Learning a Few-shot Embedding Model with Contrastive Learning

Chen Liu* [1], **Yanwei Fu*** [1], **Chengming Xu** [1], **Siqian Yang**[2], **Jilin Li** [2], **Chengjie Wang** [2], **Li Zhang**† [1]

[1] School of Data Science, and MOE Frontiers Center for Brain Science, Fudan University
[2] YouTu Lab Tencent
{chenliu18, yanweifu, cmxu18, lizhangfd}@fudan.edu.cn,
{seasonyang, jerolinli, jasoncjwang}@tencent.com

Abstract

Few-shot learning (FSL) aims to recognize target classes by adapting the prior knowledge learned from source classes. Such knowledge usually resides in a deep embedding model for a general matching purpose of the support and query image pairs. The objective of this paper is to repurpose the contrastive learning for such matching to learn a few-shot embedding model. We make the following contributions: (i) We investigate the contrastive learning with Noise Contrastive Estimation (NCE) in a supervised manner for training a few-shot embedding model; (ii) We propose a novel contrastive training scheme dubbed infoPatch, exploiting the patch-wise relationship to substantially improve the popular infoNCE; (iii) We show that the embedding learned by the proposed infoPatch is more effective; (iv) Our model is thoroughly evaluated on few-shot recognition task; and demonstrates state-of-the-art results on *mini*ImageNet and appealing performance on *tiered*ImageNet, Fewshot-CIFAR100 (FC-100).

Introduction

Humans are born with the ability of *few-shot recognition*, *i.e.*, learning from one or a few examples. For example, a child finds no problem to recognize the "rhinoceros" by only taking a glance at it from the TV. However, currently most successful deep learning based vision recognition systems (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016, 2017) still highly rely on an avalanche of labeled training data and many iterations to train their large portion of parameters. Most importantly, these systems have difficulty adapting the learned knowledge to target categories. This severely limits their scalability to open-ended learning of the long tail categories in the real-world.

Inspired by the few-shot learning ability of humans, there has been a recent resurgence of interest in one/few-shot learning (Finn, Abbeel, and Levine 2017; Snell, Swersky, and Zemel 2017; Sung et al. 2018; Rusu et al. 2018; Tseng et al. 2020). It aims to recognize target classes by adapting the prior 'knowledge' learned from source classes. Such knowledge usually resides in a deep embedding model for a general-purpose matching of the support and query image pairs. The embedding is normally learned with enough training instances on source classes and updated by a few training instances on target classes. To further address data scarcity on target classes, meta-learning is utilized to better learn the deep embedding, and thus improves its generalization ability. Particularly, the idea of *episode* (Snell, Swersky, and Zemel 2017) is utilized for FSL in meta-learning paradigm. Every episode should imitate each one-shot learning task: few train and test instances are sampled from several classes to train/test the embedding model; the sampled training set is fed to the learner to produce a classifier, and then the loss of classifiers is computed on the sampled test set. The promising methodology of solving FSL is learning to match queries with few-shot support examples via a deep convolution network followed by a linear classifier. Typically, such methods train networks with meta-learners either to learn a deep embedding space that coincides with a fixed metric, such as MatchingNet (Vinyals et al. 2016) and ProtoNet (Snell, Swersky, and Zemel 2017) or to implicitly learn a metric and classify the new class data with the binary classifiers, such as RelationNet (Sung et al. 2018).

Despite previous efforts are made, the key challenge of a few-shot learning system still lies in eliminating the inductive bias from source classes to tailor its preference for hypotheses according to the few training instances from new target classes. Such a few-shot AI system has to deal with the poor generalisation of learned few-shot embedding model over target classes. On the other hand, the recent study of (Tian et al. 2020) suggests that the core of improving FSL also lies in improving the embedding learned. Particularly, it is very important for the embedding to map instances of different categories to different clusters. Furthermore, the embedding should not, in principle, learn the inductive bias of source classes by memorizing training data, as this might undermine the generalization performance of this embedding.

To this end, several new efforts are made in this paper in order to tackle the FSL on these several challenges. Specifically, we repurpose the contrastive learning to boost the performance of few-shot learning. As a prevailing and rising research topic, contrastive learning has been widely studied and utilized in several AI related research communities. For example, very impressive performance could be achieved on several downstream tasks (Chen et al. 2020b,a), if the model of good embedding is pre-trained on the unlabelled data.

*Equal contribution.
†Corresponding author.

For these methods, infoNCE (Oord, Li, and Vinyals 2018) is widely used. Notably, the key challenge of contrastive learning is to choose informative positive pairs and negative pairs (Khosla et al. 2020).

In this paper, contrastive learning is extended and utilized to the task of few-shot learning. Specifically, we propose the algorithm of constructing the positive and negative pairs by information of source classes. In one episode, we have support instances and query instances. For every query instance, we can construct positive and negative examples using all the support instances. To find more informative pairs for training good embedding, we present the strategy of generating hard examples. Intuitively, as human beings, we are able to rely only parts of image for recoginizing objects, even the other parts of images un-observable. Such an intuition is enforced to help build our contrastive learning algorithm in FSL. Typically, for the support images, they should contain enough information for matching; so we adopt the strategy of randomly blocking part of images. Accordingly, the query images are split into patches. Each patch is illustrated in Fig. 1; and those patches are employed to help few-shot recognition. Thus the model may learn the correspondence, even only part of the image is given.

