Shot Image Classification

The general regime of an *N*-way *K*-shot classification task $\mathcal{T}$: with N previously unseen image categories, each of which contains K samples, the task aims to build a classifier to classify a set of query images into these N categories. The dataset for task $\mathcal{T}$ is $\mathcal{D}_{\mathcal{T}} = \mathcal{S} \cup \mathcal{Q}$ where $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N\times K}$ is the support set, $\mathcal{Q} = \{(\boldsymbol{x}_i, y_i)\}_{i=N\times K+1}^{N\times K+T}$ is the query set, T is the number of query samples. $\boldsymbol{x}_i$ and $y_i (\in \{C_1, ..., C_N\} = \mathcal{C}_{\mathcal{T}} \subset \mathcal{C})$ are respectively the i-the image sample and its label. State-of-the-art methods often formalize few-shot image classification as a metric learning (Zhang et al. 2020; Mangla et al. 2020) or multiclass classification task (Rizve et al. 2021; Chen et al. 2021; Yang, Liu, and Xu 2021). However, in this paper, we cast few-shot classification as an image retrieval task. Given a query image $\boldsymbol{x}_q$ and a support image collection $\mathcal{S}$, a retrieval system should return a complete ranked list $r^*(\boldsymbol{x}_q)$, that sorts the support images in $\mathcal{S}$ according to their relevance to the query as follows.

$$r^*(\boldsymbol{x}_q) : \boldsymbol{x}_{q_1} \succ \boldsymbol{x}_{q_2} \succ \cdots \succ \boldsymbol{x}_{q_{N\times K}}, \tag{1}$$

where $\boldsymbol{x}_{q_k} \in \mathcal{S}$, and $\boldsymbol{x}_{q_i} \succ \boldsymbol{x}_{q_j}$ means $\boldsymbol{x}_{q_i}$ is more relevant to $\boldsymbol{x}_q$ than $\boldsymbol{x}_{q_j}$. The formula in (1) indicates how an image retrieval system works (Ge 2018; Wang et al. 2020; Levi et al. 2021). Most metric learning based methods (Zhang et al. 2020; Snell, Swersky, and Zemel 2017) for few-shot learning aim to find a unified embedding space where every query sample is assigned the class label of its nearest support image.

Inspired by the approach in (Joachims 2002; Prosser et al. 2010), we decompose the complete relevance ranking relation among multiple variables into several binary relevance ranking relations over $\mathcal{S} \times \mathcal{S}$. Thus, we have

$$\forall \left(\boldsymbol{x}_{q_i}, \boldsymbol{x}_{q_j}\right) \in r^*\left(\boldsymbol{x}_q\right)(i < j) : \boldsymbol{x}_{q_i} \succ \boldsymbol{x}_{q_j}, \boldsymbol{x}_{q_j} \prec \boldsymbol{x}_{q_i}. \tag{2}$$

There are $\mathcal{O}\left(|\mathcal{S}|^2\right)$ such relevance ranking pairs. The following 5-way-1-shot task is given as an example:

$$\begin{aligned}
&\boldsymbol{x}_{q_1} \succ \boldsymbol{x}_{q_2}, \boldsymbol{x}_{q_1} \succ \boldsymbol{x}_{q_3}, \boldsymbol{x}_{q_1} \succ \boldsymbol{x}_{q_4}, \boldsymbol{x}_{q_1} \succ \boldsymbol{x}_{q_5};\\
&\boldsymbol{x}_{q_2} \prec \boldsymbol{x}_{q_1}, \boldsymbol{x}_{q_2} \succ \boldsymbol{x}_{q_3}, \boldsymbol{x}_{q_2} \succ \boldsymbol{x}_{q_4}, \boldsymbol{x}_{q_2} \succ \boldsymbol{x}_{q_5};\\
&\boldsymbol{x}_{q_3} \prec \boldsymbol{x}_{q_1}, \boldsymbol{x}_{q_3} \prec \boldsymbol{x}_{q_2}, \boldsymbol{x}_{q_3} \succ \boldsymbol{x}_{q_4}, \boldsymbol{x}_{q_2} \succ \boldsymbol{x}_{q_5};\\
&\boldsymbol{x}_{q_4} \prec \boldsymbol{x}_{q_1}, \boldsymbol{x}_{q_4} \prec \boldsymbol{x}_{q_2}, \boldsymbol{x}_{q_4} \prec \boldsymbol{x}_{q_3}, \boldsymbol{x}_{q_4} \succ \boldsymbol{x}_{q_5};\\
&\boldsymbol{x}_{q_5} \prec \boldsymbol{x}_{q_1}, \boldsymbol{x}_{q_5} \prec \boldsymbol{x}_{q_2}, \boldsymbol{x}_{q_5} \prec \boldsymbol{x}_{q_3}, \boldsymbol{x}_{q_5} \prec \boldsymbol{x}_{q_4}.
\end{aligned} \tag{3}$$

The query image $\boldsymbol{x}_q$ is then assigned the class label of the support sample ranked first. By exploiting such binary relations, we can obtain an extensive number of training samples in few-shot learning, avoid the necessity to obtain a complete ranked list, and tolerate minor inconsistencies in binary ranking results.

