Partial Is Better Than All: Revisiting Fine-tuning Strategy for Few-shot Learning

Zhiqiang Shen[1*], Zechun Liu[1,2*], Jie Qin[3], Marios Savvides[1] and Kwang-Ting Cheng[2]
[1]Carnegie Mellon University
[2]Hong Kong University of Science and Technology
[3]Inception Institute of Artificial Intelligence
{zhiqians,marioss}@andrew.cmu.edu; zliubq@connect.ust.hk; qinjiebuaa@gmail.com; timcheng@ust.hk

Abstract

The goal of few-shot learning is to learn a classifier that can recognize unseen classes from limited support data with labels. A common practice for this task is to train a model on the base set first and then transfer to novel classes through fine-tuning[1] or meta-learning. However, as the base classes have no overlap to the novel set, simply transferring whole knowledge from base data is not an optimal solution since some knowledge in the base model may be biased or even harmful to the novel class. In this paper, we propose to transfer partial knowledge by freezing or fine-tuning particular layer(s) in the base model. Specifically, layers will be imposed different learning rates if they are chosen to be fine-tuned, to control the extent of preserved transferability. To determine which layers to be recast and what values of learning rates for them, we introduce an evolutionary search based method that is efficient to simultaneously locate the target layers and determine their individual learning rates. We conduct extensive experiments on CUB and *mini*-ImageNet to demonstrate the effectiveness of our proposed method. It achieves the state-of-the-art performance on both meta-learning and non-meta based frameworks. Furthermore, we extend our method to the conventional *pre-training + fine-tuning* paradigm and obtain consistent improvement.

1. Introduction

Deep neural networks have shown enormous potential on understanding natural images (Krizhevsky, Sutskever, and Hinton 2012; Szegedy et al. 2015; Simonyan and Zisserman 2014; He et al. 2016; Huang et al. 2017) in the recent years. The learning ability of deep neural networks increases significantly with more labeled training data. However, annotating such data is expensive, time-consuming and laborious. Furthermore, some classes (e.g., in medical images) are naturally rare and hard to collect. The conventional training approaches for deep neural networks often fail to obtain good performance when the training data is insufficient. Considering that humans can easily learn from very few examples and even generalize to many different new images, it will be greatly helpful if the network can also learn to generalize to new classes with only a few labeled samples from unseen classes. Previous studies in this direction (i.e., few-shot learning) can be mainly divided into two categories. One is the meta-learning based methods (Snell, Swersky, and Zemel 2017; Vinyals et al. 2016; Finn, Abbeel, and Levine 2017) that model the few-shot learning process with samples belonging to the base classes, and optimize the model for the target novel classes. The other is the plain solution (non-meta and also called baseline method (Chen et al. 2019)) that trains feature extractor from abundant base class then directly predicts the weights of the classifier for the novel ones.

As the number of images in the support set of novel classes are extremely limited, directly training models from scratch on the support set is unstable and tends to be overfitting. Even utilizing the pre-trained parameters on base classes and

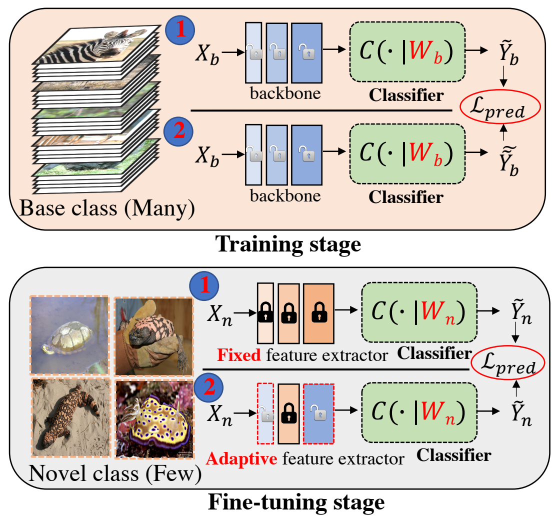


Figure 1: Illustration of the conventional procedure of pre-training and fine-tuning for few-shot learning. ① represents the standard transfer learning procedure which uses the pre-trained model as a feature extractor and the parameters are fixed during fine-tuning. ② is our proposed partial transfer strategy which can fine-tune the model trained on base data with the few novel class data. Fine-tuning with different learning rates on different layers can optimize the feature extractor to better fit the novel class and prevent the model from over-fitting on it, since the novel data has limited samples.

*indicates equal contribution.

[1]Here fine-tuning procedure is defined as transferring knowledge from base to novel data, i.e. learning to transfer in few-shot scenario.

fine-tuning all layers on the support set will still lead to poor performance due to the small proportion of target training data. A common practice utilized by either meta-based or simple baseline methods relies heavily on the pre-trained knowledge with the sufficient base classes, and then transfer the representation by freezing the backbone parameters and solely fine-tuning the last fully-connected layer or directly extracting features for distance computation on the support data, to prevent overfitting and improve generalization. However, as the base classes have no overlap with the novel ones, meaning that the representation and distribution required to recognize images are quite different between them, completely freezing the backbone network and simply transferring the whole knowledge will suffer from this discrepant domain issue, though currently the domain difference is not huge in the existing few-shot learning datasets.

