Article:
Sun, N. and Yang, P. orcid.org/0000-0002-8553-7127 (2023) T2L: Trans-transfer Learning for few-shot fine-grained visual categorization with extended adaptation. Knowledge-Based Systems. 110329. ISSN 0950-7051

https://doi.org/10.1016/j.knosys.2023.110329

T^{2}L: Trans-transfer Learning for Few-shot Fine-grained Visual Categorization with Extended Adaptation

Nan Sun[a], Po Yang[b,*]

[a]School of Information Science and Engineering, Yunnan University, 650091, China
[b] Department of Computer Science, The University of Sheffield, Sheffield, United Kingdom

ARTICLE INFO

Keywords:
Computer Vision, Fine-grained Visual Categorization, Few-shot Learning, Transfer Learning.

ABSTRACT

Fine-grained visual categorization requires the ability to distinguish categories with subtle differences, which is also a problem constantly burdened by collecting and labelling samples. While transfer learning is extensively used in Fine-grained visual categorization, these approaches generally does not work under the few-shot regime. As apposed to meta-learning which suffers limitations such as shallow adaptation, and traditional transfer learning which is not task-oriented, we propose a novel transfer learning approach which we refer to as Trans-transfer Learning (T^{2}L). Firstly, a two-phase learning framework is proposed to facilitate task-orientation. Secondly, a novel explainable-learning based procedure is employed to reconfigure the network for training on few samples, after which a deep adaptation procedure is conducted by simultaneously optimizing the feature extractor and the classification boundaries. It is motivated by the fact that a large factor of the over-fitting comes from noises that appear in every layers of the activation. The reconfigured model can train solely based on inter-class differences, which not only alleviates over-fitting, but also rediscovers more discriminating features. The proposed approach is comparatively evaluated with state-of-the-art approaches in this field and demonstrates better performances consistently.

1. Introduction

Fine-grained visual categorization (FGVC)[1, 2, 3] aims to distinguish subordinate object categories with subtle differences. For instance, recognizing natural categories such as species of birds [4], dogs [5] and plants [6]; or man-made categories such as car make models [7]. Recent advances on Deep Neural Networks for visual recognition [8, 9, 10] have stirred remarkable progress on FGVC but most of the algorithms are fueled with exorbitant data collection cost. Generally, in order to train a model with reasonable generalization performance, one needs to possess a vast amount of data, either labeled or unlabeled. DNN's extraordinary ability of fitting to data is a double-edged blade. On one hand, it provides unparalleled representation ability to any mathematical functions. But on the other hand, it has substantial requirement for data not just in terms of total volume but its distribution.

In order to recognize instances drawn from a certain concept, the number of samples from this particular category plays a vital role. It is unreasonable to ask for a huge collection of samples for each category we want to handle, especially when the model is deployed to end users. For that reason, research in the field of few-shot learning, has gained more and more interest [11, 12, 13, 14]. In few-shot learning, we refer to the rare categories as novel classes, which only have few samples available each, typically 1 to 5.

When it comes to fine-grained domains, transfer learning approaches are extensively used and achieved prevalence [1, 2, 15]. However, the approaches are designed under regular-shot assumption, whether it's adversarial-based [16], discrepancy-based [17], semi-supervised [18], unsupervised [19], or even a simple pre-training and fine-tuning. They all assume many-shot in the target domain. As shown in Fig.1, the above methods would crumble extending to few-shot dataset, as it is generally observed that even co-training or joint-training with large auxiliary data, it does not prevent the rare classes from over-fitting [20, 21, 22].

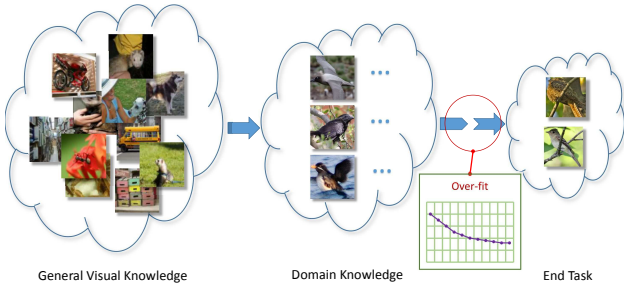


Figure 1: The two-stage transfer learning framework. From left to right, are data from generic visual database, fine-grained database (from a disjoint set of classes of the end-task), and the few-shot dataset. Traditional transfer learning are designed to complete the first stage, but does not consider few-shot tasks. A typical validation curve when a traditional approach is conducted on few-shot data is shown, which suggests that the performance would deteriorate due to severe over-fitting.

