TransVLAD: Focusing on Locally Aggregated Descriptors for Few-Shot Learning

Haoquan Li*[1], Laoming Zhang*[2], Daoan Zhang[1], Lang Fu[2], Peng Yang[2,3], and Jianguo Zhang†[1,4]

[1] Research Institute of Trustworthy Autonomous Systems, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China
[2] Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China
[3] Department of Statistics and Data Science, Southern University of Science and Technology, Shenzhen, China
[4] Peng Cheng Lab, Shenzhen, China
{12032492, 12032505, 12032503, 12032485}@mail.sustech.edu.cn, {yangp, zhangjg}@sustech.edu.cn

Abstract. This paper presents a transformer framework for few-shot learning, termed TransVLAD, with one focus showing the power of locally aggregated descriptors for few-shot learning. Our TransVLAD model is simple: a standard transformer encoder following a NeXtVLAD aggregation module to output the locally aggregated descriptors. In contrast to the prevailing use of CNN as part of the feature extractor, we are the first to prove self-supervised learning like masked autoencoders (MAE) can deal with the overfitting of transformers in few-shot image classification. Besides, few-shot learning can benefit from this general-purpose pre-training. Then, we propose two methods to mitigate few-shot biases, supervision bias and simple-characteristic bias. The first method is introducing masking operation into fine-tuning, by which we accelerate fine-tuning (by more than 3x) and improve accuracy. The second one is adapting focal loss into soft focal loss to focus on hard characteristics learning. Our TransVLAD finally tops 10 benchmarks on five popular few-shot datasets by an average of more than 2%.

Keywords: Few-Shot Learning, Transformers, Self-supervised Learning, NeXtVLAD, Focal Loss

1 Introduction

The success of deep learning largely depends on the expansion of data scaling and model capacity [25]. However, the extreme hungry for data hampered its

* H. Li and L. Zhang made equal contributions to this work.
† J. Zhang is the corresponding author.

wide application. This limitation has promoted a wide range of research fields such as transfer learning [38], domain adaptation [50], semi-supervised [4] and unsupervised learning [19]. Few-shot learning is also one of them working on the low data regime [42,48,18,12]. It aims to get adaptive models capable of learning new objects or concepts with only a few labeled samples, just like humans do.

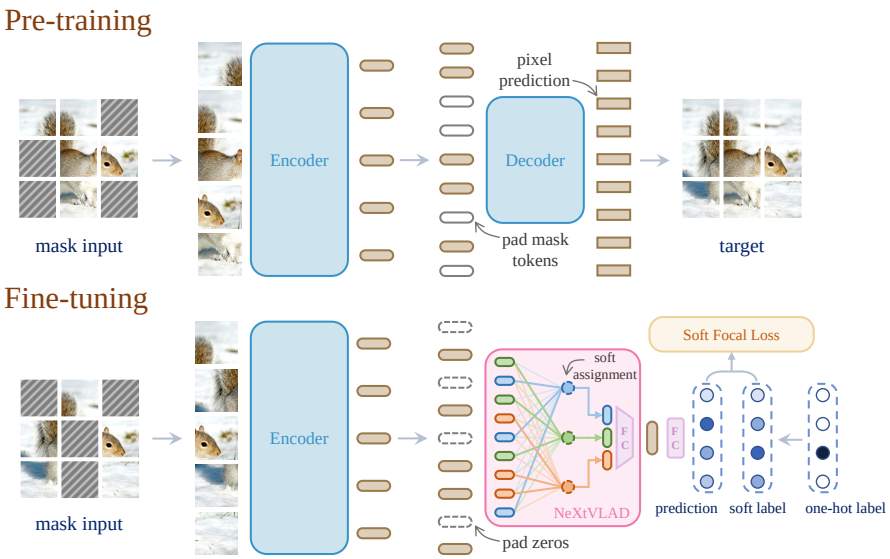

Fig. 1. Pre-training and fine-tuning use the same training set (base classes). In fine-tuning, the feature extractor (TransVLAD) consists of a transformer-based encoder and a NeXtVLAD aggregation module. Soft focal loss is proposed for reducing bias on simple characteristics

While transformer-based models [47] have many nice properties and have topped a bunch of benchmarks in many fields of computer vision, e.g., instance discrimination [34], object detection [7], and semantic segmentation [34], convolutional neural networks [25] still seem to be an inevitable choice for few-shot image classification. This is partly because transformers is more prone to overfitting on the scale of popular few-shot classification datasets. A recent study shows a new paradigm of self-supervised learning based on the idea of masking image modeling (MIM) [2,21,52] is robust to the type and size of the pre-training data [17], and even pre-training on target task data can still obtain comparable performance. This inspires us to use such an approach called *masked autoencoders* (MAE) [21] to pre-train transformer-based models on the target dataset. In addition, the MAE pre-training also benefits few-shot learning from two aspects. First, self-supervised learning can provide unbiased features for classes. Second, the prediction of patch pixels enables models to be with *spatial awareness*. The features output by the pre-trained MAE encoder is aware of spatial

information, so we call them *patch features*. In fine-tuning, without the supervision of patch pixels, the output features will become more abstract and we call them *local features*. The understanding of new concepts usually derives from the reorganization of existing concepts (centaur can be seen as recombination of a human part and a horse part). Therefore, dividing complete image information into discrete local features is beneficial to generalize to new concepts or objects.