According to Eq. (2), we need to formulate the binary relevance ranking relation between any two support images $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for a query sample $\boldsymbol{x}_q$. There are two key elements in describing such a relation: First, an informative encoding scheme $\psi(\cdot;\theta)$ (θ denotes the parameters of ψ) to encode the semantic features as well as the detailed component-wise correlations between a query image and a support image; second, a scoring function Φ to evaluate the overall similarity or relevance between a query image and a support image. Then a binary relevance ranking relation can be concretely formulated as follows:

$$\boldsymbol{x}_i \succ_{r^*(q)} \boldsymbol{x}_j \iff \Phi\left(\psi\left(x_q, x_i\right); w\right) > \Phi\left(\psi\left(x_q, x_j\right); w\right), \tag{4}$$

where w denotes the learnable parameters of the scoring function. RankSVM (Joachims 2002) adopts the class of linear scoring functions, and the binary ranking learning problem becomes finding the weight vector that fulfills the most inequalities in Eq. 2, which can be formulated as follows.

$$\begin{aligned}
&\min_{w,\vec{\xi}} \frac{1}{2}\|w\|^2 + C\sum_{i,j,k}\xi_{i,j,k}\\
&\text{s.t. } \forall \boldsymbol{x}_i \succ_{r^*(q)} \boldsymbol{x}_j : w\left[\psi\left(x_q, x_i\right) - \psi\left(x_q, x_j\right)\right] > 1 - \xi_{i,j,q}\\
&\quad \forall i \forall j \forall q : \xi_{i,j,q} > 0,
\end{aligned} \tag{5}$$

where C is a constant, $\xi_{i,j,q}$ is a slack variable. RankSVM (Joachims 2002) can be extended to use nonlinear scoring functions via kernel methods.

In Fig 2, the few-shot classification problem can now be cast as a binary classification task and solved by RankSVM. To step further, we wish to obtain a more general formulation about the ranking learning problem. Given a query sample x_q, there are two kinds of ranking relations: if $\Phi\left(\psi\left(x_q, x_i\right); w\right) > \Phi\left(\psi\left(x_q, x_j\right); w\right)$, $\langle x_q, x_i, x_j\rangle$ is identified as a positive triplet sample; otherwise, $\Phi\left(\psi\left(x_q, x_i\right); w\right) \leq \Phi\left(\psi\left(x_q, x_j\right); w\right)$, and $\langle x_q, x_i, x_j\rangle$ is a negative triplet sample. Then we can define a unified triplet encoding scheme, which not only encodes the feature vectors of the query and support samples in the triplet, but also their roles and order in the triplet as well as their interactions. In a training triplet, one of the support samples should come from the same class as the query sample while the other support sample should come from a different class. We can automatically generate the ground-truth binary class label of a triplet as follows,

$$y\left(\langle x_q, x_i, x_j\rangle\right) = \begin{cases} +1, \text{if } x_i \succ_{r^*(q)} x_j; \\ -1, \text{if } x_i \prec_{r^*(q)} x_j. \end{cases} \tag{6}$$

In this paper, we generalize the linear/nonlinear scoring function Φ in RankSVM to an MLP based binary classifier, called RankMLP. RankMLP is trained using the following overall loss function,

$$\mathcal{L}_{rank} = \frac{1}{N_t}\sum_{q,i,j}\mathcal{L}_{bce}\left(\Phi\left(\psi\left(\langle x_q, x_i, x_j\rangle; \theta\right); w\right), y_{q,i,j}\right), \tag{7}$$