We further make another contribution of removing the inductive bias of data in source classes. Critically, the inductive bias of source classes may inevitably introduce unexpected information or correlation between instances and classes. For instance, if the images of horses are highly correlated with grass, the model learned on such data may be inclined to relate to grass those target images visually similar to the horse images. We alleviate this issue by mixing patches from different pictures to enforce the embedding to learn more disentangled information.

The contributions of this work are as follows: (i) We investigate the contrastive learning with Noise Contrastive Estimation (NCE) in a supervised manner for training a few-shot embedding model; (ii) We propose a novel contrastive training scheme called infoPatch exploiting the patch-wise relationship to substantially improve the popular infoNCE. (iii) We show that the embedding learned by the proposed infoPatch is more effective. (iv) Extensive experiments show that our simple approach allow us to establish competitive results on three widely-used few-shot recognition benchmarks including *mini*ImageNet, *tiered*ImageNet and Fewshot-CIFAR100.

Related Work

Few-shot Learning Few-shot learning aims to recognize instance from target categories with few labelled samples. It demands the efficient few-shot algorithms for many practical applications, such as, classification (Fei-Fei, Fergus, and Perona 2006; Wang et al. 2020b,a), segmentation (Wang et al. 2019; Rakelly et al. 2018), generation (Liu et al. 2019) and localisation (Wertheimer and Hariharan 2019). Prior works can be roughly cast into two categories.

Optimization based approaches including MAML (Finn, Abbeel, and Levine 2017), Reptile (Nichol, Achiam, and Schulman 2018), LEO (Rusu et al. 2018) and metric learning based approaches such as ProtoNet (Snell, Swersky, and Zemel 2017), RelationNet (Sung et al. 2018), TADAM (Oreshkin, Rodriguez, and Lacoste 2018) and MatchingNet (Vinyals et al. 2016).

Metric learning based approaches attempt to learn a good embedding and an appropriate comparison metric. CAN (Hou et al. 2019) finds that the attentions are often misaligned between support and query images, a cross attention module is then used to alleviate the problem. In consideration of input variety, Cross Domain (Tseng et al. 2020) transforms the feature through an input dependent affine transformation layer. FEAT (Ye et al. 2020) combines FSL with transformer self-attention mechanism and achieves decent performance. (Wang et al. 2018a) proposes that by using triplet loss the performance of metric learning method can be improved. (Gidaris et al. 2019) adds extra self supervised tasks to improve generalization performance. DeepEMD (Zhang et al. 2020) attempts to import a new metric to solve the problem

Contrastive Learning Nowadays contrastive learning is widely used in unsupervised learning. DeepInfomax formalizes this problem in a view of mutual information. MoCo (Chen et al. 2020b) utilizes a memory bank and some implementation tricks to achieve good performance. SimClr (Chen et al. 2020a) improves contrastive learning by using larger batch size and data augmentation. CMC (Tian, Krishnan, and Isola 2019) attempts to combine information from different views. Currently (Khosla et al. 2020) suggests that infoNCE has better performance than cross entropy on supervised classification. Contrastive learning is also imported to other area such as image translation (Park et al. 2020). In (Park et al. 2020), the authors propose using the contrastive learning between the patches of target image and source images. Inspired by this, we tailor a novel constrastive learning with significant distinctive implementations in few-shot learning scenarios.

Data Augmentation Data augmentation is an important area in deep learning. With proper data augmentation (Zhang et al. 2017; Yun et al. 2019; Hendrycks et al. 2019), the performance of deep network can be improved significantly. For instance mixup (Zhang et al. 2017) can improve the classification performance on several widely used dataset. Following mixup (Zhang et al. 2017), manifold mixup (Verma et al. 2019) tries to mix the feature instead of input images. Cutout (DeVries and Taylor 2017) removes the part of the input images during training. Cutmix (Yun et al. 2019) improves them via exchanging patch with random size and uses a mixed label similar to mixup. Augmix (Hendrycks et al. 2019) combines several augmented input images with random sampled weights. By extending Cutmix (Yun et al. 2019), we present the PatchMix augmentation, the bespoken algorithm to better remove inductive bias and improve FSL. In (Summers and Dinneen 2019), the author provides an analysis of several variants of mixup (Zhang et al. 2017).

FSL and Data Augmentation Several FSL works put emphasis on data augmentation recently. Image Hallucination (Wang et al. 2018b) employs a generator to synthesise hallucinated images to enlarge the support set. IDeMe-Net (Chen et al. 2019c) samples a gallery image pool, most similar images are picked from the pool for data augmentation. Several regular augmentations are studied in (Chen et al.