To fundamentally overcome the aforementioned drawback, in this work, we propose to utilize a flexible way to transfer knowledge from base to novel classes. We introduce a partial transfer paradigm for the few-shot classification task, as shown in Figure 1. In our framework, we first pre-train a model on the base classes as previous methods did. Then, instead of transferring the learned representation by freezing the whole backbone network, we develop an efficient evolutionary search method to automatically determine which layer/layers need to be frozen and which will be fine-tuned on the support set (on novel class). During searching, the validation data will be commandeered as the ground-truth to monitor the performance of the searched strategy. Intuitively, our strategy can achieve a better trade-off of using knowledge from base and support data than previous approaches, meanwhile, our strategy can avoid incorporating biased or harmful knowledge from base classes into novel classes. Moreover, our method is orthogonal to meta-learning or non meta-based solutions, and thus can be seamlessly integrated with them. We perform extensive experiments on CUB200-2011 and *mini*-ImageNet datasets. Our results empirically show that the proposed method can favorably improve both of these two types of solutions. We further extend our method to the traditional *pre-training + fine-tuning* paradigm from ImageNet to CUB200-2011 and achieve consistent improvement, demonstrating the effectiveness and excellent expansibility of our proposed method.

In summary, our contributions are three-fold:

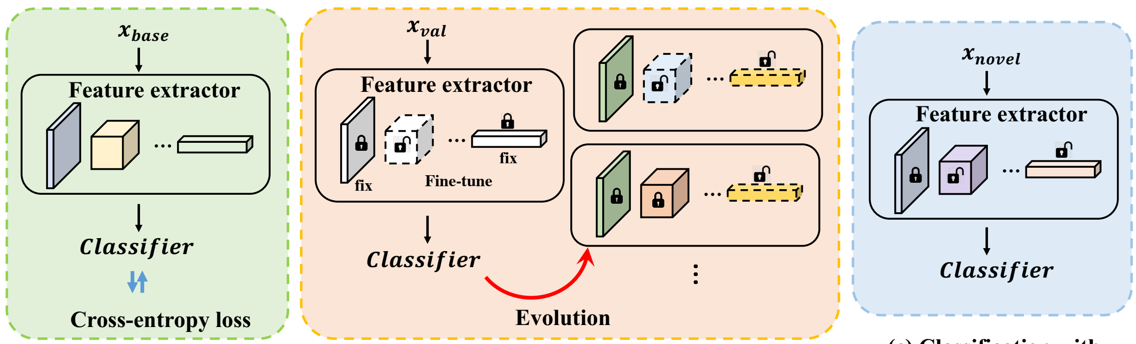
- We present Partial Transfer (P-Transfer) for the few-shot classification, a framework that enables to search transfer strategies on backbone for flexible fine-tuning. Intuitively, the conventional fixed transferring is a special case of our propose strategy when all layers are frozen. Also, to our best knowledge, this is the pioneer attempt that can achieve partial transfer with different learning rates on this challenging task.
- We introduce a layer-wise search space for fine-tuning from base classes to novel. It helps the searched transfer strategy obtain inspiring accuracies under limited searching complexity. For example, using one V100 GPU, our search algorithm only takes $\sim$6 hours with Conv6 backbone and one day with ResNet-12 backbone on average.
- Our resulting network, the P-Transfer model, outperforms the complete transfer and the hand-crafted transfer strategies by a remarkable margin. As the two baseline transfer strategies belong to our search space, thus ideally the better performance is guaranteed by our searching method. With the assistance of designing search space, we show the effectiveness of P-Transfer in different few-shot learning methods on various datasets. The searched strategy has consistently better performance and meaningful structural patterns.

2. Background

Few-shot learning is defined as given abundant labeled training examples in the base classes, the trained network can be generalized to classify the new classes with few labeled samples. Recently, few-shot learning, enabled by transferring knowledge from the base to novel data, has been increasingly important. Existing few-shot learning methods can mainly be categorized into meta-learning based methods and non-meta learning methods. Here we also review the searching based methods for few-shot learning in this section.