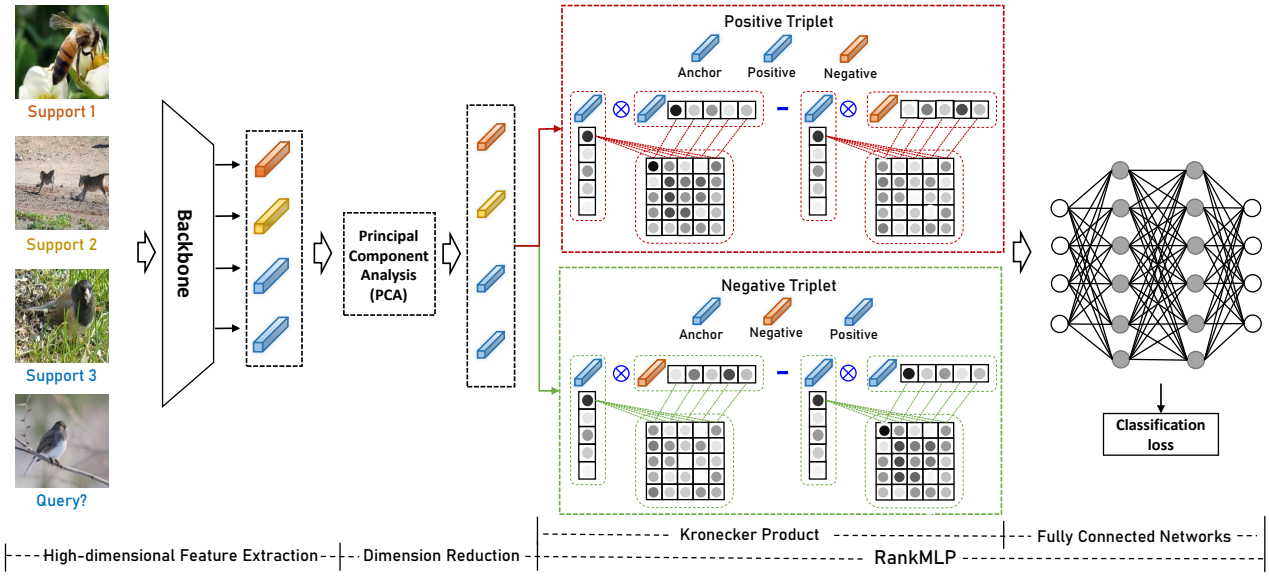

Figure 3: The ranking deep neural network is composed of a high-dimensional feature extractor, a dimension reduction part and a ranking multilayer preceptron. The parameters feature extractor are frozen all the time and the RankMLP are trained consecutively in different stages.

where N_t is the number of triplets in a mini-batch, $\mathcal{L}_{bce}$ is the binary cross-entropy loss, and $y_{q,i,j}$ is the ground-truth binary class label of a triplet.

Discussion. According to Eqs. (6) and (7), we solve the ranking learning problems with deep neural networks, as in information retrieval (Palangi et al. 2016; Pang et al. 2017) and metric learning (Cakir et al. 2019; Liu et al. 2021). The main difference is that RankMLP focuses on binary ranking relation classification but not the distance constraints in metric learning. Given an N-way-K-shot task in the training set of a few-shot learning problem, we can construct $(N-1)K(K-1)$ training triplets for every query sample, which significantly alleviates the data shortage.

The Ranking Deep Neural Network

Overview

The ranking deep neural network (RankDNN) takes in an image triplet and outputs whether this triplet is a positive sample or a negative sample. Although some existing deep metric learning methods (Cakir et al. 2019; Liu et al. 2021) already use the ranking loss, a key difference here is that inter-class variations are much more challenging in few-shot learning than those in image retrieval. In our experiments, directly training a deep neural network from end to end to classify triplets would lead to gradient explosion. In Fig 3, the RankDNN pipeline contains two parts: a frozen high-dimensional feature extractor $f(\cdot, \boldsymbol{\theta}_e)$ and a ranking deep neural network $\Phi(\cdot, \boldsymbol{\theta})$.

Algorithm 1 outlines the training phase of RankDNN for few-shot learning. Given an image triplet $\langle x_q, x_i, x_j \rangle$, RankDNN uses state-of-the-art feature extractors, such as S2M2 (Mangla et al. 2020) and IE (Rizve et al. 2021), to extract discriminative features $f(x_q; \boldsymbol{\theta}_e), f(x_i; \boldsymbol{\theta}_e), f(x_j; \boldsymbol{\theta}_e) \in \mathbb{R}^{640}$. For simplicity, we write $f(x_q; \boldsymbol{\theta}_e)$ as f_q. To generate subsequent features for triplet classification, we use PCA (Yang et al. 2004) to reduce the image feature dimension. During the training stage, we freeze both the feature extractor and the PCA transform, and train RankMLP with the loss function $\mathcal{L}_{rank}$ defined in Eq. (7).

Triplet Encoding Scheme

Encoding triplets of extracted features plays a precursory role for RankMLP. The Kronecker product is a matrix algebra operation that produces structured descriptions for modeling complex systems. If $A \in \mathbb{R}^{n_a \times m_a}$ and $B \in \mathbb{R}^{n_a \times m_a}$, their Kronecker product is denoted by $A \otimes B$ and defined as follows:

$$A \otimes B \equiv \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{n1}\mathbf{B} & \cdots & a_{nn}\mathbf{B} \end{bmatrix}.$$

When both matrices degenerate to vectors $x \in \mathbb{R}^{d \times 1}$ and $y \in \mathbb{R}^{1\ timesd}$, their Kronecker product is actually the same as their outer product; and is called vector-Kronecker product in this paper. It is formulated in the following equation:

$$\mathbf{x} \otimes \mathbf{y} \equiv \begin{bmatrix} x_1 y_1 & \cdots & x_1 y_n \\ \vdots & \ddots & \vdots \\ x_n y_1 & \cdots & x_n y_n \end{bmatrix}. \quad (8)$$

Algorithm 1: Meta-Training of RankDNN for Few-shot Learning

Input: Training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{k=1}^{N}$. Network $f(\cdot, \boldsymbol{\theta}_e)$ initialized with a state-of-the-art pretrained model. The ranking neural network $\Phi(\cdot, \boldsymbol{\theta})$ initialized with randomly noises.

Output: The learnable parameters $\boldsymbol{\theta}$ of the neural networks $\Phi(\cdot, \boldsymbol{\theta})$.