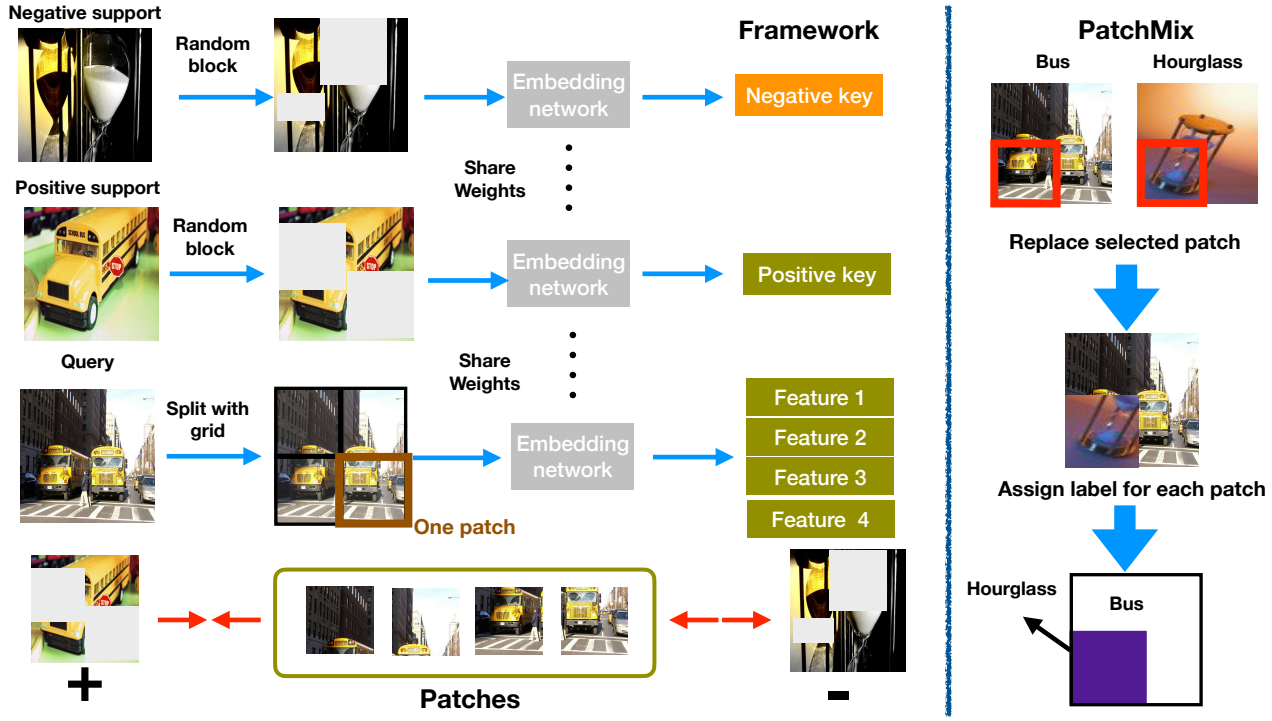

Figure 1: Our infoPatch is illustrated in this figure. The left part is the framework of our method. We try to use hard sample for contrastive learning. The definition of patch is shown with the grid. The right part shows the process of our PatchMix.

2019b). (Mangla et al. 2020) adds manifold mixup (Verma et al. 2019) to enhance the model embedding.

Method

Problem Definition

In this section, we introduce the problem of few-shot recognition. X_{train}, X_{val} and X_{test} denote the train, validation and test set respectively. The label sets are Y_{train}, Y_{val} and Y_{test}. The whole train, validation and test set are defined as $D_{train} = \{X_{train}, Y_{train}\}$, $D_{val} = \{X_{val}, Y_{val}\}$ and $D_{test} = \{X_{test}, Y_{test}\}$. We denote the categories of train set, validation set and test set as C_{train}, C_{val} and C_{test}.

For FSL, it is slightly different from common supervised learning. The categories of train set and test set are totally different, i.e., $C_{train} \cap C_{test} = \varnothing$. The goal of FSL is to recognize samples for new categories. In common, we need some labelled samples from new categories, called **support set**. Those samples staying to be classified are defined as **query set**. Images from support set are named as support images, similar definition for query images. A standard way to formalize this setting uses **way** and **shot**. **way** means the number of new categories during one test process. **shot** represents the number of support images for each category; here we suppose that we have same number of support images for each category. So we commonly call the FSL setting as N-**way**, k-**shot**. We focus on two mainstream settings 5-**way**, 1-**shot** and 5-**way**, 5**shot**. Additionally we denote the number of query images for each category as n_q.

Naïve Baseline

Recently, some works focus on rebuilding the baseline using supervised pretraining (Liu et al. 2020b; Dhillon et al. 2019). As supposed in their works, supervised pretraining shall achieve a very competitive FSL performance. Particularly, these methods usually train the network with a classification layer with a cross entropy loss on source classes. The network serves as a feature extractor on target classes; and nearest neighbour classifier is employed to classify the examples from target classes. Due to its simplicity, we adopt it as the naïve baseline.

Overview of InfoPatch To improve the naïve baseline with more representative embedding, we propose a novel model named **infoPatch** including two components. One is a contrastive learning scheme which modifies infoNCE loss into a few-shot manner and utilizes augmentation methods to mine hard samples. The other is a data augmentation technique, called PatchMix, which aims to alleviate the inductive bias introduced in the training process of few-shot learning.