To get the global features of an image, the most common way of aggregating all of the local features is by "mean pooling" [16]. This method, a little bit reckless, will lose abundant local characteristics. Another opposite choice is by keeping all of the local features as the complete representation of an image. This operation will contrarily lose the ability of global representation and result in learning incompact data distributions. We finally choose the NeXtVLAD module [29], a compromise choice between mean pooling and doing nothing, to aggregate local features provided by the transformer encoder. NeXtVLAD softly assigns local features to each cluster center, and concatenates the features of all clusters as the image-level feature, Fig. 1. The output features of NeXtVLAD are called *locally aggregated descriptors*. So far, our proposed feature extractor, TransVLAD, has been established.

Few-shot learning exists two learning biases. The first bias is *supervision bias* [15]. Models will only keep features useful for minimizing the losses for base classes (visible for training) but discard information that might help with novel classes. To eliminate the influence of supervision bias, we transfer masking operation from pre-training into fine-tuning. This operation is somewhat like adding "dropout" [43] to the input. Local features that are crucial for base classes may be masked, then the classification must rely on the minor features to infer the class. According to the experimental results, we finally choose the masking ratio of 70% and it accelerates our fine-tuning process by more than 3 times.

The second bias is *simple-characteristic bias*. To efficiently fit data to the real distribution, the model will tend to learn simpler features or fit simpler classes first. Then many hard features or those features for hard samples might be underestimated. This is not good for the generalization of novel classes. To avoid simple-characteristic bias, we replace the cross-entropy loss with soft focal loss, a soft version of focal loss [30] we proposed, which gives more weight to hard samples. The soft focal loss works at the sample level, but together with the masking operation, it will be able to work at the feature level. This is because the masking operation transforms a training sample into some uncertain parts which only contain part features of an image.

Our contributions can be summarized as follows:

1. In contrast to the prevailing use of CNN as part of the feature extractor, we are the first to prove self-supervised learning like MAE can deal with the overfitting of transformers in few-shot image classification.
2. We introduce NeXtVLAD to aggregate local features for few-shot learning. Masking operation and soft focal loss are yielded to solve the supervision bias and simple-characteristic bias, respectively.

3. Our TransVLAD tops 10 benchmarks on five popular few-shot datasets by an average of more than 2%. It also shows a great effect on the cross-domain few-shot scenario.

2 Related Work

2.1 Transformers in Few-Shot Learning

Since ViT introduced transformers to computer vision, transformer-based algorithms have conquered lots of visual tasks [34,7]. Recently, some researchers tried to transfer it in few-shot learning and achieved significant improvements [15,31,20,9]. CrossTransformers [15] designed a transformer block to capture the spatial-alignment relationship between images. SSFormer [9] proposed a sparse spatial transformer layer to select key patches for few-shot classification. These studies all regard the transformer block as an auxiliary module following CNN to enhance accuracies. As far as we know, we are the first to use transformer blocks without CNN as the feature extractor for few-shot learning.

2.2 Self-supervised Learning

Self-supervised learning [10,22,8,11,21,2] often designs surrogate supervision signal with image intrinsic properties. Recently, inspired by natural language processing's great success in masked pre-training methods [14,33,13,39], like Bert [14], a similar implementation for image patches have been studied [21,17,2,52]. BEiT [2] predicts the corresponding discrete tokens for masked patches. Those tokens are generated by another autoencoder trained in advance. MAE [21] simply masks random patches and reconstructs the missing pixels. SplitMask [17] deeply studies these kinds of approaches and concludes that a large-scale dataset is not necessary for masked pre-training. In addition, MAE has lower computing resource requirements than contrastive learning which often needs a large batch size for best performance.

2.3 Traditional Few-Shot Learning

Traditional few-shot learning can be roughly divided into the following classes. (1) Optimized-based methods [26,18]. MAML [18] tries to find a set of initialization parameters where the model can converge fast and effectively to novel tasks. (2) Metric-based methods [42,54,45,3,27]. Since prototypical network [42] was proposed, Metric-based methods have become the most common methods in few-shot learning. The key to this method is how to get a feature extractor with better generalization. In evaluation, the average of features for each class in the support set (labeled few samples) is viewed as a class prototype (center). Query sample (test sample) will be finally classified into a class whose prototype is nearest to this feature. (3) Methods based on data augmentation [53,1,51]. [53] expands data scale at the feature level by treating each value as a Gaussian

sampling. In our paper, we use a metric-based evaluation method as most recent papers do.

Recently, some methods based on local features are proposed in the few-shot image classification field. They usually generate local features by a fully convolutional network without the final global average pooling. DeepEMD [55] measures the Earth Mover's Distance of local features as image-to-image distance; DN4 [28] revisits NBNN(Naive-Bayes Nearest-Neighbor) [6] approach and suggests a method based on k-nearest neighbors for producing image-to-class distance at local-level features.