Meta-based few-shot learning. To tackle the data deficiency in few-shot learning, previous studies adopt meta-learning to learn the model or optimizer that can fast update the weights for adapting to the unseen tasks (Thrun and Pratt 2012; Andrychowicz et al. 2016; Ye et al. 2020; Tian et al. 2020; Kim, Kim, and Kim 2020). For example, MetaNetwork (Munkhdalai and Yu 2017) learned a meta-level knowledge for rapid generalization. Ravi and Larochelle (Ravi and Larochelle 2017) proposed to use the LSTM-based meta-learner model to learn the optimization algorithm. MAML (Finn, Abbeel, and Levine 2017; Antoniou, Edwards, and Storkey 2018) simplified the aforementioned MetaNetwork by only learning the initial learner parameters to achieve rapid adaptation w.r.t. those initial parameters and high generalizability to the new tasks. Furthermore, meta-learning methods are utilized to learn the similarity between two images. MatchingNet (Vinyals et al. 2016) proposed to map a small labeled support set to its label, and determine the class of an instance in the query set by finding its nearest labeled example. ProtoNet (Snell, Swersky, and Zemel 2017) further utilized class-wise mean and the Euclidean distance to generalize the MatchingNet from one-shot learning to few-shot learning. RelationNet (Sung et al. 2018) use CNN-based relation modules and Few-shot GNN (Garcia and Bruna 2017) employed graph neural networks to learn useful metrics.

Non-meta few-shot learning. Besides those meta-learning based methods, there are non-meta methods which utilize cosine similarity to predict the novel class classifier with weight generator (Gidaris and Komodakis 2018), directly set the weights based on the embedding layer's activations (Qi, Brown, and Lowe 2018) or use dense representations from image regions to calculate the distances (Zhang et al. 2020). Chen et al. (Chen et al. 2019) proposed to reduce intra-class variation along with the confine similarity and achieves competitive performance. Both the meta and non-meta methods used the fixed feature extractor trained from the based classes, which can hardly take the domain discrepancy between the base and novel classes into consideration. Instead of learning more advanced optimizers or classification metrics, we tackle the few-shot problem by discovering an meta knowledge transfer scheme through evolutionary search, which is



(a) Train the feature extractor (b) Evolutionary search for finetuning configuration (c) Classification with searched feature extractor

Figure 2: Our overall framework overview consists of three stages: (a) train a feature extractor from scratch on the base dataset; (b) apply evolutionary search to explore optimal combination of layers that requires fine-tuning on the validation dataset. Note that the blocks with dashed lines denote the fine-tuning layers in a specific evolution iteration; (c) use the best fine-tuning scheme discovered by the evolutionary search to fine-tune the selected layers on the support set of the novel dataset and inferring the final accuracy on the query set.

compatible with both meta and non-meta methods.

Neural architecture search for few-shot learning. Evolutionary algorithm has been adopted in the neural architecture search (NAS) to obtain the optimal neural architecture (Miikkulainen et al. 2019; Real et al. 2017; Xie and Yuille 2017; Liu et al. 2017; Real et al. 2019). Evolutionary-based NAS evolves within a given architecture search space and updates a population of genes (i.e., the operation choices in an architecture) to select the top gene for the final model. Recent study (Elsken et al. 2020) proposed to integrate NAS algorithm with gradient based meta-learning for few-shot learning task. Different from neural architecture search, our proposed P-Transfer utilizes evolutionary algorithm to seek the optimal fine-tuning scheme, instead of the network architecture. Meta-SGD (Li et al. 2017) and MAML++ (Antoniou, Edwards, and Storkey 2018) can also learn diverse learning rates for each layer in the networks, but they were mainly designed for MAML-like methods and only suitable for the meta-based scenarios. In contrast, our proposed method can completely turnoff the learning rate to zero and fix the weights in a layer, which is a more general design for the few-shot learning task.

3. Methodology

In this section, we start by introducing the problem definition of few-shot classification, then we present our whole framework, which consists of three steps: 1) train a base model on base class samples (left sub-illustration in Figure 2), 2) apply evolutionary search to explore optimal transfer strategy based on accuracy metric (middle sub-figure, curve arrow indicates looping), and 3) transfer base model to novel class with the searched strategy through partially fine-tuning. Lastly, we elaborate how to design our search space for transferring and present our search algorithm in detail.

3.1. Preliminary and Definition

In the few-shot classification task, given abundant labeled images $\mathbf{X}_b$ in base classes $\mathbf{L}_b$ and a small proportion of labeled images $\mathbf{X}_n$ in novel classes $\mathbf{L}_n$, $\mathbf{L}_b \cap \mathbf{L}_n = \emptyset$. Our goal is to train models for recognizing novel classes with the labeled large amount of base data and limited novel data. Considering an N-way K-shot few-shot task, where the support set on novel class has N classes with K labeled images and the query set contains the same N classes with Q unlabeled images in each class, the few-shot classification algorithms are required to learn classifiers for recognizing the $N \times Q$ images in the query set of N classes.

Our objective of P-Transfer aims to discover the best transfer learning scheme V_{lr}^{*}, such that, the network achieves maximal accuracy when fine-tuning under that scheme:

$$V_{lr}^{*} = \arg\max \mathcal{A}cc(W, V_{lr}), \tag{1}$$

where $V_{lr} = [v_1, v_2, ..., v_L]$ defines the layer-wise learning rate for fine-tuning the feature extractor, W is the network's parameters and L is the total number of layers.

3.2. Framework

As shown in Figure 2, our method consists of three steps: base class pre-training, evolutionary search, and partially transfer based on the searched strategy.