Episodic Contrastive Learning

Before fully developing our model, we clarify some notations and definitions. In FSL, we define the **episode** as one sample of data that is composed of $N \times k$ support data and $N \times n_q$ query data. The query instance and support instance are denoted by x^q and x^s. Their labels are denoted as y^q and y^s.

We denote Φ as the embedding network such as ResNet12 (Oreshkin, Rodriguez, and Lacoste 2018). For convenience, we define the tensor shape of input and out

for the embedding network as $C_{in} \times H_{in} \times W_{in}$ and $C_{out} \times H_{out} \times W_{out}$. The training and testing process both utilise a N-**way**, k-**shot** setting for illustration. The output feature of embedding network Φ is denoted as f. f^q and f^s stand for output feature of query and support. For contrastive learning, normalized features are required for better comparison. In our paper, f^q and f^s are normalized by default. We normalize the output feature following (Chen et al. 2020b).

Training Phase Follow the idea of contrastive learning, we construct contrastive pair for every query instance. This construction way is in accordance with testing phase. The constrastive pairs are constructed using support features. For every query instance, we have its label. So for every query instance x_i^q, we regard support instance with same label as its positive pair. Negative pairs are those with different labels. For both query and support instances, we use the same embedding network Φ.

The infoNCE for one query instance x_i^q can be written as:

$$L_i = -log \frac{\sum_{y_j^s = y_i^q} e^{f_i^q f_j^s}}{\sum_{y_j^s = y_i^q} e^{f_i^q f_j^s} + \sum_{y_k^s \neq y_i^q} e^{f_i^q f_k^s}} \tag{1}$$

Here $f_i^q f_j^s$ means the inner product of the two feature vectors. For training one episode, the whole loss is the mean over all query samples as $L = \sum_{i=1}^{N \times n_q} L_i$. In our work, we combine supervised loss in naive baseline and infoNCE together during training. We set the weights for supervised loss and infoNCE loss as 1 and 0.5.

Testing Phase For testing, we have labelled support samples and unlabelled query samples. The goal is to predict query samples. For each query sample, we calculate the feature inner product between all the support samples. Here the network Φ is frozen. In detail, for each query sample x_i^q, we first get its feature f_i^q. We find the support instance with largest inner product with f_i^q.

$$j^* = \arg\max_j f_i^q f_j^s \tag{2}$$

Then we assign the prediction as $\hat{y}_i^q = y_{j^*}^s$.

Construct Hard Samples

As illustrated in CMC (Tian, Krishnan, and Isola 2019), one key point of contrastive learning lies in finding hard samples. The good way of finding hard samples forces the model to learn more useful information. To recognize an instance, humans do not always need to see the whole picture. Under most circumstances, part of the picture is enough. It can be similar for neural networks. We believe that using part of the picture can also add to the generalization ability. Meanwhile giving part of them can make the model learn more useful information. So we suppose that this can be a good way to construct hard samples.

Following this idea, we suggest that during training phase we should modify the input. During episode training, support images and query images act different roles. So we choose different modifications for them.

For support images, they are regarded as matching template. So we try to keep them intact. For dropping part of its information, we apply random masks to the support images. This process is illustrated in Fig.1. Using this modification, the support images are harder to recognize than original ones. We call this modification random block.

For query images, we try to match them with support images, we hope that we can get a correct match even if we only have part of the query images. We can split the input query instance into several **patches** using grids. The definition of **patch** is one unit of the grids as shown in Fig.1. For convenience, we suggest that we have a $W \times H$ patches. Now we fed them into the embedding network, and finally get $W \times H$ features. For query sample x_i^q, we denote it wh_{th} vector as f_{iwh}^q. Each of them has part of the information of this query sample. In order to fully learn the correlation between pixels, we still input the whole image into the embedding network. Then we can get the $W \times H$ output features for different patches of one query instance. The loss function should be modified slightly as follows.

$$L_{iwh} = -log \frac{\sum_{y_j^s = y_i^q} e^{f_{iwh}^q f_j^s}}{\sum_{y_j^s = y_{iwh}^q} e^{f_i^q f_j^s} + \sum_{y_k^s \neq y_i^q} e^{f_{iwh}^q f_k^s}} \tag{3}$$

The whole loss is $L = \sum_{i=1}^{N \times n_q} \sum_{w=1}^{W} \sum_{h=1}^{H} L_{iwh}$ This alternation is only used for training, the testing phase is kept the same.

Enhancing Contrastive Learning via PatchMix

For FSL, we demand a generalization on target classes. During training phase, the data bias of the source classes may do harm to the generalization. The data bias can be caused by learning incorrect correlation between pixels. For example, the background of some specific classes may be similar in color or texture. The embedding network may just memorize this property. To alleviate the issue, we suggest that we can mix some patches. For instance, after we mixing the patches, the images have more diversity, some simple correlations can not work any more. Then the network can learning some real rules.