3 Methodology

Our method is a two-stage method that contains the MAE pre-training and a designed fine-tuning. After the pre-training, our TransVLAD model is constructed by the MAE encoder followed by a NeXtVLAD aggregation module. Subsequently, we optimize the fine-tuning process by two simple improvements, masking input patches and applying soft focal loss, under the assumptions of supervision bias and simple-characteristic bias in few-shot learning.

3.1 Problem Definition

Few-shot classification can be defined by two data sets with different classes. Training data set contains base classes with abundant labeled data, $D_{train} = \{(\boldsymbol{x_i}, y_i)|y_i \in C_{base}\}$. Testing data set contains novel classes with scarce labeled data, $D_{eval} = \{(\boldsymbol{x_i}, y_i)|y_i \in C_{novel}\}$. The two data sets do not intersect, $C_{base} \cap C_{novel} = \varnothing$. The standard N-way K-shot testing condition is that given N novel classes with K labeled samples per class (termed as support set), to classify the same N classes with Q unlabeled samples per class (termed as query set). A classic idea to this problem is to train a feature extractor on D_{train} and then test by assigning classes to query samples with the feature difference between query and support set. Then the problem has changed to be how to train a more generalized feature extractor.

3.2 Masked Autoencoders Pre-training

Masked Autoencoders (MAE) pre-training is a simple approach to learn general features by reconstructing missing pixels from masked patches. Its main contributions include yielding a very big masking ratio to eliminate redundancy and an ingenious framework design. These creativities make MAE training efficiently and effectively.

Encoder. MAE encoder is a standard ViT. It projects patches into linear embeddings by a linear projection before adding positional embeddings. Subsequently, transformer layers will process them into more representative patch-wise embeddings. The only difference is that MAE encoder just processes 25% patches to reduce image redundancy while increasing computing speed.

Decoder. MAE decoder is designed to be lightweight to process full set of patch tokens (padded with mask tokens) as Fig. 1 shows. Decoder is also composed of several transformer layers following a linear projection to project patch embeddings to original pixels of corresponding masked patches. Before inputting decoder, tokens need to be added positional embeddings again for noticing positional information of masked tokens. The small fraction of input to encoder and the lightweight design of decoder together speed up the training of MAE model by three times or more.

3.3 NeXtVLAD Module

In fine-tuning, we yield NeXtVLAD [29] module to aggregate features output by the transformer-based encoder, Fig. 1. NeXtVLAD is a neural network version of VLAD [23] which focuses on the distribution differences between local features and cluster centers. It was first proposed to aggregate frame-level features into a compact video feature. This method models learnable cluster centers and computes their differences to local features, in our trials, patch-level features. With this module, we can get aggregated features of local features. The whole principle of NeXtVLAD is shown in Fig. 2.

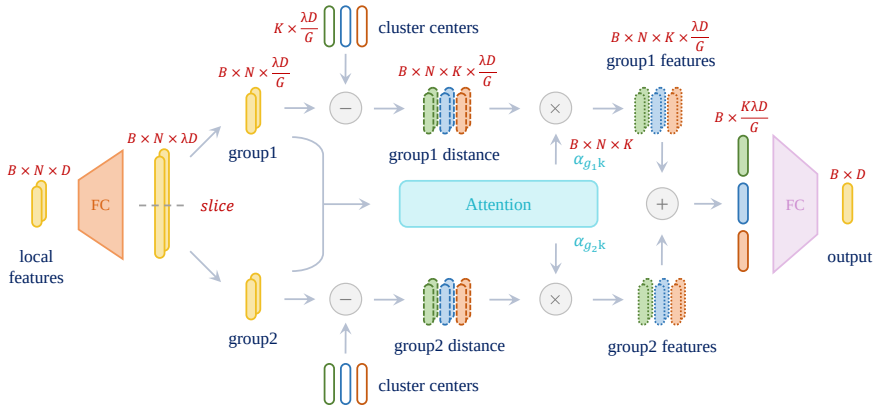

Fig. 2. The principle of NeXtVLAD module. In this case, $B = 1$, $N = 2$, $G = 2$, $K = 3$. It works by computing weighted sum of differences between cluster centers and grouped features. Finally, aggregated features will be flattened into a vector and projected to dimension 768

Considering an image with N patches, after transformer-based encoder processing, the D-dimensional local features can be denoted as $\{\boldsymbol{x}_i\}_{i=1}^{N}$. First, local features will be expanded to higher dimensions λD as $\{\widetilde{\boldsymbol{x}}_i\}_{i=1}^{N}$ by a linear projection L_e, λ is a multiplier:

$$\widetilde{\boldsymbol{x}}_i = L_e\left(\boldsymbol{x}_i|\theta_e\right), \tag{1}$$

in which θ_e denotes the parameters of L_e. Then it will be equally sliced into G groups $\{\widetilde{\boldsymbol{x}}_i^{(g)}\}_{i=1}^N, g \in \{1, 2, ..., G\}$. Grouping here is meant to decompose features into more independent parts before aggregating together. The underlying assumption is that each local feature can incorporate multiple concepts or objects. In order to calculate the differences between grouped features and cluster centers $\{\boldsymbol{c}_k\}_{k=1}^K$, cluster centers are initialized to $\lambda D/G$ dimensions. For each cluster, it will finally provide a compact feature F_k by weighted summing all corresponding differences:

$$\begin{gathered} F_k = \mathbb{N}\left(X|C, \theta_e, \theta_g, \theta_k\right) = \sum_{i,g} \alpha_{gk}\left(\widetilde{\boldsymbol{x}}_i|\theta_g, \theta_k\right)\left(\widetilde{\boldsymbol{x}}_i^{(g)} - \boldsymbol{c}_k\right) \\ k \in \{1, ..., K\}, i \in \{1, ..., N\}, g \in \{1, ..., G\}, \end{gathered} \tag{2}$$

where $\mathbb{N}$ is the NeXtVLAD module, θ means corresponding parameters and C means all learnable cluster centers. α_{gk} decides the soft assignment of the corresponding difference in the final representation and is calculated by two parts:

$$\alpha_{gk}\left(\widetilde{\boldsymbol{x}}_i|\theta_g, \theta_k\right) = \alpha_g\left(\widetilde{\boldsymbol{x}}_i|\theta_g\right)\alpha_k^{(g)}\left(\widetilde{\boldsymbol{x}}_i|\theta_k\right) \tag{3}$$

where

$$\alpha_g\left(\widetilde{\boldsymbol{x}}_i|\theta_g\right) = \sigma\left(L_g\left(\widetilde{\boldsymbol{x}}_i|\theta_g\right)\right), \tag{4}$$

$$\alpha_k^{(g)}(\widetilde{\boldsymbol{x}}_i|\theta_k) = \frac{e^{L_k^{(g)}\left(\widetilde{\boldsymbol{x}}_i|\theta_k^{(g)}\right)}}{\sum_{k'=1}^K e^{L_{k'}^{(g)}\left(\widetilde{\boldsymbol{x}}_i|\theta_{k'}^{(g)}\right)}}. \tag{5}$$

α_g represents the attention for each group while $\alpha_k^{(g)}$ measures the soft assignment of $\widetilde{\boldsymbol{x}}_i^{(g)}$ to the cluster k in probabilistic form. $\sigma(.)$ is a sigmoid function to scale group attention into $(0, 1)$. L_g and L_k are different linear layers with parameters θ_g and θ_k respectively. To show formula concisely, we split L_k into $L_k^{(g)}, g \in \{1, 2, ..., G\}$. In practice, a single layer L_k is used.

Finally, we will get K features of $\lambda D/G$ dimensions for an image. We flatten and project it into a lower dimension with another linear layer, followed by a batch normalization and a ReLU activation function. Until now, NeXtVLAD have compacted patch-level features into an image-level feature.

3.4 Masking Operation in Fine-tuning

In MAE pre-training, encoder only observes part of patches (25%). This brings great benefits on both efficiency and effectiveness. The idea of prompt [32], a new idea about narrowing the gap between pre-training and fine-tuning, inspired us to design similar operation. Therefore, we decided to transfer masking operation

in fine-tuning as well. However, in evaluation, masking patches will hinder full access to image information which is obviously unreasonable. To deal with this problem, a variable-length processing for our model is necessary. Fortunately, transformers and NeXtVLAD were created for text or video field and they naturally support for variable-length processing.

The difference between MAE and our masking operation is that we need to restore the original order for visible patches and pad zeros to masked ones before input to NeXtVLAD module. In evaluation, we just let all patches get through our whole feature extractor, including transformer blocks and NeXtVLAD module. Similar to MAE, we surprisingly find 70% masking ratio is a more suitable choice according to our ablation experiments, Fig. 5.

Masking operation in our TransVLAD model has four benefits:

1. It eliminates information redundancy in images [21].
2. It can be regarded as a kind of data augmentation.
3. It speeds up the training process by about three times.
4. It shortens the difference of tasks between pre-training and fine-tuning.

3.5 Soft Focal Loss

To avoid simple-characteristic bias and enhance attention on difficult characteristics or hard samples in training, we use focal loss [30] to replace the cross-entropy loss. Focal loss is a changed version of cross-entropy loss and proposed to address extreme imbalance between foreground and background classes in one-stage object detection scenario. It is defined as:

$$\mathcal{FL}(p) = -\alpha_i \left(1-p\right)^{\gamma} \log\left(p\right) \tag{6}$$

in which p is the prediction of the class that the sample really belongs to. $\gamma \geq 0$ is the tunable focusing parameter. The term $(1-p)$ measures the learning degree of each sample. The bigger γ is, the more focusing on hard classified samples. α_i is appied to balance the classes with different sample sizes.

This formula uses one-hot encoded labels by default. However, in our settings, mixup [56] and label smoothing [36] have been used to reduce overfitting. They encode labels into a soft version which may have target values for all classes. To suitable for soft targets, we expand focal loss to a soft version, called soft focal loss:

$$\mathcal{SFL}(\boldsymbol{p}, \boldsymbol{t}) = -\sum_{i=1}^{C} \alpha_i \left[\left(t_i - p_i\right)^2\right]^{\frac{\gamma}{2}} t_i \log\left(p_i\right), \tag{7}$$

in which p_i, t_i are the prediction and target to the i_{th} class respectively. The term $(t_i - p_i)$ measures the difference between them to influence the weight of relative loss term. C is the number of classes to predict. In our setting, α_i is always set to be one because our training classes are balanced.