Step 1: Base class pre-training. Base class pre-training is the fundamental step of our whole pipeline. As shown in Figure 2 (a), for the simple baseline, we follow the common practice to train the model from scratch by minimizing a standard cross-entropy objective with the training samples in base classes. For the meta-learning pipeline, the meta-pretraining also follows the conventional strategy that a meta-learning classifier is conditioned on the base support set. More specifically, in the meta-pretraining stage, support set and query set on the base class are first sampled randomly from N classes, and then train the parameters to minimize the N-way prediction loss.

Step 2: Evolutionary search. The second step is to perform evolutionary search with different fine-tuning strategies to determine which layers will be fixed and others will be fine-tuned in the representation transfer stage. We also consider the above two circumstances: simple baseline through pre-training + fine-tuning, and meta-based methods. In these two scenarios the evolutionary searching operations are slightly

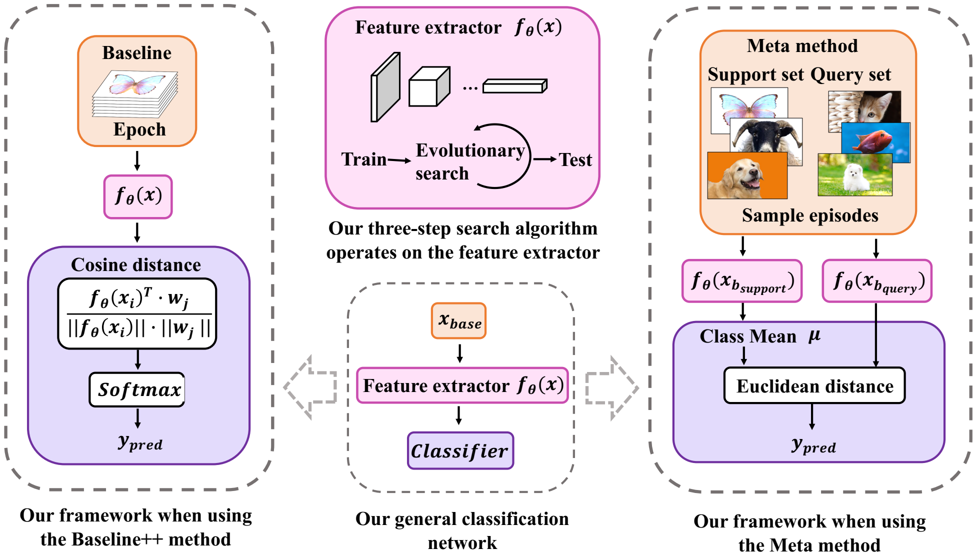


Figure 3: In this figure, we show that our three-step search algorithm operates on the feature extractor $f_\theta(x)$. Our general framework can easily be incorporated into the baseline method with cosine distance, denoted as baseline++ (Chen et al. 2019), as well as the meta-learning based methods.

different, as shown in Figure 2 (b) and Figure 3. Generally, our method searches the optimal strategy for transferring from base classes to novel classes through fixing or re-activating some particular layers that can help novel classes. As this is the core of our framework, we will elaborate in the following sections individually (Section 3.4 and 3.5).

Step 3: Partially transfer via searched strategy. As shown in Figure 2 (c), the final step is to apply our searched transfer strategy to the novel classes. Different from the simple baseline that fixes backbone and fine-tunes the last linear layer only, or meta-learning methods that use the base network as a feature extractor for the meta-testing, we will partially fine-tune our base network on the novel support set based on the searched strategies for both types of methods. This is also the core component to achieve significant improvement.

3.3. Search Space

Our search space is related to the model architecture we utilize for the few-shot classification. Generally, it contains the layer-level selection (fine-tuning or freezing) and learning rate assignment for fine-tuning. The search space can be formulated as m^K, where m is the number of choices for learning rate values and K is the number of layers in networks. For example, if we choose learning rate $\in \{0, 0.01, 0.1, 1.0\}$ as a learning rate zoo ("learning rate $= 0$" indicates we freeze this layer during fine-tuning), i.e., $m = 4$. For Conv6 structure, the search space includes 4^6 possible transfer strategies. Our searching method can automatically match the optimal choice for each layer from the learning rate zoo during fine-tuning. A brief comparison of the search space is described in Table 1. It increases sharply if we choose deeper networks.

3.4. Search Algorithm

Our searching step is following the evolutionary algorithm. Evolutionary algorithms, a.k.a genetic algorithms, base on the natural evolution of creature species. It contains reproduction, crossover, and mutation stages. Here in our scenario, first a population of strategies is embedded to vectors $\mathcal{V}$ and initialized randomly. Each individual v consists of its strategy for fine-tuning. After initialization, we start to evaluate each individual strategy v to obtain its accuracy on the validation set. Among these evaluated strategies we select the top K as parents to produce posterity strategies. The next generation strategies are made by mutation and crossover stages. By repeating this process in iterations, we can find a best fine-tuning strategy with the best validation performance. The detailed search pipeline is presented in Algorithm 1 and the hyper-parameters for this algorithm are introduced in Section 4.