// conduct PCA transform for the dimension reduction

Extract features of all training images and learn a principal component analysis transform (PCA) T;

// train the ranking neural network Φ and freeze the feature extractor f_e all the time

while *not converge* **do**

  $t \leftarrow t + 1$;

  Sample anchors randomly and their neighborhoods according to the method in HTL (Ge 2018);

  Extract features with $f_e(\cdot, \boldsymbol{\theta}_e)$, and freeze it;

  Reduce the feature dimension with PCA;

  Construct the triplet ranking description with Eq. 9 and pass it through the RankMLP $\Phi(\cdot, \boldsymbol{\theta})$;

  Compute the binary cross entropy loss in a mini-batch $\mathcal{L}_{rank}$ with Eq. 7;

  Backpropagate the gradients produced at the loss layer and update the learnable parameters $\boldsymbol{\theta}$.

end

In the literature (De Launey and Seberry 1994; Langville and Stewart 2004; Weichsel 1962), the Kronecker product of graphs is one of the usual names of the categorical product of graphs, also called the tensor product. This product of graphs was studied by various authors (Azevedo et al. 2020; Weichsel 1962; Mamut and Vumar 2008), who proved that Kronecker product can encode the connectivity and relationship between graphs.

Given a triplet of feature descriptors $\langle f_q, f_i, f_j \rangle$, state-of-the-art methods (Chen et al. 2021; Rizve et al. 2021) calculate cosine similarities, $f_q f_i / (\|f_q\| \, \|f_i\|)$ and $f_q f_j / (\|f_q\| \, \|f_j\|)$, and further rank the similarities. In fact, this cosine similarity focuses on the correlation between corresponding entries in the two feature vectors, and may not be able to capture all types of correlations between two vectors. Therefore, we design the following encoding scheme for feature triplets:

$$\psi(\langle x_q, x_i, x_j \rangle; \theta) = f_q \otimes f_i - f_q \otimes f_j. \tag{9}$$

Note that popular backbones in few-shot learning include ResNet12 (Rizve et al. 2021),ResNe-18 (Ye et al. 2020) and WRN-28-10 (Gidaris et al. 2019; Zagoruyko and Komodakis 2016), both of which output 640-dimensional features. Then the result $\psi(\langle x_q, x_i, x_j \rangle; \theta) \in \mathbb{R}^{d \times d}$ in Eq. (9) would have 409600 dimensions, which is too high. Therefore, we reduce the dimension of each extracted feature vector to 80 or 128 with PCA (Yang et al. 2004).

Experiments

Experimental Details

Datasets and Performance Metrics. We use four popular benchmark datasets in our experiments: *mini*ImageNet(Vinyals et al. 2016), *tiered*ImageNet (Ren et al. 2018), Caltech-UCSD Birds-200-2011 (CUB)(Chen et al. 2019b), and CIFAR-FS(Bertinetto et al. 2018). All datasets follow a standard division and all images are resized to predefined resslutions following standard settings. We evaluate the performance under standard 5-way-1-shot and 5-way-5-shot settings.

Implementation Details. For the feature extractor, we consider three state-of-the-art backbones, S2M2 (Mangla et al. 2020), IE (Rizve et al. 2021) and FEAT (Ye et al. 2020). The number of neurons in different layers of RankMLP is $[6400, 1024, 512, 256, 1]$ for S2M2 and FEAT, and $[16384, 1024, 512, 256, 1]$ for IE. We set the weight decay to 10^{-6} and the momenta to 0.9. The learning rate is fixed at 0.0005 for both networks. Note that during the meta test stage, RankDNN does not need to finetune on 1-shot, but on 5-shot, RankMLP needs to be finetuned with the support set to get good performance, where we sample 100 triplets randomly in each mini-batch, and the parameters of RankMLP is updated in 100 iterations. The learning rate is set to 0.01. RankDNN is optimized with SGD.

Ranking Voting. For an N-way-K-shot task, there are a total of NK support images. After feature extraction, we average the K features to one per class, so we get N support features whether on the 1-shot or n-shot setting. Each query feature is treated as an anchor, and every two support out of the N average features can be used to form a triplet with the query. Thus, $N \times (N-1)$ valid triplets can be constructed for each query. If a triplet is predicted positive by our RankDNN, the first support image in the triplet is given one positive point. We define the ranking score of a support sample to be the total number of positive points received by the support sample. The class label of the query image is the same as the class label of the support sample with the highest ranking score.

Comparison Experiments with the State-of-the-arts

As shown in Table 1, Table 2 and Table 3, RankDNN outperforms its baselines obviously on all benchmarks. It demonstrates that the relevance ranking is helpful to distinguish similar images. Also, RankDNN surpasses all previous state-of-the-art algorithms. On the *mini*ImageNet,when compared with the previous state-of-the-art methods(IE on ResNet-12, FEAT on ResNet-18 and S2M2 on WRN), RankDNN achieves 1.44 %, 0.88%, 0.88%,5.47%,0.49%,1.74% and 1.61% improvements in 5-way-1-shot and 5-way-5-shot accuracies respectively. We attribute this to the strong generalization ability of RankDNN, which can be used to enhance various few-shot learning algorithms. On *tiered*ImageNet, RankDNN shows obvious performance improvements over the S2M2, FEAT and IE baselines under both 5-way-1-shot and 5-way-5-shot settings. RankDNN with the ResNet-12 backbone surpasses previously best performing TAS by 1.06% and 1.86% un-