Few-shot learning is a research filed that specifically addresses machine learning under input scarce condition. Most existing FSFG algorithms directly stem from generic few-shot learning field [23, 24]. There are meta-learning approaches [25, 26, 27, 28, 29], which mostly works on small-scale classifiers. And there are non-meta learning approaches [11, 30] which are more computationally efficient, and able to leverage generic visual databases such as ImageNet. But we

po.yang@sheffield.ac.uk (P. Yang)
ORCID(s):

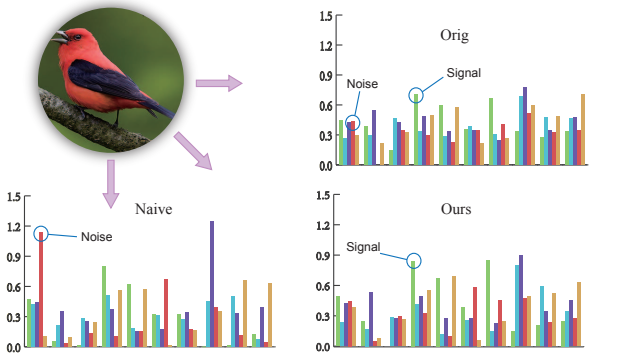


Figure 2: A visualization of embeddings by different models of a randomly selected sample from the CUB novel set. Upper right is extracted by the original model. The lower left is extracted by naive method on novel classes. The lower right is produced by the desired model which is facilitated by the technique proposed in this work. As seen from the circled-out bins for example, naive method tends to amplify noise which are very likely misleading while our approach does the opposite.

believe it is still an under-explored area and characteristics about the fine-grained domains are not fully exploited considering that most of the previous works are designed for general domains.

To address the above mentioned limitations, we argue that Extended Adaptation (EAD) is a key for further improvement, which means the model should be both tasks-oriented, and adjusted for a long horizon in its deep layers. The model is designed with the following objectives in mind.

The first goal is task-orientation. As shown in Fig.1, general visual databases are rich in both volume and semantic diversity which should be incorporated into the model. In order to achieve that, we hypothesize that three levels of data sources will be used in consecutive order. Thus, we formulate our approach as a two-phase process which we refer to as Trans-transfer Learning (T^2L). Phase one is essentially a regular-shot transfer learning which infuses general visual knowledge into a fine-grained domain. Based on theoretical and empirical evidences that metric-based meta-learning are able to render the model to be more sensitive for few-shot data, we propose to combine metric-based learning and transfer learning for our purpose. Phase two seeks to further adapt the above learned model to novel classes, not only for the classification layer, but also for deep layers of the feature extractor, resulting in an improved classification layer, as well as a better feature extractor, the later of which could potentially be utilized by various visual learning tasks [31].

The second goal is deep adaptation. Adaptation to novel classes has been the goal for many few-shot learning works. Most of them are under the principle of meta-learning [12, 26, 27, 28, 29], which is motivated by the idea of learning to learn [32]. This genre of algorithms tries to learn a parameter initialization that is more sensitive to few-shot data and can quickly adapt to novel classes. It's adaptation approach is essentially fine-tuning, but only for very few steps, typically 4 to 5. And recently it is pointed out that this adaptation mostly affects only the last layer but not the feature extractor [33].

Thus, the above mentioned prior arts are not able to achieve EAD and their performances are therefore limited. The main obstacle is the over-fitting. In traditional supervised learning, the objective functions are designed to lower the intra-class divergence and increase the inter-class differences. The optimization process will iteratively discover similarities between intra-class samples and weight these similarities under conditions they do not intervene with other classes, thereby features are discovered. As a result, intra-class sample plays a vital role for shaping the manifold. But in few-shot learning, *e.g.*, 1-shot cases, there is only one sample for each class thus no intra-class similarity to discover and only inter-class differences are in effect. However, observably many raw visual differences could be existing between different categories, naively applying the optimization often gravitates towards irrelevant artifacts, i.e., noise, since they are detriment for the classification. A preliminary experiment in fact proves our intuition: Fig.2 shows an example of how naive fine-tuning leads to some over extended dimensions in feature space, which are very likely to be just noise according to the original graph. This is the fundamental reason for the catastrophic over-fitting.