For implementation, we mix the image patches randomly. Following Cutmix (Yun et al. 2019), we use a similar rule. The PatchMix operation is performed inner one episode. To avoid importing too much noise, we only conduct PatchMix for query samples. In detail, for every query instance x_i^q, we sample a different instance x_k^q from samples in this episode. Then we randomly select a box (w_1, h_1, w_2, h_2). Here w_1, h_1 denotes the left upper point and w_2, h_2 stands for the right lower point. The way to sample the random box is similar to Cutmix (Yun et al. 2019). We simply replace the patch of x_i^q by patch of x_k^q as

$$x_i^q[w1 : w2, h1 : h2] = x_k^q[w1 : w2, h1 : h2] \tag{4}$$

The difference between PatchMix and Cutmix lies in the label after mix. For Cutmix, it uses a mixed label for training the cross entropy loss. As metioned before, we use patches for contrastive learning, so we assign deterministic label for every patch. Note that we mix patch just for avoiding simple correlations, every patch keeps its original label. The instances after PatchMix is then fed into the embedding network. The loss is the same as previous section.

Model	Backbone	*mini*ImageNet		*tiered*ImageNet	
		1-shot	5-shot	1-shot	5-shot
ProtoNet (Snell, Swersky, and Zemel 2017)	Conv4	44.42±0.84	64.24±0.72	53.31±0.89	72.69±0.74
MatchingNet (Vinyals et al. 2016)		48.14±0.78	63.48±0.66	—	—
RelationNet (Sung et al. 2018)		49.31±0.85	66.60±0.69	54.48±0.93	71.32±0.78
MAML (Finn, Abbeel, and Levine 2017)		46.47±0.82	62.71±0.71	51.67±1.81	70.30±1.75
LEO (Rusu et al. 2018)	WRN-28	61.76±0.08	77.59±0.12	66.33±0.05	81.44±0.09
PPA (Qiao et al. 2018)		59.60±0.41	73.74±0.19	—	—
wDAE (Gidaris and Komodakis 2019)		61.07±0.15	76.75±0.11	68.18±0.16	83.09±0.12
CC+rot (Gidaris et al. 2019)		62.93±0.45	79.87±0.33	70.53±0.51	84.98±0.36
ProtoNet (Chen et al. 2019a)	Res-10	51.98±0.84	72.64±0.64	—	—
MatchingNet (Chen et al. 2019a)		54.49±0.81	68.82±0.65	—	—
RelationNet (Chen et al. 2019a)		52.19±0.83	70.20±0.66	—	—
MAML (Chen et al. 2019a)		51.98±0.84	66.62±0.83	—	—
Cross Domain (Tseng et al. 2020)		66.32±0.80	81.98±0.55	—	—
TapNet (Yoon, Seo, and Moon 2019)	Res-12	61.65±0.15	76.36±0.10	—	—
MetaOptNet (Lee et al. 2019)		62.64±0.61	78.63±0.46	65.99±0.72	81.56±0.53
CAN (Hou et al. 2019)		63.85±0.48	79.44±0.34	69.89±0.51	84.23±0.37
FEAT (Fei et al. 2020)		66.78±0.20	82.05±0.14	70.80±0.23	84.79±0.16
DeepEMD (Zhang et al. 2020)		65.91±0.82	82.41±0.56	71.16±0.87	**86.03±0.58**
Negative Margin (Liu et al. 2020a)		63.85±0.81	81.57±0.56	—	—
Rethink-Distill (Tian et al. 2020)		64.82±0.60	82.14±0.43	**71.52±0.69**	**86.03±0.49**
infoPatch		**67.67±0.45**	**82.44±0.31**	71.51±0.52	85.44±0.35

Table 1: 5-way few-shot accuracies with 95% confidence interval on *mini*ImageNet and *tiered*ImageNet. All results of competitors are from the original papers.

Experiments

Dataset and Setting

To validate our method, we conduct experiments on several widely used datasets. *mini*ImageNet (Vinyals et al. 2016) is a sub-dataset from ImageNet (Russakovsky et al. 2015). It has 100 categories in all, each category has 600 instances. These categories are split into train, val and test with 64, 16 and 20 classes respectively. The partition follows the instruction of (Ravi and Larochelle 2017). *tiered*ImageNet is also sampled from ImageNet (Russakovsky et al. 2015). It is made up of 779,165 images from 608 categories. They are separated into 351 classes for training, 97 for validation and 160 for testing as suggested in (Ren et al. 2018). Fewshot-CIFAR100 (*FC100*) dataset (Oreshkin, Rodriguez, and Lacoste 2018) is a subset of CIFAR-100. A common split is 60, 20 and categories for train, val and test set.

Images of *tiered*ImageNet and *mini*ImageNet are firstly resized to 84×84 during training and testing process. Images of *FC100* are resized to 32×32. For training process, random horizontal flip and random crop are utilized as common data augmentation as used in (Hou et al. 2019).

Implementation Details. ResNet12 is our selected model structure, the details follow the one proposed in TADAM (Oreshkin, Rodriguez, and Lacoste 2018). We use he-normal (He et al. 2015) to initialize the model. Stochastic Gradient Descent(SGD) (Bottou 2010) is taken as our optimizer. The initial learning rate is 0.1. For *mini*ImageNet, we decrease the learning rate at $12,000$-th, $14,000$-th and $16,000$-th episode. For *tiered*ImageNet, the learning rate is halved at every 24,000 episodes. For all the experiments, we test the model for 2000 episodes. 4 episodes are picked for every batch during training.