Method	Backbone	*mini*ImagNet		*tiered*Imagenet	
		1-shot	5-shot	1-shot	5-shot
DMF CVPR21	ResNet-12	$67.76_{\pm 0.46}$	$82.71_{\pm 0.31}$	$71.89_{\pm 0.52}$	$85.96_{\pm 0.35}$
IE CVPR21	ResNet-12	$67.28_{\pm 0.80}$	$84.78_{\pm 0.52}$	$72.21_{\pm 0.90}$	$87.08_{\pm 0.58}$
DMN4 AAAI22	ResNet-12	66.58	83.52	72.10	85.72
HGNN AAAI22	ResNet-12	$67.02_{\pm 0.20}$	$83.00_{\pm 0.13}$	$72.05_{\pm 0.23}$	$86.49_{\pm 0.15}$
APP2S AAAI22	ResNet-12	$66.25_{\pm 0.20}$	$83.42_{\pm 0.15}$	$72.00_{\pm 0.22}$	$86.23_{\pm 0.15}$
UNICORN ICLR22	ResNet-12	$65.17_{\pm 0.20}$	$84.30_{\pm 0.13}$	$69.24_{\pm 0.20}$	$86.06_{\pm 0.16}$
TAS ICLR22	ResNet-12	$65.68_{\pm 0.45}$	$83.92_{\pm 0.55}$	$72.81_{\pm 0.48}$	$86.06_{\pm 0.16}$
RankDNN(ours)	ResNet-12	$\mathbf{68.72_{\pm 0.15}}$	$\mathbf{85.66_{\pm 0.27}}$	$\mathbf{73.87_{\pm 0.40}}$	$\mathbf{87.92_{\pm 0.15}}$
$\triangle-$encode NIPS18	ResNet-18	59.90	69.70	—	—
SS ECCV20	ResNet-18	—	76.60	—	78.90
FEAT CVPR20	ResNet-18	55.15	66.78	62.40	77.81
Hyperbolic CVPR20	ResNet-18	$57.05_{\pm 0.20}$	$76.84_{\pm 0.14}$	$66.20_{\pm 0.28}$	$76.50_{\pm 0.40}$
RankDNN(ours)	ResNet-18	$\mathbf{62.52_{\pm 0.25}}$	$\mathbf{77.33_{\pm 0.45}}$	$\mathbf{66.97_{\pm 0.18}}$	$\mathbf{81.07_{\pm 0.23}}$
S2M2 WACV20	WRN	$64.93_{\pm 0.18}$	$83.18_{\pm 0.11}$	$73.71_{\pm 0.22}$	$88.59_{\pm 0.14}$
FEAT CVPR20	WRN	65.10	81.11	70.41	84.38
PSST CVPR21	WRN	$64.16_{\pm 0.44}$	$80.64_{\pm 0.32}$	—	—
RankDNN(ours)	WRN	$\mathbf{66.67_{\pm 0.15}}$	$\mathbf{84.79_{\pm 0.11}}$	$\mathbf{74.00_{\pm 0.15}}$	$\mathbf{88.80_{\pm 0.25}}$

Table 1: Comparison of 5-way few-shot accuracies on *mini*ImageNet and *tiered*ImageNet with ResNet backbones.

der both 5-way few-shot settings respectively. RankDNN with the ResNet-18 backbone achieves new state-of-the-art performance (66.97% and 81.07%) under both settings. RankDNN with the WRN backbone also surpasses previously best performing S2M2. This proves the effectiveness and robustness of our RankDNN.

Method	Backbone	CUB	
		1-shot	5-shot
IE CVPR21	ResNet-12	80.92	90.13
RENet ICCV21	ResNet-12	79.49	91.11
HGNN ICCV21	ResNet-12	78.58	90.02
CCG+HAN ICCV21	ResNet-12	74.66	88.37
APP2S AAAI22	ResNet-12	77.64	90.43
RankDNN(ours)	ResNet-12	**82.93**	**91.47**
S2M2 WACV20	WRN	80.68	90.85
DC ICLR21	WRN	79.56	90.67
RankDNN(ours)	WRN	**81.78**	**91.12**

Table 2: Comparison of 5-way few-shot accuracies on CUB.

On the CUB dataset, RankDNN with both S2M2 and IE backbones achieves impressive performance under both 5-way few-shot settings (Table 2). On the CIFAR-FS, where images have a very low resolution, RankDNN with the IE backbone surpasses its baseline, which is also the recent best, by 1.14% and 0.90% under 5-way-1-shot and 5-way-5-shot settings, respectively (Table 3). These results indicate RankDNN are complementary to state-of-the-art methods, and can generalize well on previously unseen visual concepts.