Network	Conv6	ResNet-12	ResNet-K
Complexity	m^6	m^{12}	m^K

Table 1: Search Space of P-Transfer.

In this work we conduct the evolutionary search in transfer learning for few-shot classification. We target at fine-tuning with diverse learning rates to explore suitable transfer patterns in terms of knowledge with a simple and effective strategy design. At each layer, the learning rate is selected from a pre-defined zoo with all possible choices.

3.5. Incorporating into Few-Shot Frameworks

As in Figure 3, we introduce how to incorporate our search algorithm into existing few-shot classification frameworks. We choose the non-meta baseline++ (Chen et al. 2019) and meta ProtoNet (Snell, Swersky, and Zemel 2017) as examples.

Upon simple baseline++. Baseline++ aims to explicitly reduce intra-class variation among features by applying cosine distances between the feature and weight vector in the training and fine-tuning stages. As shown in Figure 3 (left sub-figure), we follow the design of distance-based classifier in searching but adjust the backbone feature extractor $f_\theta(x)$ through exploring different learning rates for different layers during fine-tuning. Intuitively, the learned backbone and distance-based classifier from our searching method are more harmonious and powerful than freezing backbone network and only fine-tuning weight vectors for few-shot classification, since our whole model is tuned end-to-end.

Algorithm 1: Evolutionary algorithm for searching the best fine-tuning configuration.

Input: Trained feature extractor: $\mathcal{N}$, layer index in a network: l, the meta-validation loss: $\mathcal{L}$, number of *Random* sampling operation: R, number of *Mutation*: M, number of *Crossover*: C, max number of *Iterations*: I.
Output: Optimized fine-tuning configuration v^*

```
define miniEval(v):
    N_i = Load(N)   # Inherit the weights from the feature extractor trained on the base dataset.
    Set grad_l = 0 if v_i[l] = 0   # Set the gradient to 0 according to the scheme vector.
    {v_i, accuracy} = miniFinetune(N_i)   # Fine-tune the targeting layers.
    return {v_i, accuracy}
for i = 0 : R do
    v_i = RandomChoice([0, m], L)   # Randomly sample fine-tuning schemes, i.e., chose lr for each layer.
    {v_i, accuracy} = miniEval(v_i)   # Evaluate accuracy on the validation dataset.
end for
v_topK = TopK({V, accuracy})   # Initialize the population with TopK vectors. V is the set of {v_i}.
for j = 0 : I do
    for i = 0 : M do
        v_i = Mutation(v_topK, L)   # Generate the off-spring fine-tuning vectors based on the top ones.
        {v_i, accuracy} = miniEval(v_i)   # Evaluate off-springs' accuracy on the validation dataset.
    end for
    for i = 0 : C do
        v_i = Crossover({v_topK1, v_topK2}, L)   # Generate the crossover vectors between two parents.
        {v_i, accuracy} = miniEval(v_i)   # Evaluate off-springs' accuracy on the validation dataset.
    end for
    v_topK = TopK({V, accuracy})   # Update the population by choosing the TopK vectors.
end for
v*, acc* = Top1({V, accuracy})   # Select the best scheme vector with highest validation accuracy.
return v*;
```

Upon meta-learning based methods. As shown in Figure 3 (right sub-figure), we describe the formulation of how to apply our searching method to meta-learning method for few-shot classification. In the meta-training stage, the algorithm first randomly chooses N classes, and samples small base support set $x_{b(s)}$ and a base query set $x_{b(q)}$ from samples within these classes. The objective is to learn a classification model M that minimizes N-way prediction loss of the samples in the query set Q_b. Here, the classifier M is conditioned on the provided support set x_b. Similar to baseline++, we train the classification model M by fine-tuning the backbone network and classifier simultaneously, to discover the optimal fine-tuning strategy. As the predictions from a meta-based classifier are conditioned on the given support set, the meta-learning method can learn to learn from limited labeled data through a collection of episodes.

4. Experiments

Dataset. We verify our method for few-shot learning on both *mini*-ImageNet dataset and CUB200-2012 dataset. *mini*-ImageNet dataset is a commonly used dataset for few-shot classification. It consists of 100 classes from ImageNet dataset (Deng et al. 2009), and 600 images for each class. We follow (Ravi and Larochelle 2017) to split the data into 64 base classes, 16 validation classes and 20 novel classes. CUB200-2011 contains 200 classes of birds (Wah et al. 2011). Follow (Hilliard et al. 2018), we split the data into 100 base classes, 50 validation classes and 50 novel classes. We validate the effectiveness of our method for generic classification on *mini*-ImageNet, and for fine-grained classification on CUB, as well as for cross-domain adaptation through transferring knowledge learned from *mini*-ImageNet to CUB.