Soft focal loss is suggested to be used with masking operation. Because with masking, the difficulty varies by the remaining part of images, so the soft focal loss can work at the local feature level.

3.6 Whole Classification

Our paper uses whole classification task, the most common setting for classification with cross-entropy loss (soft focal loss in our method), instead of meta-training method which is popular in few-shot learning. This is because some recent studies have found that whole classification can obtain comparable results to meta-training and has better generalization in novel classes while meta-training concerns more on testing condition [12].

4 Experiments

4.1 Datasets

During the evaluation, we compared our methods on five standard datasets for few-shot classification, *mini*ImageNet [48], *tiered*ImageNet [40], CIFAR-FS [5], FC100 [37], and CUB [49].

*mini*ImageNet and *tiered*ImageNet are subsets of ImageNet [41]. *mini*ImageNet consists of 100 classes with 600 samples per class and is randomly divided into three disjoint sets of the training set (64 classes), validation set (16 classes), and testing set (20 classes). *tiered*ImageNet, a bigger version of *mini*ImageNet, contains 608 classes with 1200 samples per class and is randomly split into 351/97/160 for train/val/test. CIFAR-FS and FC100 both are variants of CIFAR-100. CIFAR-FS is randomly split into 64/16/20 classes for train/val/test, while FC100 is divided into 60/20/20 classes according to 20 superclasses. Each superclass contains five similar classes with a more generalized concept. So, FC100 is a harder dataset created for reducing *semantical overlaps* between training and evaluation. CUB-200-2011 (CUB) is a fine-grained dataset of 200 bird species with total 11,788 images. It is randomly split into 100/50/50 for train/val/test. The image size of *mini*ImageNet, *tiered*ImageNet, CUB is 84x84 while for CIFAR-FS and FC100 is 32x32.

4.2 Implementation Details

Training Details. For each dataset, we use self-supervised pre-training and supervised fine-tuning to train the model with base classes and test the performance using novel classes. In the pre-training phase, we use the MAE strategy and parameters. The encoder is a 12-layer transformer model with dimension 768 and the decoder is a 4-layer transformer model with dimension 384. The masking ratio is 75% and training for 1600 epochs. In the fine-tuning stage, we just keep the encoder and add NeXtVLAD module ($\lambda = 4, K = 64, G = 8$) after it. We follow the default training settings for BEiT [2] except for the initial learning rate is 7e-4. Our masking ratio here is 70%, focusing parameter γ is 2 and fine-tuning for 100 epochs. More specific settings can be found in supplemental material.

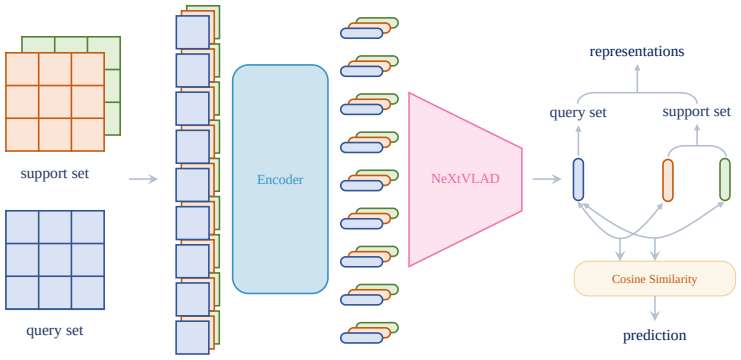

Fig. 3. Given a few-shot task, we compute an average feature for samples of each class in *support set*, then we classify the sample in *query set* by nearest neighbor method with cosine similarity

Evaluation. The overall test flow is shown in Fig. 3. It is worth noting that we do not perform masking operation during validation and testing. We evaluate our method based on ProtoNet [42] setting with randomly sample 600 N-way K-shot tasks from the novel set, and take averaged top-1 classification accuracy. To get a fair comparison with the previous works, we perform model selection based on the validation set.

Table 1. Comparison with the prior and current state-of-the-art methods on *mini*ImageNet, *tiered*ImageNet datasets

Year	Methods	*mini*ImageNet		*tiered*ImageNet	
		1-shot	5-shot	1-shot	5-shot
2017	ProtoNet [42]	54.16±0.82	73.68±0.65	53.31±0.89	72.69±0.74
2018	RelationNet [44]	52.19±0.83	70.20±0.66	54.48±0.93	71.32±0.78
2019	DCO [26]	62.64±0.61	78.63±0.46	65.99±0.72	81.56±0.53
2020	Meta-baseline [12]	63.17±0.23	79.26±0.17	68.62±0.27	83.29±0.18
2020	DeepEMD [55]	65.91±0.82	82.41±0.56	71.16±0.87	83.95±0.58
2020	S2M2 [35]	64.93±0.18	**83.18**±0.11	**73.71**±0.22	**88.59**±0.14
2021	RENet [24]	**67.60**±0.44	82.58±0.30	71.61±0.51	85.28±0.35
2021	SSFormers [9]	67.25±0.24	82.75±0.20	72.52±0.25	86.61±0.18
2022	TransVLAD	**68.24**±0.59	**84.42**±0.23	**77.20**±0.60	**90.74**±0.32