Based on the above observation, we make a preliminary attempt towards this direction. As apposed to previous works which use the pre-trained model simply as a parameter initialization, we make dual use of it. In fine-grained domain, the objects share structural similarity even when shifting to novel classes since this information is class-agnostic but domain-relevant. Inspired by explainable-learning method Class-agnostic Activation Map [34], suppose we are able to locate a subset of the activations which are the key characteristic of this class. Subsequently, these selected activations can be back-propagated, so as to identify corresponding lower layer features and the weights that connect between them. The pre-trained model are in turn used as initialization and train for many epochs, after which the model would exhibit improved performance. Our contributions are summarized below:

- In order to facilitate EAD, we propose Trans-transfer Learning (T^2L), which is a two-phase learning approach. Using our proposed technique, metric-based learning and a novel deep adaptation module can be concatenated seamlessly and the chain of knowledge transfer is extended onto few-shot tasks. The overall accuracy out-performs previous SoTAs, *e.g.*, 54.10% top-1 accuracy when recognizing bird species 50-ways.

- For phase two, an CAAM-based reconfiguration strategy is proposed in order to further adjust the model on a dataset as small as *1-shot*, achieving up to 3.2% further performance gain (by absolute value), while successfully avoiding over-fitting for networks as deep as Resnet-50. Up to our knowledge, it is the first approach that is capable to conduct a full-training on such small sample size and such deep architectures.

- For a more realistic scenario where well established meta-training set is not given, we propose a $\mathcal{H}$-based domain extraction technique to form a pseudo meta-training set out of a generic dataset with wide domain coverage. Then the proposed two-phase method can be applied to transfer the learned model onto a few-shot fine-grained domain. The results out-perform previous one-phase method and domain generalization techniques.

The rest of this paper is organized as follows: Section 2 discusses related works and literatures. Section 3 formalizes the notations and backgrounds. Section 4 presents the proposed methodology and techniques. Section 5 shows the results of empirical evaluation. Section 6 discussions the advantage and limitations of our approach and some directions for our future work. Section 7 summarizes and concludes the paper.

2. Related Work

2.1. Few-shot Learning

Recent years have witnessed a vast amount of works on the few-shot classification task. Few-shot learning algorithms can be roughly categorized into the following branches. The metric-based approaches are centered around a pair-wise metric function. The first meta-learning style approach are proposed by Vinyals *et al.* [35], which introduced an episodic training routine. The meta-task process is trained to be more sensitive to few-shot data, which can be seen as a utilization of the concept of meta-learning. This genre of algorithms includes Prototypical Networks [36] which follows the episodic training procedure, and propose to use prototype to represent the novel samples and a simple euclidean distance as metric function. There are more sophisticated ways of constructing the metric function but most of them introduce a substantial computation overhead especially in terms of run-time memory [25, 37, 38]. In light that they all follow a similar episodic training routine and differs only in the way they calculate the distance function, we choose to use protonet as the baseline to build our approach. However we note that our approach can be extended to any metric-based algorithms in general.

Another branch are the optimization-based approaches [3, 12, 27, 28, 29]. They follow the episodic training procedure as well. The main difference from the metric-based approaches are the way they integrate the support-set into the model. Instead of directly using the extracted feature vectors, the optimization-based approaches use a few steps of gradient descent to train the model on the support-set, a process often referred to as the inner-loop. The outer-loop uses the query-set to optimize this entire process by gradient descent, which is similar to the metric based one.

Related work also includes non-meta learning approaches that achieved competitive performance [11, 30]. Chen *et al.* [11] pointed out that training on all base classes with a simple softmax classifier achieves comparable performance on few-shot learning benchmarks. This result indicates that features learned on the base class are automatically discriminative on the novel class to some extent. In a way, our approach can be seen as a combination of meta-learning and non-meta learning.