Metric for Comparison. We conduct experiments on two settings: 5-way 1-shot and 5-way 5-shot. We report mean accuracy as well as the 95% confidence interval for comparison with other methods. For ablation study and further discussions, only mean accuracy is reported.

Comparison with State-of-the-art

Competitors. To testify how our model performs, several previous methods are selected for comparison. For instance ProtoNet (Snell, Swersky, and Zemel 2017), MAML (Finn, Abbeel, and Levine 2017), CAN (Hou et al. 2019), FEAT (Fei et al. 2020), Cross Domain (Tseng et al. 2020) and so on. These methods are either classical methods in FSL or methods with best reported results.

Discussion The results are reported in Tab. 1. Compared to other method with complex structure or larger network(WRN28), we achieve conspicuous increment, about 1% compared with FEAT (Fei et al. 2020). Due to no extra structure adds to the model, we have more clear inference logic compared to some other methods such as CAN (Hou et al. 2019).

Results on FC100 are shown in Tab. 2, our model achieves competitive performance among them.

Model	FC100 accuracies	
	5-way 1-shot	5-way 5-shot
MAML	38.1±1.7	50.4±1.0
MAML++	38.7±0.4	52.9±0.4
T-NAS++	40.4±1.2	54.6±0.9
TADAM	40.1±0.4	56.1±0.4
ProtoNet	37.5±0.6	52.5±0.6
MetaOptNet	41.1±0.6	55.5±0.6
DC	42.0±0.2	57.1±0.2
DeepEMD	**46.5±0.8**	**63.2±0.7**
Rethink-Distill	44.6±0.7	60.1±0.6
infoPatch	43.8±0.4	58.0±0.4

Table 2: 5-way few-shot accuracies with 95% confidence interval on FC100. All results of competitors are from the original papers

Model	k	
	1	5
R-Net	52.78	68.11
R-Net + P-mix	53.50	68.67
CAN	63.85	79.44
CAN + P-mix	64.65	79.86

(a)

Type	k	
	1	5
Ind-mix	67.67	82.44
S-mix	67.53	81.94
E-mix	67.48	82.06

(b)

Table 3: R-Net: RelationNet, P-mix: PatchMix. Ind-mix: independent mix, S-mix: share mix, E-mix: exchange mix. Table(a) shows combination of PatchMix and other methods(RelationNet and CAN). Table(b) shows the ablation study on different implementations of PatchMix.

Ablation Study

Analysis of Our Method For our method, we have different parts:infoNCE, hard sample, PatchMix. As shown in Tab. 5, each part contributes to the improvement. For this analysis, we only use *mini*ImageNet. Among them, we find that the each part has significant contribution. With our infoNCE, we can improve more than 2% compared with the baseline. Using the hard sample proposed by us, the model has better generalization ability, and the performance comes to 66.8% for 1-shot classification. For PatchMix, we find that it can improve the model further by about 1%.

Ablation on Grid Size During constructing the hard examples, we have to define the grid for patches. For convenience, we only conduct analysis experiments on *mini*ImageNet. We choose three kinds of grid size $1 \times 1, 6 \times 6$ and 11×11. For 1×1, we use the whole image for contrastive learning. As shown in Tab. 4(b), using a large grid size leads to better results. We do not try larger grid size. For we have an input size of 84×84, larger grid size may lead to larger noise in patches. For a moderate grid size, we can find hard samples, which can improve the performance.

Augment	k	
	1	5
mixup	66.64	80.99
augmix	66.90	81.27
cutmix	66.34	81.43
IDeMeNet	66.59	81.12
M-mixup	66.92	81.41
PatchMix	67.67	82.44

(a)

grid size	k	
	1	5
1× 1	64.23	79.17
6×6	66.19	81.27
11× 11	66.80	81.35

(b)

Table 4: M-mixup: Manifold Mixup. Table(a) contains the comparisons between other augmentation methods. Table(b) shows the results of different grid size. Note that we do not include PatchMix in this experiment for 6×6 and 11×11. We find that using the size of 11×11 is a good choice for our setting.

Model	*mini*ImageNet	
	5-way 1-shot	5-way 5-shot
Baseline	61.69	78.31
+ infoNCE	64.23	79.17
+ hard sample	66.80	81.35
+ PatchMix	67.67	82.44

Table 5: Ablation study on our model, we can find that each part of our model has important contribution.

Effectiveness on PatchMix To verify the effectiveness of our proposed method, we conduct following two experiments. We plug our PatchMix into other existing few-shot learning methods, i.e., RelationNet and CAN. Note that we modify the output of RelationNet to be a rather than a scalar to make it compatible with patch-wise loss. Results in Tab. 3(a)are similar to those in Tab. 5. The first experiment illustrates that our augmentation method can be applied to other FSL methods directly and boost their performance. It proves that our method is a general method that can be used in FSL widely.