Fusion Experiments with the State-of-the-arts

We reproduce the state-of-the-arts methods, including ProtoNet (Snell, Swersky, and Zemel 2017), E^3BM (Liu, Schiele, and Sun 2020), DeepEMD (Zhang et al. 2020), Distill (Tian et al. 2020), FRN (Wertheimer, Tang, and Hariharan 2021), RENet (Kang et al. 2021)and DC (Yang, Liu, and Xu 2021)), and fusion them with RankDNN. As shown in Table 4, fusion with RankDNN improves the performance of all original methods, both on 1-shot and 5-shot. In particular, for ProtoNet, RankDNN improves 3.55% and 1.46% on *mini*ImageNet and 2.01% and 1.09% on *tiered*ImageNet. These results indicate that RankDNN can be flexibly adapted to different baselines, and various methods can benefit from RankDNN.

Method	Backbone	CIFAR-FS	
		1-shot	5-shot
IE CVPR21	ResNet-12	77.87	89.74
PAL ICCV21	ResNet-12	77.10	88.00
TPMN ICCV21	ResNet-12	75.50	87.20
CCG+HAN ICCV21	ResNet-12	73.00	85.80
APP2S AAAI22	ResNet-12	73.12	85.69
LH AAAI22	ResNet-12	78.00	90.50
RankDNN(ours)	ResNet-12	**78.93**	**90.64**

Table 3: Comparison of 5-way few-shot on CIFAR-FS.

Discriminative Ability of Feature Descriptors in RankDNN

We claim that features of the rank-triples formed by the Kronecker are more discriminative and stable than the original deep features. To validate, for a 5-way-100-shot task, we visualize features extracted by the backbone and the second last layer of RankMLP with t-SNE in Fig 4. It can be found that backbone features will result some ambiguity around the classification boundaries. However, the features generated by RankMLP have a stronger discriminative ability in distinguishing positive and negative ranked samples.

Method	Backbone	*mini*ImagNet		*tiered*Imagenet	
		1-shot	5-shot	1-shot	5-shot
ProtoNet (Snell, Swersky, and Zemel 2017)	ResNet-12	61.20	77.55	68.01	83.91
ProtoNet+RankDNN	ResNet-12	$64.75_{3.55}$	$79.01_{+1.46}$	$70.12_{+2.01}$	$85.00_{+1.09}$
E^3BM (Liu, Schiele, and Sun 2020)	ResNet-12	63.80	80.10	71.20	85.30
E^3BM+RankDNN	ResNet-12	$65.00_{+1.20}$	$80.45_{+0.35}$	$71.72_{+0.52}$	$86.22_{+0.92}$
DeepEMD (Zhang et al. 2020)	ResNet-12	66.50	82.41	72.65	86:03
DeepEMD+RankDNN	ResNet-12	$67.01_{+0.51}$	$84.32_{+1.91}$	$72.80_{+0.15}$	$86.10_{+0.07}$
Distill (Tian et al. 2020)	ResNet-12	64.82	82.14	71.52	86.03
Distill+RankDNN	ResNet-12	$66.34_{+1.52}$	$83.98_{+1.48}$	$72.20_{+0.68}$	$86.45_{+0.42}$
FRN (Wertheimer, Tang, and Hariharan 2021)	ResNet-12	66.25	82.50	72.06	86.37
FRN+RankDNN	ResNet-12	$66.58_{+0.33}$	$82.98_{+0.48}$	$72.21_{+0.15}$	$86.62_{+0.25}$
RENet (Kang et al. 2021)	ResNet-12	67:60	82:58	71:61	85:28
RENet+RankDNN	ResNet-12	$68.01_{+0.41}$	$84.00_{+1.42}$	$71.85_{+0.24}$	$85.98_{+0.70}$
DC (Yang, Liu, and Xu 2021)	WRN	68.57	82.88	78.19	89.90
DC+RankDNN	WRN	$69.54_{+0.97}$	$83.66_{+0.78}$	$78.92_{+0.73}$	$90.20_{+0.30}$

Table 4: Comparison of fusing RankDNN with other methods on *mini*ImageNet and *tiered*ImageNet.

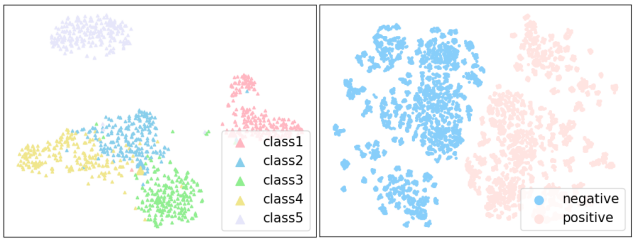

Figure 4: For a 5-way-100-shot task, we visualize the feature descriptors generated by the feature backbone and RankMLP with t-SNE.