2.2. Fine-Grained Visual Categorization

Fine-Grained Visual Categorization can be seen as a subarea of General Visual Classification. In theory, general image classification algorithms can be use to train the model on in-domain data. However, to achieve reasonably good performance with CNNs, one needs to train networks with vast amounts of in-domain data, preferably with supervision. But collecting a labeled fine-grained dataset often requires expert-level domain knowledge and long data curation therefore is difficult to scale. As a result, commonly used FGVC datasets [4, 5, 7] are relatively small, typically containing around 10k of labeled training images.

In such a scenario, transfer learning base on large generic visual databases such as ImageNet [39], iNaturalist [6] and Places [40] are often adopted. Due to the remarkable success of using pre-trained CNNs for transfer learning, extensive efforts have been made on understanding transfer learning [41, 42, 43].

In recent years, deep neural networks have developed advanced abilities in feature extraction and function approximation [8, 10], bringing significant progress in the fine-grained image classification task. Notably, second order bilinear feature interactions was shown to be very effective [44]. To capture subtle visual differences, visual attention [15] and deep metric learning [45] are often used.

2.3. Few-shot Fine-grained Visual Categorization

A few generic few-shot learning literatures explored the performance of their algorithms on FSFG settings [11, 23, 24]. However, the nature of fine-grained domain is hardly addressed except for some recently published specialized approaches [46, 47, 48, 49, 50, 51].

Wei et al. [46] proposed a few-shot learning model specifically designed for FSFG by employing bilinear-networks to tackle the problem exploiting the rich feature provided by the outer-product of two alexnet [9] on par with a piece-wise mapping network which maps the bilinear feature to be a concatenation of modulated features with small dimensions each focusing on a certain part of the object thereby learns the decision boundaries of the input data more discriminatively. Li et al. [47] further replace the naive self-bilinear pooing as the covariance pooling. On the contrary, we did not add any complex structures to the network but rather focus on how it is trained.

Huang et al. [48] propose a Low- Rank Pairwise Alignment Bilinear Network to capture the nuanced differences between the support and query images for learning an effective distance metric. Sun *et al.* [49] proposed a feature fusion model to explore the largest discriminative features by focusing on key regions. The model utilizes focus-area location to discover the perceptually similar regions among objects and a Center Neighbor Loss to form robust embedding space dis-

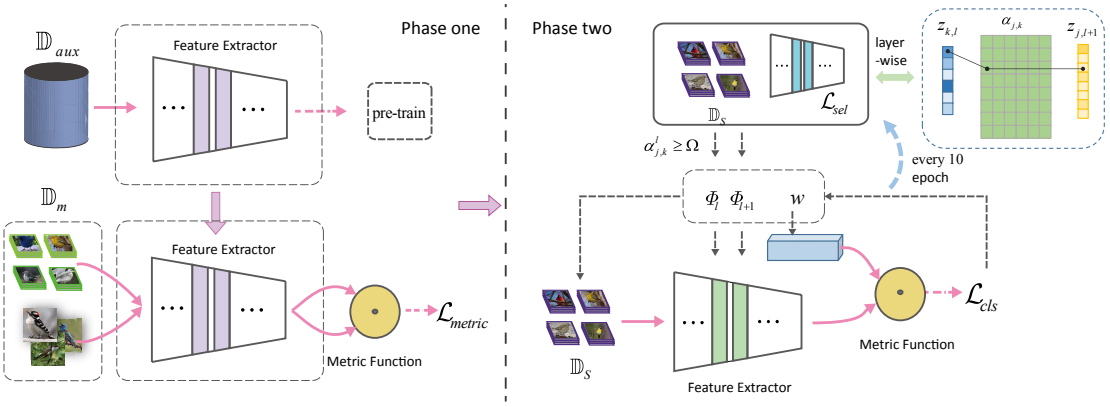


Figure 3: System overview of the proposed approach. On the left, is transfer learning phase one, which is a composite of pre-training on $\mathbb{D}_{\text{aux}}$ and metric-based training on $\mathbb{D}_{\text{m}}$ successively. On the right, is the phase two adaptation, which transforms a metric-based model into a Implicit Metric-model, and train the selected feature extractor parameter and the weight vector collectively in an adaptive manner. The layer-wise importance value calculation is detailed in the figure which is the product of the channel-wise importance score of two consecutive layers.

tribution for generating discriminative features. But they do not address adaptation in the deep layers.