Implementation. For meta methods, we sample episodes with 5 classes from the target dataset. Then for each class, we sample k instances as the support set and 15 instances as the query set for a k-shot task. In training, we train 60,000 episodes for 1-shot and 40,000 episodes for 5-shot tasks on the base dataset. In search, we sample 20 episodes from the validation dataset. We fine-tune the network on the support set for 100 iterations and evaluate the network on the query set. In evaluation, We fine-tune layers following the searched configuration on the support set and evaluate on the query set with episodes sampled from novel dataset. We ran 600 episodes and report the average accuracy and the 95% confidence intervals. The non-meta method differs only in the training stage, where we train the feature extractor for 400 epochs with a batchsize of 16 on the base dataset.

We adopt Adam optimizer with learning rate of 1e-3 for training. In fine-tuning, we use SGD with 0.01 learning rate for fully-connected layer and other searched learning rates for the corresponding layers. We use standard data augmentation including random crop, horizontal flip and color jitter. For Algorithm 1, we set population size $P = 20$, max iterations $I = 20$, and number of *random sampling* (R), *mutation* (M) and *crossover* (C) to 50.

4.1. Ablation Study

Comparison to fixed and manually designed fine-tuning. We first compare our proposed method with fixed and manually designed fine-tuning schemes using Conv6 and ResNet-12 structures on CUB and *mini*-ImageNet. The reason that we compare with fine-tuning the last convolutional layer as generally, the last layer is more domain-specific. Thus, in manually designed fine-tuning scheme, researchers usually

	CUB		*mini*-ImageNet	
	1-shot	5-shot	1-shot	5-shot
Fixed	**66.75±0.91**	79.48±0.63	52.52±0.79	71.16±0.66
Manual	64.27±0.97	79.26±0.61	53.22±0.77	71.43±0.64
Searched	65.82±0.97	**80.48±0.60**	**53.55±0.80**	**71.45±0.68**

Table 2: We validate on the non-meta method with Conv6 structure. We report the mean of 600 randomly generated episodes and the 95% confidence intervals. We compare the original learning algorithm (i.e., fine-tuning the fully-connected layer only and referring as "Fixed" in the table) with fine-tuning the human-defined last convolutional layer (i.e., "Manual" in the table) and fine-tuning the layers based on the evolutionary-searched scheme (i.e., "Searched" in the table).

	CUB		*mini*-ImageNet	
	1-shot	5-shot	1-shot	5-shot
Few-shot learning method: Baseline++				
Fixed	70.72±0.88	85.59±0.54	59.35±0.82	77.51±0.59
Searched	**73.88±0.87**	**87.81±0.48**	**64.21±0.77**	**80.38±0.59**
Few-shot learning method: ProtoNet				
Fixed	**73.82±0.92**	87.28±0.48	53.88±0.81	74.87±0.67
Searched	73.16±0.92	**88.32±0.46**	**54.36±0.81**	**76.59±0.64**

Table 3: Few-shot classification results on *mini*-ImageNet and CUB datasets with ResNet-12 structure. We apply our proposed algorithm to Baseline++ (non-meta) few-shot method as well as the meta-learning method (ProtoNet). The results show that in most cases, the proposed algorithm can discover a better knowledge transferring scheme than the original scheme.

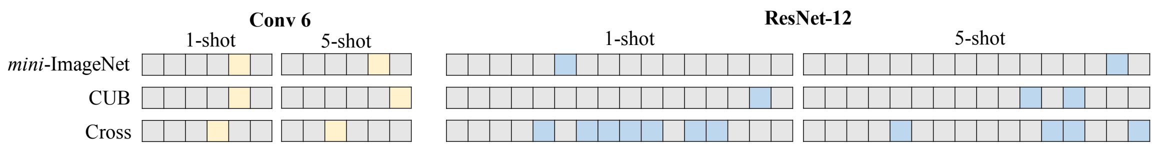


Figure 4: We visualize the fine-tuning scheme discovered by our evolutionary algorithm. The grey boxes denote layers without fine-tuning, the colored boxes denote layers that require fine-tuning. Different scenarios have different searched scheme. For example, in the cross-domain transfer-learning, more layers need to be fine-tuned to adapt the knowledge for the target domain.

fine-tune the last convolutional layer as a solution. Our results are shown in Table 2 (Conv6) and 3 (ResNet-12), where in general, our evolutionary strategy achieves better accuracy than fixing backbone and human-defined strategy. As a baseline, we obtain 40.84±0.8% (1-shot) and 50.95±0.9% (5-shot) when fine-tuning all the layers on *mini*-ImageNet.