To further validate our PatchMix, we pick several other data augmentation method for comparison. Experiments are conducted on *mini*ImageNet with the same setting except for the detailed data augmentation method. We add Augmix (Hendrycks et al. 2019), Cutmix (Yun et al. 2019) to our baseline method. Meanwhile, we utilize the augmentation method from manifold mixup (Mangla et al. 2020) and IDeMeNet (Chen et al. 2019b). We report the results on Tab. 4(a) It is obvious that our PatchMix gives the best results.

On the Implementation of PatchMix We implement the PatchMix by exchanging the patch between samples. In this session, we also discuss the detailed implementation. For our default implementation, we conduct the PatchMix inner every episode. We call this kind of implementation as independent mix. Currently, some works propose to modify the sampling strategies. For instance, we can sample two episode that have the same categories. The images of these two episodes are

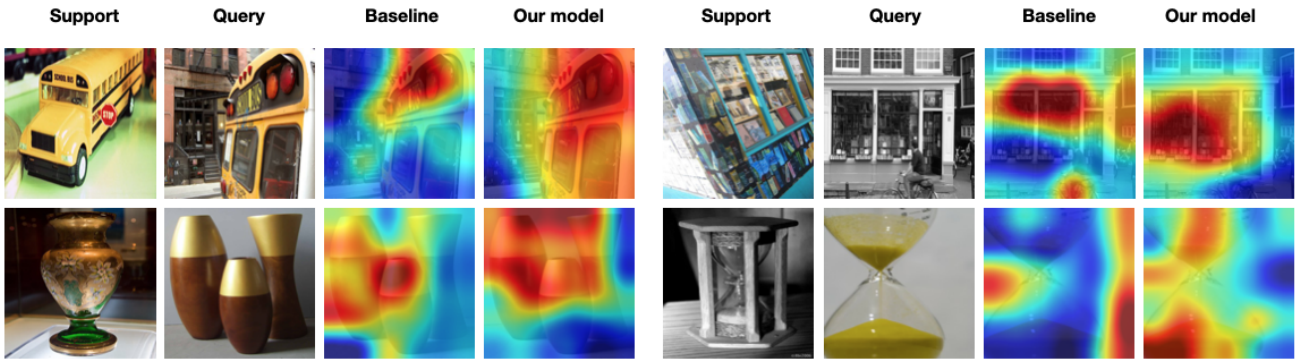

Figure 2: Images and the heatmaps for spatial correspondence are shown. We use the feature of support generated by the network to calculate inner product with features of query images. We visualize the inner product in form of heatmap. We can find that our model locates the object more precisely. This part is done with images from **target classes**.

totally different. The two episodes are similar under this sampling strategy. So we try two variants of implementation. The first is called share mix. For share mix, we conduct PatchMix inner the two episodes. The other one is named as exchange mix. It conduct PatchMix by using samples from similar episode instead of the episode they belong to. By observing the results of Tab. 3(b), we can find that PatchMix is robust in terms of mix strategies.

Visualizations

The effectiveness of our method is conspicuous. We explore the mechanism of the improvement with visualizations in this section.

We firstly visualize the embedding by using tSNE plot. In detail, we sample one episode from target classes of *mini*ImageNet, and feed it into baseline model as well as our full model. The embedding is visualized in Fig. 3. From fig. 3, we can observe that the cluster generated by our method is more compact than that of baseline method.

Besides, we verify whether we can recognize the images by using part of the information by visualizing the spatial correspondence. Similarly, we sample one episode from target classes of *mini*ImageNet. We use the feature of support images to calculate inner product of each patch of the query images. Heatmap score is visualizd in Fig. 2. From Fig. 2, we can observe that our method outperforms the baseline method in terms of spatial relationship. Our model covers more accurately and completely of the foreground. It can also be viewed as proof for better representation.

Conclusion

In this paper, we have shown that contrastive learning with Noise Contrastive Estimation (NCE) in a supervised manner can be used to train a deep embedding model for few-shot recognition. Based on this observation, we have proposed a novel contrastive training scheme called infoPatch, exploiting the patch-wise relationship to substantially improve the popular infoNCE. We have shown that the embedding learned

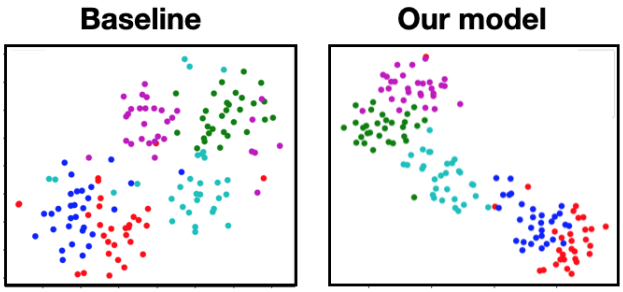

Figure 3: We visualize the tSNE plot of some samples from target classes. The left one is visualization for baseline and the right for our model. It is clear that our model cluster the samples better. Here different colors represents different categories.

by the proposed infoPatch is more effective. We thoroughly evaluate our method on the few-shot recognition task and demonstrates state-of-the-art results on *mini*ImageNet and appealing performance on *tiered*ImageNet, Fewshot-CIFAR100 (FC-100).