Deep data-augmentation based method have also been proposed in FSFG [50] which achieve significant improvement in generalization performance, but with exorbitant computation cost. And in theory, augmentation-based techniques can work in tandem with other techniques including ours but the engineering required to bind them together is not explored here. In this paper, we have not explored the integration of any previous deep data-augmentation techniques [52, 53, 54] but only applies standard rotation and random cropping.

Pahde *et al.* [55] proposed a cross-modality FSFG model, which embeds the textual annotations and image features into a common latent space and introduced a discriminative text-conditional GAN for the sample generation to subsidize for the absence of training samples. However, it is costly to obtain rich annotations for the entire fine-grained dataset. Recently Tang *et al.* [56] introduces another multi-modal approach dubbed deep Pose Normalization which aims to push the ability of FSFG to the next level by exploiting the information provided by rich annotation. However they require much more costly annotation which could even be prohibitively expensive.

3. Background

3.1. Problem Setting

We first introduce some notation and formalize the few-shot image classification problem. Let (x, y) denote an image and its ground-truth label respectively. The training and testing set for a few-shot task are in turn referred to as *Support-set* and *Query-set*, and respectively denoted as:

$$\mathbb{D}_{\text{s}} = \left\{(x_i, y_i)\right\}_{i=1}^{|\mathbb{D}_{\text{s}}|}, \tag{1}$$

$$\mathbb{D}_{\text{q}} = \left\{(x_i, y_i)\right\}_{i=1}^{|\mathbb{D}_{\text{q}}|}. \tag{2}$$

where $y_i \in C_{\text{s}}$ for some set of classes C_{s}. The number of objective classes, also refer to as *ways*, is equal to $|C_{\text{s}}|$. The set $\left\{x_i \mid y_i = k,\ (x_i, y_i) \in \mathbb{D}_{\text{s}}\right\}$ is the *support-set* for class k and its cardinality is referred to as *shots* since it gives the number of support samples for each class. Shot is typically between 1-5 in few-shot settings. The set $\left\{x_i \mid y_i = k, (x_i, y_i) \in \mathbb{D}_{\text{q}}\right\}$ is the *query* of class k and its cardinality is q.

A few-shot task is to categorize each $x_i \in \mathbb{D}_{\text{q}}$ in to C_{s}, by only showing the set of samples $\mathbb{D}_{\text{s}}$, suppose this learned parameterized function Φ^* is:

$$\hat{y} = \Phi^*(x;\ \mathbb{D}_{\text{s}}), \tag{3}$$

where Φ^* is, ideally, sufficient to classify $\mathbb{D}_{\text{s}}$, as measured by, say, the cross-entropy loss

$$\Phi(\mathbb{D}_{\text{s}}) = \arg\min_{\Phi} \frac{1}{|\mathbb{D}_{\text{s}}|} \sum_{(x,y)\in\mathbb{D}_{\text{s}}} -\log p_{\Phi}(y|x). \tag{4}$$

When presented with a test datum, the classification rule is typically chosen to be of the form

$$\Phi^*(x; \mathbb{D}_{\text{s}}) \triangleq \arg\max_{k} p_{\Phi^*}(k|x), \tag{5}$$

A simple way to form p_{Φ^*} is to employ some measurement d and normalize the output with softmax function, the model can be written as:

$$p(y_i|x_i, \Phi, w) = \frac{\exp(\text{d}_w(f_{\Phi}(x_i), w_{y_i}))}{\sum_k \exp(\text{d}_w(f_{\Phi}(x_i), w_k))}, \tag{6}$$

$\mathbb{D}_{\text{s}}$ is thereby learned by the model and is represented by Φ^*. This is referred to as Direct Training in this paper.

However, the aforementioned $\mathbb{D}_{\text{s}}$ has very few shots, directly optimizing this model would lead to severe over-fitting. In analogy to human learning which are able to summon previously knowledge, the model are required to be given other sources of information. The *status quo* in few-shot learning is exploiting a *meta-training* set, which is labeled samples

from the same (or similar) domain of $\mathbb{D}_{\mathrm{s}}$, but from a disjoint set of classes and has far more samples, denoted as $\mathbb{D}_{\mathrm{m}}$

$$\mathbb{D}_{\mathrm{m}} = \left\{(x_i, y_i)\right\}_{i=1}^{|\mathbb{D}_{\mathrm{m}}|}, \tag{7}$$

where $y_i \in C_{\mathrm{m}}$, with set of classes C_{m} disjoint from C_{s}.