Comparison to different normalization layers in cross-domain setting. As fine-tuning backbone networks will significantly be affected by the size of batchsize, while for the few-shot classification scenario, we do not have enough samples to increase the batchsize, and also the conventional fixing backbone solutions do not encounter this problem. Thus, here we further explore whether a better batch norm technique can deliver further improvement. Our results are shown in Table 4, in the cross-domain settings, group norm (Wu and He 2018) can achieve much better accuracy (about 2∼8% higher) than batch norm (Ioffe and Szegedy 2015) since it can overcome the drawback of optimization issue from small batchsize in traditional batch norm. Nevertheless, for a fair comparison, we only apply group norm in this ablation study to verify our conjecture that the limited batchsize may be a restriction to fully liberate the effectiveness and potential of our partial transfer method during backbone fine-tuning. As other state-of-the-art methods used standard batch norm, in our other results we still use the same batch norm method.

4.2. Searched Schemes and Final Results

To better understand our partial transfer method, we visualize the searched schemes in Figure 4. We observe two interesting phenomena which are in line with the intuition: (1) Deeper networks will always have more layers that require to be fine-tuned for few-shot learning; (2) When the domain difference between base and novel data is increased (in the cross-domain scenario), more layers are required to be fine-tuned.

Our final results are shown in Table 5, we can see that our partial transfer method can consistently outperform other state-of-the-art on both 1 and 5 shots settings. Even without additional training techniques like DropBlock (Ghiasi, Lin, and Le 2018) and label smoothing (Szegedy et al. 2016), our method still obtains a significant improvement, as our flexible transfer/fine-tuning can benefit from few support samples to adjust the backbone parameters.

4.3. Extension to Traditional Transfer Learning

We further explore the traditional transfer learning from ImageNet (Deng et al. 2009) to CUB200-2012 with the Inception V3 network (Szegedy et al. 2016). We use SGD optimizer with initial learning rate being 0.01 and linearly decay to 0. In transferring, we observe that, the weights learned from our method i.e., re-initializing and fine-tuning a few layers for

		1-shot		5-shot	
		BatchNorm	GroupNorm	BatchNorm	GroupNorm
Few-shot learning method: Baseline++					
Conv6	Fixed	40.77±0.70	44.80±0.78	58.15±0.72	64.15±0.74
	Searched	**41.69±0.72**	**45.34±0.76**	**61.53±0.70**	**67.42±0.72**
ResNet-12	Fixed	43.14±0.72	43.52±0.71	63.25±0.70	64.02±0.71
	Searched	**43.77±0.74**	**45.86±0.72**	**68.30±0.73**	**74.22±0.66**
Few-shot learning method: ProtoNet					
Conv6	Fixed	36.34±0.71	**38.74±0.77**	**55.38±0.71**	59.30±0.73
	Searched	**36.36±0.73**	38.36±0.69	54.06±0.73	**62.33±0.75**
ResNet-12	Fixed	39.14±0.73	44.88±0.78	60.11±0.73	67.18±0.74
	Searched	**39.38±0.72**	**45.29±0.77**	**60.24±0.73**	**68.42±0.75**

Table 4: In this table, we further evaluate our method on the cross-domain few-shot learning tasks, i.e., transferring the knowledge from *mini*-ImageNet to CUB. We conduct experiments on both meta and non-meta methods. We find that when there exist domain difference, fine-tuning more layers are required. Moreover, we further discovered using GroupNorm can also bridge the distribution difference between training and testing, which outperforms the results of using BatchNorm.

Method	Backbone	1-shot	5-shot
MatchingNet (Vinyals et al. 2016)	Conv4	43.56±0.84	55.31±0.73
MatchingNet* (Vinyals et al. 2016)	ResNet-12	**54.76±0.82**	**70.01±0.70**
ProtoNet (Snell, Swersky, and Zemel 2017)	Conv4	48.70±1.84	63.11±0.92
ProtoNet* (Snell, Swersky, and Zemel 2017)	ResNet-12	**53.88± 0.81**	**74.87± 0.67**
Parameters from Activations (Qiao et al. 2018)	WRN-28-10	59.60±0.41	73.74±0.19
Closer Look (Chen et al. 2019)	ResNet-18	51.87±0.77	75.68±0.63
SNAIL (Mishra et al. 2017)	ResNet-12	55.71±0.99	68.88±0.92
AdaResNet (Munkhdalai et al. 2017)	ResNet-12	56.88±0.62	71.94±0.57
TADAM (Oreshkin, López, and Lacoste 2018)	ResNet-12	58.50±0.30	76.70±0.30
MetaOptNet (Lee et al. 2019)	ResNet-12	60.33±0.61	76.61±0.46
MetaOptNet† (Lee et al. 2019)	ResNet-12	62.64±0.61	78.63±0.46
Meta-Baseline (Chen et al. 2020)	ResNet-12	63.17± 0.23	79.26± 0.17
P-Transfer (ours)	ResNet-12	**64.21±0.77**	**80.38±0.59**

Table 5: Comparison with the state-of-the-art results on *mini*-ImageNet dataset. * denotes the results re-implemented by us. † indicates the results are from more training techniques like DropBlock and label smoothing.

partial transfer achieves higher accuracy than inherit all the weights and do fine-tuning, as shown in Table 6.