4.3 Compare to the State-of-the-art Methods

Table 1 and Table 2 compare our method with the current state-of-the-art methods on five datasets. As shown in the tables our method achieves the best results

Table 2. Comparison with the prior and current state-of-the-art methods on CIFAR-FS, FC100, CUB datasets

Year	Methods	CIFAR-FS		FC100		CUB	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
2017	ProtoNet [42]	55.50±0.70	72.00±0.60	37.50±0.60	52.50±0.60	71.88±0.91	87.42±0.48
2018	RelationNet [44]	55.00±1.00	69.30±0.80	-	-	68.65±0.91	81.12±0.63
2019	DCO [26]	72.00±0.70	84.20±0.50	41.10±0.60	55.50±0.60	-	-
2020	DeepEMD [55]	-	-	-	-	75.65±0.83	88.69±0.50
2020	S2M2 [35]	**74.81**±0.19	**87.47**±0.13	-	-	**80.68**±0.81	90.85±0.44
2021	RENet [24]	74.51±0.46	86.60±0.32	-	-	79.49±0.44	**91.11**±0.24
2021	SSFormers [9]	74.50±0.21	86.61±0.23	**43.72**±0.21	**58.92**±0.18	-	-
2022	TransVLAD	**77.48**±0.41	**89.82**±0.42	**47.66**±0.12	**64.25**±0.18	**82.66**±1.22	**92.45**±0.80

on all five datasets. Since we only tune parameters on *mini*ImageNet and copy them for the other four datasets, these results demonstrate the excellent generalization of our model.

We find that the increase of performance on *tiered*ImageNet (1-shot +3.5%, 5-shot +2.2%) is apparently larger than on *mini*ImageNet (1-shot +0.6%, 5-shot +1.2%), while the only difference is the data scale. We hypothesize that this is mainly because pre-training on a bigger dataset can get better generalization which is beneficial to distinguish new objects. Secondly, our model presents great transferability across superclasses on FC100 (1-shot +3.9%, 5-shot +5.3%). It inspired us to do cross-domain few-shot studies.

4.4 Cross-Domain Few-Shot Learning

Cross-domain few-shot image classification, where unseen classes and examples come from diverse data sources, has seen growing interest [46]. To further validate the transferability of our method, we also conducted cross-dataset evaluation experiments in a simple way, training with one dataset and testing with another different dataset. From table 3, We can see that the transferring effect of *mini*ImageNet on CUB is better than the previous method, and the mutual evaluation effect of *mini*ImageNet and CIFAR-FS is close to the result of the intra-domain training, partly because they both are randomly divided. The poor transfer of the CUB is expected, because there are only pictures of birds in the CUB, resulting in difficulty to learn contents from other fields. These results demonstrate that our model has great cross-domain transferability.

5 Discussion and Ablation Studies

5.1 Overall Analysis

Our method consists of four main components: MAE pre-training, NeXtVLAD feature aggregation module, masked fine-tuning, and soft focal loss. In this section, we conduct ablation studies to analyze how each component affects the

Table 3. Few-shot cross-domain evaluation results between *mini*ImageNet, CIFAR-FS and CUB where grey numbers denote intra-domain results

Methods	Training Dataset	*mini*ImageNet		CIFAR-FS		CUB	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
DCO [26]	*mini*ImageNet	62.64±0.61	78.63±0.46	-	-	44.79±0.75	64.98±0.68
S2M2 [35]	*mini*ImageNet	64.93±0.18	83.18±0.11	-	-	**48.42**±0.84	**70.44**±0.75
TransVLAD	*mini*ImageNet	68.24±0.59	84.42±0.23	66.00±0.85	83.26±0.50	**50.45**±0.12	**72.20**±0.34
TransVLAD	CIFAR-FS	54.56±1.35	72.55±1.00	77.48±0.41	89.82±0.42	46.29±1.23	65.81±1.01
TransVLAD	CUB	39.36±0.86	52.78±1.34	34.12±0.59	45.91±1.56	82.66±1.22	92.45±0.80

few-shot recognition performance. We show the individual and combined effects of these components in Table 4. Specifically, if the NeXtVLAD is not selected, mean pooling will replace it for feature aggregation. Similarly, the cross-entropy loss is selected as the replacement for soft focal loss.

Table 4. The individual and combined effects of MAE pre-training, NeXtVLAD module, masked fine-tuning, and soft focal loss are studied. The experiments are conducted on *mini*ImageNet. we can find that each part of our model has an important contribution

Pre-train	NeXtVLAD	Mask	Soft Focal Loss	1-shot	5-shot
				34.55	45.73
	✓	✓	✓	44.56	61.34
✓				64.10	81.40
✓		✓		65.46	82.05
✓		✓	✓	66.79	83.40
✓	✓			65.62	81.70
✓	✓	✓		66.91	83.32
✓	✓	✓	✓	**68.24**	**84.42**

We can see that the transformer-based model is seriously overfitting compared to the CNN baseline [12] when directly trained. But with the join of MAE pre-training, the performance of the model will be comparable to the CNN baseline. Interestingly, our method can still improve the performance with no pre-training. In all of our experiments, NeXtVLAD module provides a steady boost. With the masking operation, it further improves the *mini*ImageNet by 2.8% (1-shot), while the consuming time of fine-tuning is shortened by one-third. The soft focal loss also plays an important role in our TransVLAD. It surely improves the generalization of novel classes and shortens the prediction bias for different classes which we will discuss it later.