3.2. Training with Auxiliary Data

Solving few-shot learning usually requires training with auxiliary data. The above mentioned $\mathbb{D}_{\mathrm{m}}$ is one type of auxiliary data, which has very small domain distance from the end task.The other type we use in this paper are from more general domain [9] or a distant domain [40] both of which have a generally large distance from the target task. Some of the methods of making use of auxiliary data are listed below.

3.2.1. Feature training&reusing

To avoid over-fitting of the data-scarce part, a simple remedy is to simply train the model on $\mathbb{D}_{\mathrm{m}}$, ideally, the trained model would work well on $\mathbb{D}_{\mathrm{s}}$ in light that $\mathbb{D}_{\mathrm{m}}$ and $\mathbb{D}_{\mathrm{s}}$ are from similar distributions, it is referred to as *Domain Generalization*.

$$\Phi^* w^*(;\mathbb{D}_{\mathrm{m}}) = \arg\min_{\Phi w} \sum_{(x_i,y_i)\in\mathbb{D}_m} -\log p(y_i|x_i,\Phi,w)\,. \tag{8}$$

It is clear that the above optimization is irrelevant of $\mathbb{D}_{\mathrm{s}}$, Φ^* is only trained on $\mathbb{D}_{\mathrm{m}}$. But during testing, dds is required, since it represents the identity of classes in the testing test. For this purpose, a simple Nearest Neighbor classification can be employed based on the already trained feature extractor.

3.2.2. Joint training

In this work, we propose to train not only on $\mathbb{D}_{\mathrm{m}}$, but on $\mathbb{D}_{\mathrm{s}}$ as well. The simplest way to use both $\mathbb{D}_{\mathrm{m}}$ and $\mathbb{D}_{\mathrm{s}}$ in training of the objective is to just join the data together, than train on the expanded dataset, which is referred to as *joint training*:

$$\Phi^* w^*(;\mathbb{D}_{\mathrm{m}},\mathbb{D}_{\mathrm{s}}) = \arg\min_{\Phi w} \sum_{(x_i,y_i)\in\mathbb{D}_{\mathrm{m}}\cup\mathbb{D}_{\mathrm{s}}} -\log p(y_i|x_i,\Phi,w)\,. \tag{9}$$

During testing, the output logits related to $\mathbb{D}_{\mathrm{m}}$ is shut-off since the output labels are from $\{y|y \in \mathbb{D}_{\mathrm{s}}\}$.

3.2.3. Re-training

Another way to incorporate different dataset in training is Re-training, take Eq.8 for example, the model trained on $\mathbb{D}_{\mathrm{m}}$ has infused different levels of visual knowledge in a parameterized model. These parameterized can be seen as a initialization of next stage training, *e.g.*, $\mathbb{D}_{\mathrm{s}}$:

$$\Phi^* w^*(;\mathbb{D}_{\mathrm{s}})|_{\Phi_0 w_0} = \arg\min_{\Phi w} \sum_{(x_i,y_i)\in\mathbb{D}_s} -\log p(y_i|x_i,\Phi,w)\,. \tag{10}$$

There is no guarantee that the later stage will preserve any characteristic of the former stage in that their optimization objective could be completely different. Therefore Re-training on few-shot data would nevertheless causes over-fitting.