Ablation Study

As in Table 5, the results of baselines and the dimension reduction module are obtained from cosine classifiers and linear regression classifiers respectively. It indicates that the PCA will slightly decrease the performance of baselines. Then, we verify the effectiveness of RankDNN by replacing RankDNN with RankSVM (Joachims 2002), and the performance on miniImageNet and CUB drops by 9.37% and 6.01% respectively, and even becomes worse than the S2M2 baseline. Second, we explore different triplet encoding schemes by replacing the vector-Kronecker product with feature disparity(Eq. 10), feature concatenation of triplets, feature disparity of pairwise concatenation, Hadamard product(Eq. 11) and the combination of Kronecker and Hadamard products(Eq. 13).We can see that the Kronecker product has the best generalization performance and is applicable to all backbones and datasets. Besides, the cross-Domain challenge, applicability, learnable parameters and layer numbers of RankDNN experiments are shown in the supplementary material.

$$\psi\left(\langle x_q, x_i, x_j\rangle; \theta\right) = |f_q - f_i| - |f_q - f_j|. \quad (10)$$

$$\psi\left(\langle x_q, x_i, x_j\rangle; \theta\right) = f_q \odot f_i - f_q \odot f_j. \quad (11)$$

$$\psi\left(\langle x_q, x_i, x_j\rangle; \theta\right) = (x_q \otimes x_i)^2 - (x_q \otimes x_j)^2. \quad (12)$$

$$\psi\left(\langle x_q, x_i, x_j\rangle; \theta\right) = (f_q \otimes f_i; f_q \odot f_i) - (f_q \otimes f_j; f_q \odot f_j). \quad (13)$$

Model	*mini*ImageNet	CUB
IE (Rizve et al. 2021)	67.15	80.92
After PCA	65.32	80.11
Feature Differentiation	60.57↓	77.33↓
Triple Concat	20	20
Pairwise Concat	42.50↓	44.65↓
Hadamard	65.00↓	81.64↑
Kronecker	**68.72**↑	**82.93**↑
Polynomial	21.00↓	22.00↓
S2M2 (Mangla et al. 2020)	64.50	80.68
After PCA	65.32	80.11
Kronecker+RankSVM	60.1↓	78.24↓
Feature Differentiation	61.24↓	77.83↓
Hadamard	64.22	79.93↑
Kronecker	66.67↑	**81.78**↑
The Combined Product	**66.88**↑	81.50↑

Table 5: Ablation study on *mini*ImageNet and CUB using two baselines and under the 5-way-1-shot setting.

Conclusions

In this paper, we have introduced RankDNN, a novel pipeline for few-shot learning. It can convert a multiclass classification task into a binary ranking relation classification problem. Accurate binary ranking relation classification is made possible by an informative encoding scheme of image triplets and later uses the simple fully connected network to predict whether there are in a query-relevant-irrelevant order or not. Experiments demonstrate the proposed method achieves new state-of-the-art performance on four benchmarks. Meanwhile, experimental results for different backbones and cross-domain settings demonstrate RankDNN is data efficient and domain agnostic.

Acknowledgments

This work was supported by National Key R&D Program of China (2020AAA0108301), National Natural Science Foun-

dation of China (No.62072112 and No.62106051), Scientific and Technological innovation action plan of Shanghai Science and Technology Committee (No.22511102202), Fudan University Double First-class Construction Fund (No.XM03211178), and the Shanghai Pujiang Program (No.21PJ1400600).

References

Afrasiyabi, A.; Lalonde, J.-F.; and Gagne, C. 2021. Mixture-Based Feature Space Learning for Few-Shot Image Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9041–9051.

Arpit, D.; Jastrzkbski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M. S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. 2017. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, 233–242. PMLR.

Azevedo, A.; Bentes, C.; Castro, M. C.; and Tadonki, C. 2020. Performance Analysis and Optimization of the Vector-Kronecker Product Multiplication. In *2020 IEEE 32nd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, 265–272. IEEE.

Bertinetto, L.; Henriques, J. F.; Torr, P. H.; and Vedaldi, A. 2018. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*.

Briakou, E.; and Carpuat, M. 2020. Detecting Fine-Grained Cross-Lingual Semantic Divergences without Supervision by Learning to Rank. *arXiv preprint arXiv:2010.03662*.

Burges, C.; Shaked, T.; Renshaw, E.; Lazier, A.; Deeds, M.; Hamilton, N.; and Hullender, G. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, 89–96.

Cakir, F.; He, K.; Xia, X.; Kulis, B.; and Sclaroff, S. 2019. Deep metric learning to rank. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1861–1870.

Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C.; and Huang, J.-B. 2019a. A Closer Look at Few-shot Classification. In *International Conference on Learning Representations*.

Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C. F.; and Huang, J.-B. 2019b. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*.

Chen, Y.; Liu, Z.; Xu, H.; Darrell, T.; and Wang, X. 2021. Meta-Baseline: Exploring Simple Meta-Learning for Few-Shot Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9062–9071.

De Launey, W.; and Seberry, J. 1994. The strong Kronecker product. *Journal of Combinatorial Theory, Series A*, 66(2): 192–213.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 1126–1135. PMLR.

Ge, W. 2018. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 269–285.

Ge, W.; and Yu, Y. 2017. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1086–1095.

Gidaris, S.; Bursuc, A.; Komodakis, N.; Pérez, P.; and Cord, M. 2019. Boosting few-shot visual learning with self-supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8059–8068.

Gidaris, S.; and Komodakis, N. 2018. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4367–4375.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hong, J.; Fang, P.; Li, W.; Zhang, T.; Simon, C.; Harandi, M.; and Petersson, L. 2021. Reinforced attention for few-shot learning and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 913–923.

Jamal, M. A.; and Qi, G.-J. 2019. Task Agnostic Meta-Learning for Few-Shot Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 11719–11727.