5.2 NeXtVLAD Feature Aggregation

We perform a t-SNE visualization of embeddings generated by Meta-Baseline [12], MAE++ (MAE adds masked fine-tuning and soft focal loss), and our TransVLAD (MAE ++ adds NeXtVLAD module) on the test set of *mini*ImageNet (see Fig. 4). We observe that our methods, both MAE++ and TransVLAD, produce more compact features for the same class. Besides, TransVLAD only adds the NeXtVLAD module compared to MAE++, and apparently gets a more discriminative feature distribution with larger boundaries between clusters. We hypothesize that this behavior is due to the learned cluster centers of NeXtVLAD. Features are guided to be close to those related centers.

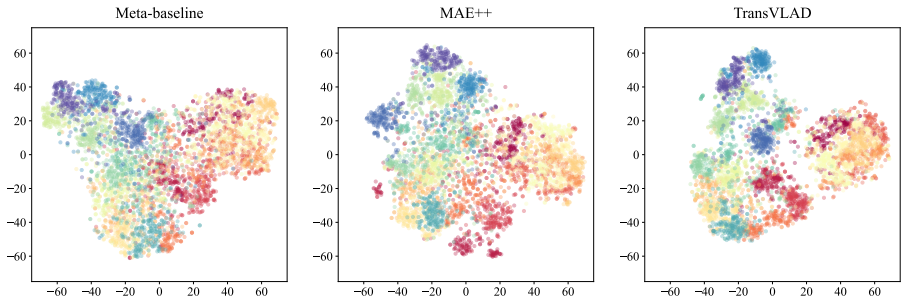

Fig. 4. t-SNE visualization of features on *mini*ImageNet test set produced by Meta-Baseline [12], MAE++ (added masking operation and soft focal loss) and TransVLAD (added NeXtVLAD over MAE++)

5.3 Masking Ratio

Our feature extractor, including transformer blocks and NeXtVLAD module, is very flexible with the input size while the output size is fixed. So, we can mask input patches at any ratio, called masking ratio. Fig. 5 shows the influence of the masking ratio. Similar to the MAE's result, the optimal masking ratio is surprisingly high, up to 70%. This behavior makes sense. Masking operation is somewhat like *dropout* at the input. It prevents our model from over-relying on some key features but ignoring the learning of other features, namely, overfitting at base classes.

By skipping the masked patches, we accelerate the training process by more than 3 times. In addition, the memory usage is greatly reduced (about 70%), which allows us to train larger models with the same batch size. In fact, fine-tuning our TransVLAD for 100 epochs with 128 batch size on *mini*ImageNet only needs less than 4 hours on one RTX3090, and the memory usage is less than 8G. This is extremely fast for a ViT-Base [16] encoder.

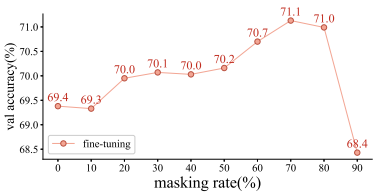

Fig. 5. Validation accuracy for changing masking ratio

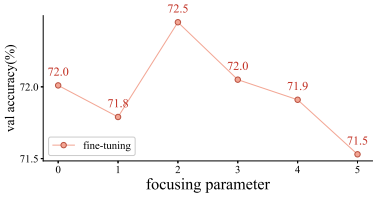

Fig. 6. Validation accuracy for changing focusing parameter γ

5.4 Focusing Parameter

Similar to the masking ratio, we conduct a separate experiment to evaluate the influence of the soft focal loss with different values of focusing parameter γ. Fig. 6 shows that the best result is obtained when γ is equal to 2. When the focusing parameter is small, the effect of focusing on hard examples is not obvious. When it is too big, the model will continue to overreact to unpredictable samples and ignore what it could have learned.

Soft focal loss is suggested to be used with masking operation. Because if there is no mask, soft focal loss only works at the sample level. With masking operation, the difficulty varies by the remaining part of images, so the soft focal loss can work at the local feature level.

6 Conclusion

The transformer-based network has strong potential for few-shot learning, and the core problem that restricts it to dominate few-shot learning is its serious overfitting. Our paper gives an efficient solution. The new self-supervised pre-training paradigm, such as masked autoencoders, is able to reduce its overfitting and simultaneously improve the model generalization.

The model we designed, TransVLAD, takes full advantage of the nature of transformers and few-shot tasks, and it not only significantly improves the performance but also improves the training speed.

Acknowledgement

This work is supported in part by National Key Research and Development Program of China (Grant No. 2021YFF1200800), and the Stable Support Plan Program of Shenzhen Natural Science Fund (Grant No. 20200925154942002).