3.2.4. Meta training

Previous Re-training methods are mostly used when both $\mathbb{D}_{\mathrm{m}}$ and $\mathbb{D}_{\mathrm{s}}$ are many-shot. Meta-learning is seen to borrow strength from both Re-training and Feature Train&reusing in that it both train the model to be able to re-train on few-shot data and also make (part) use of the trained feature. It uses the two datasets in a bi-level programming paradigm:

$$\begin{aligned}\Phi^*(\mathbb{D}_{\mathrm{s}};\mathbb{D}_{\mathrm{m}}) = \arg\min_{\Phi} \sum_{(x,y)\in\mathbb{D}_{\mathrm{m}}} L(p_{\theta^*}(x;\mathbb{D}_{\mathrm{s}}),y)\\ st.\theta^* = \arg\min_{\theta} \sum_{(x,y)\in\mathbb{D}_{\mathrm{s}}} L(p_\theta(x),y)\,.\end{aligned} \tag{11}$$

where L is chosen in a set of objective functions. θ is optimized (usually coarsely) on a small dataset $\mathbb{D}_{\mathrm{s}}$, then this model is readjusted by optimizing on $\mathbb{D}_{\mathrm{m}}$, ultimately, the model will gravitate towards a state that works well with small dataset such as $\mathbb{D}_{\mathrm{s}}$ with which few adaptation is enough. Note that Φ and θ might overlap or even equivalent with each other, meaning the superset of parameter in this meta-learning process is bounded, the two levels of optimization just choose a subset of parameters to adjust.

4. Proposed Method

4.1. Implicit Metric-model

First, we formalize two different but correlated few-shot model and their respective training agenda, the later of which is introduced as Implicit Metric-model which is a key to adapt our model to different levels of dataset.

4.1.1. Metric-based model

Metric-based model is a unique branch of meta-learning model which is extensively used in classification tasks [35, 36, 37]. Its training agenda is denoted as V_{exp}. It typically employs an episodic sampling technique, which is randomly sampling $\mathbb{D}_{\mathrm{m}}$ into a series of batches in the form of $\{\mathbb{D}_{\mathrm{s}}, \mathbb{D}_{\mathrm{q}}\}$, to simulate the end task. Each batch is referred to as an episode. Metric-based training is to directly optimize the distance between $\mathbb{D}_{\mathrm{s}}$ and $\mathbb{D}_{\mathrm{q}}$. Specifically, we introduce metric-based model:

$$\mathfrak{A}_\Phi(x,\ y) := \mathcal{X}(f_\Phi(x),\ \mathbf{c}_y)\,, \tag{12}$$

where d is the metric function. For each episode, $\mathbf{c}$ is obtained by direct calculation and kept fixed, which is defined to be the centroid of the support set. Specifically, we denote the centroid of support set for class k:

$$\mathbf{c}_k = \frac{1}{|\mathbb{D}_{\mathrm{s}}|} \sum_{(x_i,y_i)\in\mathbb{D}_{\mathrm{s}}} \mathbb{1}_{[y_i==k]} f_\Phi(x_i)\,, \tag{13}$$

where $\mathbb{1}_{[y_i==k]}$ is the indication function of whether x_i belong to category k. $\mathbf{c}$ is an explicit representation of the class identities. Typically a cross-entropy is used to construct the loss function for explicit model L_{exp}.

$$L_{exp}(\Phi;\mathbb{D}) = \sum_k \sum_{x_i\in\mathbb{D}_{\mathrm{q}}}^{y_i=k} \Big[-\mathfrak{A}_\Phi(x_i,\ k)$$

$$+ \log \sum_j \exp(\mathfrak{A}_\Phi(x_i,\, k))\Big], \quad (14)$$

where $\sum_{x_i \in \mathbb{D}_q}$ is the input of the *query set*. The gradient of L_{exp} is used to update parameters Φ, thus the procedure V_{exp} is given by the following:

$$V_{exp}(\mathfrak{A}_{\Phi_0}(x,\, y); \mathbb{D}) : \\ \Phi^{t+1} := \Phi^t + \gamma \frac{\partial L_{exp}(\Phi; \mathbb{D})}{\partial \Phi}. \quad (15)$$

where $t = \{0, 1, \ldots, n\}$, n is the number of iterations.