	Baseline	**Partial transfer**
Top-1 Accuracy	82.9%	**83.8%**

Table 6: The comparison between inheriting all weights and partially reinitializing weights in transfer learning.

Why Is Partial Better Than All in Few-shot? Usually the base and novel class are in the same domain, so using the pre-trained feature extractor on base data and then transferring to novel data can obtain good or moderate performance. However, as shown in Figure 4, in the cross-domain transfer-learning, more layers need to be fine-tuned to adapt the knowledge for the target domain since the source and target domains are discrepant in content. In this circumstance, the conventional transfer learning is no longer applicable. Our proposed partial transferring with diverse learning rates on different layers is competent for this intractable situation, and intuitively, fixed transferring is generally a special case of our strategy and ours has better potential in few-shot learning.

5. Conclusion

We have introduced a partial transfer (P-Transfer) method for the few-shot classification. Our method is the first attempt to thoroughly explore the capability of transferring through searching strategies in few-shot scenario without any proxy. Our method boosts both the meta and non-meta based methods by a large margin under 1-shot or 5-shot circumstances, as our flexible transfer/fine-tuning can benefit from few support samples to adjust the backbone parameters. Intuitively, our partial transfer has larger potential for few-shot classification and even for the traditional transfer learning. We hope our method can inspire more methods along this direction. In the future, we will perform more analyses about how partial transfer helps the few-shot problems. We will apply our method on other few-shot tasks like detection, segmentation to explore the upper limit of our proposed transfer method.

References

Andrychowicz, M.; Denil, M.; Gomez, S.; Hoffman, M. W.; Pfau, D.; Schaul, T.; Shillingford, B.; and De Freitas, N. 2016. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, 3981–3989.

Antoniou, A.; Edwards, H.; and Storkey, A. 2018. How to train your MAML. *arXiv preprint arXiv:1810.09502* .

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

Chen, Y.; Wang, X.; Liu, Z.; Xu, H.; and Darrell, T. 2020. A new meta-baseline for few-shot learning. *arXiv preprint arXiv:2003.04390* .

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Elsken, T.; Staffler, B.; Metzen, J. H.; and Hutter, F. 2020. Meta-Learning of Neural Architectures for Few-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12365–12375.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1126–1135. JMLR. org.

Garcia, V.; and Bruna, J. 2017. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043* .

Ghiasi, G.; Lin, T.-Y.; and Le, Q. V. 2018. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*, 10727–10737.

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.

Hilliard, N.; Phillips, L.; Howland, S.; Yankov, A.; Corley, C. D.; and Hodas, N. O. 2018. Few-shot learning with metric-agnostic conditional embeddings. *arXiv preprint arXiv:1802.04376* .

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* .

Kim, J.; Kim, H.; and Kim, G. 2020. Model-Agnostic Boundary-Adversarial Sampling for Test-Time Generalization in Few-Shot learning. In *ECCV*.

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.

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

Li, Z.; Zhou, F.; Chen, F.; and Li, H. 2017. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835* .

Liu, H.; Simonyan, K.; Vinyals, O.; Fernando, C.; and Kavukcuoglu, K. 2017. Hierarchical representations for efficient architecture search. *arXiv preprint arXiv:1711.00436* .

Miikkulainen, R.; Liang, J.; Meyerson, E.; Rawal, A.; Fink, D.; Francon, O.; Raju, B.; Shahrzad, H.; Navruzyan, A.; Duffy, N.; et al. 2019. Evolving deep neural networks. In *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, 293–312. Elsevier.

Mishra, N.; Rohaninejad, M.; Chen, X.; and Abbeel, P. 2017. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141* .

Munkhdalai, T.; and Yu, H. 2017. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2554–2563. JMLR. org.

Munkhdalai, T.; Yuan, X.; Mehri, S.; and Trischler, A. 2017. Rapid adaptation with conditionally shifted neurons. *arXiv preprint arXiv:1712.09926* .

Oreshkin, B.; López, P. R.; and Lacoste, A. 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, 721–731.

Qi, H.; Brown, M.; and Lowe, D. G. 2018. Low-shot learning with imprinted weights. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5822–5830.

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.

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

Real, E.; Aggarwal, A.; Huang, Y.; and Le, Q. V. 2019. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, 4780–4789.

Real, E.; Moore, S.; Selle, A.; Saxena, S.; Suematsu, Y. L.; Tan, J.; Le, Q. V.; and Kurakin, A. 2017. Large-scale evolution of image classifiers. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2902–2911. JMLR. org.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* .

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.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Thrun, S.; and Pratt, L. 2012. *Learning to learn*. Springer Science & Business Media.

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* .

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

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. *California Institute of Technology* (CNS-TR-2011-001).

Wu, Y.; and He, K. 2018. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19.

Xie, L.; and Yuille, A. 2017. Genetic cnn. In *Proceedings of the IEEE international conference on computer vision*, 1379–1388.

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.

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.