References

1. Antoniou, A., Storkey, A., Edwards, H.: Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340 (2017)
2. Bao, H., Dong, L., Wei, F.: Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254 (2021)
3. Bateni, P., Goyal, R., Masrani, V., Wood, F., Sigal, L.: Improved few-shot visual classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14493–14502 (2020)
4. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: A holistic approach to semi-supervised learning. Advances in Neural Information Processing Systems **32** (2019)
5. Bertinetto, L., Henriques, J.F., Torr, P.H., Vedaldi, A.: Meta-learning with differentiable closed-form solvers. arXiv preprint arXiv:1805.08136 (2018)
6. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: 2008 IEEE conference on computer vision and pattern recognition. pp. 1–8. IEEE (2008)
7. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
8. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9650–9660 (2021)
9. Chen, H., Li, H., Li, Y., Chen, C.: Shaping visual representations with attributes for few-shot learning. arXiv preprint arXiv:2112.06398 (2021)
10. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
11. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9640–9649 (2021)
12. Chen, Y., Liu, Z., Xu, H., Darrell, T., Wang, X.: Meta-baseline: exploring simple meta-learning for few-shot learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9062–9071 (2021)
13. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 (2020)
14. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
15. Doersch, C., Gupta, A., Zisserman, A.: Crosstransformers: spatially-aware few-shot transfer. Advances in Neural Information Processing Systems **33**, 21981–21993 (2020)
16. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
17. El-Nouby, A., Izacard, G., Touvron, H., Laptev, I., Jegou, H., Grave, E.: Are large-scale datasets necessary for self-supervised pre-training? arXiv preprint arXiv:2112.10740 (2021)

18. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning. pp. 1126–1135. PMLR (2017)
19. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International conference on machine learning. pp. 1180–1189. PMLR (2015)
20. Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P., Cord, M.: Boosting few-shot visual learning with self-supervision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8059–8068 (2019)
21. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377 (2021)
22. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
23. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 3304–3311. IEEE (2010)
24. Kang, D., Kwon, H., Min, J., Cho, M.: Relational embedding for few-shot classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8822–8833 (2021)
25. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
26. Lee, K., Maji, S., Ravichandran, A., Soatto, S.: Meta-learning with differentiable convex optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10657–10665 (2019)
27. Li, W.H., Liu, X., Bilen, H.: Improving task adaptation for cross-domain few-shot learning. arXiv preprint arXiv:2107.00358 (2021)
28. Li, W., Wang, L., Xu, J., Huo, J., Gao, Y., Luo, J.: Revisiting local descriptor based image-to-class measure for few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7260–7268 (2019)
29. Lin, R., Xiao, J., Fan, J.: Nextvlad: An efficient neural network to aggregate frame-level features for large-scale video classification. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops. pp. 0–0 (2018)
30. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
31. Liu, L., Hamilton, W., Long, G., Jiang, J., Larochelle, H.: A universal representation transformer layer for few-shot image classification. arXiv preprint arXiv:2006.11702 (2020)
32. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586 (2021)
33. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
34. Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al.: Swin transformer v2: Scaling up capacity and resolution. arXiv preprint arXiv:2111.09883 (2021)
35. 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 Applications of Computer Vision. pp. 2218–2227 (2020)
36. Müller, R., Kornblith, S., Hinton, G.E.: When does label smoothing help? Advances in neural information processing systems **32** (2019)
37. 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)
38. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2009)
39. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding with unsupervised learning (2018)
40. Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J.B., Larochelle, H., Zemel, R.S.: Meta-learning for semi-supervised few-shot classification. arXiv preprint arXiv:1803.00676 (2018)
41. 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)
42. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. Advances in neural information processing systems **30** (2017)
43. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
44. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1199–1208 (2018)
45. Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J.B., Isola, P.: Rethinking few-shot image classification: a good embedding is all you need? In: European Conference on Computer Vision. pp. 266–282. Springer (2020)
46. Triantafillou, E., Zhu, T., Dumoulin, V., Lamblin, P., Evci, U., Xu, K., Goroshin, R., Gelada, C., Swersky, K., Manzagol, P.A., et al.: Meta-dataset: A dataset of datasets for learning to learn from few examples. arXiv preprint arXiv:1903.03096 (2019)
47. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
48. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. Advances in neural information processing systems **29** (2016)
49. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
50. Wang, M., Deng, W.: Deep visual domain adaptation: A survey. Neurocomputing **312**, 135–153 (2018)
51. Wang, Y.X., Girshick, R., Hebert, M., Hariharan, B.: Low-shot learning from imaginary data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7278–7286 (2018)
52. Wei, C., Fan, H., Xie, S., Wu, C.Y., Yuille, A., Feichtenhofer, C.: Masked feature prediction for self-supervised visual pre-training. arXiv preprint arXiv:2112.09133 (2021)
53. Yang, S., Liu, L., Xu, M.: Free lunch for few-shot learning: Distribution calibration. arXiv preprint arXiv:2101.06395 (2021)

54. Zhang, B., Li, X., Ye, Y., Huang, Z., Zhang, L.: Prototype completion with primitive knowledge for few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3754–3762 (2021)
55. Zhang, C., Cai, Y., Lin, G., Shen, C.: 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. pp. 12203–12213 (2020)
56. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)