Acknowledgements

This work was supported in part by NSFC Project (U62076067), Science and Technology Commission of Shanghai Municipality Projects (19511120700), Shanghai Municipal Science and Technology Major Project (2018SHZDZX01), and Shanghai Research and Innovation Functional Program (17DZ2260900).

References

Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, 177–186. Springer.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A simple framework for contrastive learning of visual repre-

sentations. In *International conference on machine learning*, 1597–1607. PMLR.

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

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020b. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297* .

Chen, Z.; Fu, Y.; Chen, K.; and Jiang, Y.-G. 2019b. Image block augmentation for one-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3379–3386.

Chen, Z.; Fu, Y.; Wang, Y.-X.; Ma, L.; Liu, W.; and Hebert, M. 2019c. Image deformation meta-networks for one-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8680–8689.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552* .

Dhillon, G. S.; Chaudhari, P.; Ravichandran, A.; and Soatto, S. 2019. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729* .

Fei, N.; Lu, Z.; Gao, Y.; Tian, J.; Xiang, T.; and Wen, J.-R. 2020. Meta-learning across meta-tasks for few-shot learning. *arXiv preprint arXiv:2002.04274* .

Fei-Fei, L.; Fergus, R.; and Perona, P. 2006. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence* 28(4): 594–611.

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.

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. 2019. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21–30.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

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.

Hendrycks, D.; Mu, N.; Cubuk, E. D.; Zoph, B.; Gilmer, J.; and Lakshminarayanan, B. 2019. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781* .

Hou, R.; Chang, H.; Ma, B.; Shan, S.; and Chen, X. 2019. Cross attention network for few-shot classification. *arXiv preprint arXiv:1910.07677* .

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362* .

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25: 1097–1105.

Lee, K.; Maji, S.; Ravichandran, A.; and Soatto, S. 2019. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10657–10665.

Liu, B.; Cao, Y.; Lin, Y.; Li, Q.; Zhang, Z.; Long, M.; and Hu, H. 2020a. Negative margin matters: Understanding margin in few-shot classification. In *European Conference on Computer Vision*, 438–455. Springer.

Liu, C.; Xu, C.; Wang, Y.; Zhang, L.; and Fu, Y. 2020b. An Embarrassingly Simple Baseline to One-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 922–923.

Liu, M.-Y.; Huang, X.; Mallya, A.; Karras, T.; Aila, T.; Lehtinen, J.; and Kautz, J. 2019. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10551–10560.

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.

Nichol, A.; Achiam, J.; and Schulman, J. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999* .

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* .

Oreshkin, B. N.; Rodriguez, P.; and Lacoste, A. 2018. TADAM: task dependent adaptive metric for improved few-shot learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 719–729.

Park, T.; Efros, A. A.; Zhang, R.; and Zhu, J.-Y. 2020. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, 319–345. Springer.

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.

Rakelly, K.; Shelhamer, E.; Darrell, T.; Efros, A. A.; and Levine, S. 2018. Few-shot segmentation propagation with guided networks. *arXiv preprint arXiv:1806.07373* .

Ravi, S.; and Larochelle, H. 2017. Optimization as a model for few-shot learning. In *In International Conference on Learning Representations (ICLR)*.

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* .

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115(3): 211–252.

Rusu, A. A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; and Hadsell, R. 2018. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960* .

Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175* .

Summers, C.; and Dinneen, M. J. 2019. Improved mixed-example data augmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1262–1270. IEEE.

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.

Tian, Y.; Krishnan, D.; and Isola, P. 2019. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849* .

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? *arXiv preprint arXiv:2003.11539* .

Tseng, H.-Y.; Lee, H.-Y.; Huang, J.-B.; and Yang, M.-H. 2020. Cross-domain few-shot classification via learned feature-wise transformation. *arXiv preprint arXiv:2001.08735* .

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; and Bengio, Y. 2019. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, 6438–6447. PMLR.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2016. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080* .

Wang, K.; Liew, J. H.; Zou, Y.; Zhou, D.; and Feng, J. 2019. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9197–9206.

Wang, Y.; Wu, X.-M.; Li, Q.; Gu, J.; Xiang, W.; Zhang, L.; and Li, V. O. 2018a. Large margin meta-learning for few-shot classification. In *Proc. 2nd Workshop Meta-Learn. NeurIPS*, 1–8.

Wang, Y.; Xu, C.; Liu, C.; Zhang, L.; and Fu, Y. 2020a. Instance credibility inference for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12836–12845.

Wang, Y.; Zhang, L.; Yao, Y.; and Fu, Y. 2020b. How to trust unlabeled data? Instance Credibility Inference for Few-Shot Learning. *arXiv preprint arXiv:2007.08461* .

Wang, Y.-X.; Girshick, R.; Hebert, M.; and Hariharan, B. 2018b. Low-shot learning from imaginary data. In *CVPR*.

Wertheimer, D.; and Hariharan, B. 2019. Few-shot learning with localization in realistic settings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6558–6567.

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.

Yoon, S. W.; Seo, J.; and Moon, J. 2019. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *International Conference on Machine Learning*, 7115–7123. PMLR.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6023–6032.

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, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* .