Joachims, T. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 133–142.

Kang, D.; Kwon, H.; Min, J.; and Cho, M. 2021. Relational Embedding for Few-Shot Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8822–8833.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Langville, A. N.; and Stewart, W. J. 2004. The Kronecker product and stochastic automata networks. *Journal of computational and applied mathematics*, 167(2): 429–447.

Levi, E.; Xiao, T.; Wang, X.; and Darrell, T. 2021. Rethinking Preventing Class-Collapsing in Metric Learning With Margin-Based Losses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10316–10325.

Li, P.; Wu, Q.; and Burges, C. 2007. Mcrank: Learning to rank using multiple classification and gradient boosting. *Advances in neural information processing systems*, 20: 897–904.

Liu, C.; Yu, H.; Li, B.; Shen, Z.; Gao, Z.; Ren, P.; Xie, X.; Cui, L.; and Miao, C. 2021. Noise-resistant Deep Metric Learning with Ranking-based Instance Selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6811–6820.

Liu, Y.; Schiele, B.; and Sun, Q. 2020. An ensemble of epoch-wise empirical bayes for few-shot learning. In *European Conference on Computer Vision*, 404–421. Springer.

Lu, J.; Gong, P.; Ye, J.; and Zhang, C. 2020. Learning from Very Few Samples: A Survey. *arXiv preprint arXiv:2009.02653*.

Mamut, A.; and Vumar, E. 2008. Vertex vulnerability parameters of Kronecker products of complete graphs. *Information Processing Letters*, 106(6): 258–262.

Mangla, P.; Kumari, N.; Sinha, A.; Singh, M.; Krishnamurthy, B.; and Balasubramanian, V. N. 2020. Charting the right manifold: Manifold mixup for few-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2218–2227.

Masuda, K. 2003. A ranking model of proximal and structural text retrieval based on region algebra. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, 50–57.

Palangi, H.; Deng, L.; Shen, Y.; Gao, J.; He, X.; Chen, J.; Song, X.; and Ward, R. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4): 694–707.

Pang, L.; Lan, Y.; Guo, J.; Xu, J.; Xu, J.; and Cheng, X. 2017. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 257–266.

Prosser, B. J.; Zheng, W.-S.; Gong, S.; Xiang, T.; Mary, Q.; et al. 2010. Person re-identification by support vector ranking. In *BMVC*, volume 2, 6. Citeseer.

Qiao, S.; Liu, C.; Shen, W.; and Yuille, A. L. 2018. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7229–7238.

Ren, M.; Triantafillou, E.; Ravi, S.; Snell, J.; Swersky, K.; Tenenbaum, J. B.; Larochelle, H.; and Zemel, R. S. 2018. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*.

Rizve, M. N.; Khan, S.; Khan, F. S.; and Shah, M. 2021. Exploring Complementary Strengths of Invariant and Equivariant Representations for Few-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10836–10846.

Shashua, A.; Levin, A.; et al. 2003. Ranking with large margin principle: Two approaches. *Advances in neural information processing systems*, 961–968.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 4077–4087.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1199–1208.

Tang, S.; Chen, D.; Bai, L.; Liu, K.; Ge, Y.; and Ouyang, W. 2021. Mutual CRF-GNN for Few-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2329–2339.

Tian, Y.; Wang, Y.; Krishnan, D.; Tenenbaum, J. B.; and Isola, P. 2020. Rethinking few-shot image classification: a good embedding is all you need? In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, 266–282. Springer.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29: 3630–3638.

Wang, X.; Hua, Y.; Kodirov, E.; Hu, G.; Garnier, R.; and Robertson, N. M. 2019. Ranked list loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5207–5216.

Wang, X.; Zhang, H.; Huang, W.; and Scott, M. R. 2020. Cross-batch memory for embedding learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6388–6397.

Weichsel, P. M. 1962. The Kronecker product of graphs. *Proceedings of the American mathematical society*, 13(1): 47–52.

Wertheimer, D.; Tang, L.; and Hariharan, B. 2021. Few-Shot Classification With Feature Map Reconstruction Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8012–8021.

Yang, J.; Zhang, D.; Frangi, A. F.; and Yang, J.-y. 2004. Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 26(1): 131–137.

Yang, S.; Liu, L.; and Xu, M. 2021. Free lunch for few-shot learning: Distribution calibration. *arXiv preprint arXiv:2101.06395*.

Ye, H.-J.; Hu, H.; Zhan, D.-C.; and Sha, F. 2020. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8808–8817.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2021. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3): 107–115.

Zhang, C.; Cai, Y.; Lin, G.; and Shen, C. 2020. DeepEMD: Few-Shot Image Classification With Differentiable Earth Mover's Distance and Structured Classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12203–12213.

Zhang, J.; and van Genabith, J. 2020. Translation Quality Estimation by Jointly Learning to Score and Rank. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2592–2598.

Zhou, H.-Y.; Chen, X.; Zhang, Y.; Luo, R.; Wang, L.; and Yu, Y. 2022. Generalized Radiograph Representation Learning via Cross-supervision between Images and Free-text Radiology Reports. *Nature Machine Intelligence*, 4: 32–40.