4.1.2. Implicit Metric-model

Now we discuss a more general form of the metric-based model. On top of the traditional metric-based models, the class representation can be replaced by internal variables which evolves by solving the equation:

$$w^*(; \mathbb{D}) = \arg\min_{\Phi w} \sum_k \sum_{x_i \in \mathbb{D}}^{y_i = k} \Big[- \mathfrak{A}_{\Phi w}(x_i,\, k) \\ + \log \sum_j \exp(\mathfrak{A}_{\Phi w}(x_i,\, k))\Big]. \quad (16)$$

where

$$\mathfrak{A}_{\Phi w}(x,\, y) := \mathcal{X}(f_\Phi(x),\, w_y), \quad (17)$$

which implies w is no longer fixed during each episode and is optimized along with the model. $\mathcal{X}$ is the exact metric function as in Eq.12. Note that if you consider the negation of the dot-product in traditional classification baseline methods as a metric function, they can be categorized as Implicit Metric-models. The class representation is now parameterized in w instead directly calculate from class prototype. We call w the implicit class representation.

The loss function is typically chosen to be the cross-entropy loss of the support-set going through the model:

$$L_{imp}(\Phi, w; \mathbb{D}) = -\sum_{(\mathbf{x}_i, y_i) \in \mathbb{D}} \log \frac{\exp(\mathfrak{A}_{\Phi w}(x_i,\, y_i))}{\sum_{j=0}^{N} \exp(\mathfrak{A}_{\Phi w}(x_i,\, y_j))}. \quad (18)$$

Thus we establish the optimization procedure of a metric-based model with implicit class representations on a dataset $\mathbb{D}$:

$$V_{imp}(\mathfrak{A}_{\Phi_0 w_0}(x,\, y); \mathbb{D}) : \\ \Phi^{t+1} := \Phi^t + \gamma \frac{\partial L_{imp}(\Phi, w; \mathbb{D})}{\partial \Phi}, \quad (19)$$

$$w^{t+1} := w^t + \gamma \frac{\partial L_{imp}(\Phi, w; \mathbb{D})}{\partial w}. \quad (20)$$

where $t = \{0, 1, \ldots, n\}$, n is the number of iterations.

4.2. T^{2}L: Trans-transfer Learning

We established two training agendas V_{imp} and V_{exp}, which could be used interchangeably, and are given three datasets, $\mathbb{D}_{aux}$, $\mathbb{D}_m$, $\mathbb{D}_s$. As discussed in the introduction, we assume that they will be used to train the model in a consecutive order, in that they are each more closing on to the target task. A total three rounds of optimization will be conducted, denoted as V, V', V'', the entire learning process can be written as:

$$\mathfrak{A} \mapsto V''(V'(V(\mathfrak{A}; \mathbb{D}_{aux}); \mathbb{D}_m); \mathbb{D}_s). \quad (21)$$

where V and V' constitutes the conventional transfer learning stage and V'' is the further adaptation stage. We refer to this framework for few-shot learning as *Tran-transfer Learning* (T^{2}L). V'' is based on few exemplars, traditionally trained model would incur over-fitting immediately on this task therefore no performance improvement can be captured. As a result, almost all traditionally trained models would only have two means to incorporate few-shot data, which is either weight imprinting [30] or fine-tuning [11] of the last layer. We will show in the experimental section that this direct incorporation of target data forces the feature extractor to be oblivious of target domain, therefore only provides limited generalization ability. In order to achieve EAD, a novel re-configuration and re-training strategy is introduced later base-on the aforementioned implicit class representation which is an extension of the model from phase one.

4.3. T^{1}L: Phase One

Tradition transfer learning solutions have two limitations: first, they do not address the nature of few-shot learning, which demands the ability to work with very few samples. Second, they can't adapt on few-shot data generally due to over-fitting. To address the first problem, in the field of few-shot learning, various techniques have been proposed to deal with this extreme scarcity of samples, notably, metric-based learning, which performs well in classification tasks.

In the proposed transfer learning phase one, there are *four* variants of combination of training agenda, as shown in Tab. 1, where 'I' is for implicit, and 'E' is for explicit. Note that the two initial letters represent the training agenda applied on $\mathbb{D}_{aux}$ and $\mathbb{D}_m$. For instance, 'II' corresponds to

$$\mathfrak{A} \mapsto V_{imp}(V_{imp}(\mathfrak{A}; \mathbb{D}_{aux}); \mathbb{D}_m), \quad (22)$$

which is in fact the pre-training agenda adopted by [30] where the linear softmax classifier can be seen as a case of Implicit Metric-model which measures the "distance" with the negation of dot-product. In Section 5, we will show that this re-use of the feature extractor only achieves sub-optimal performance because it is not deeply adapted. The difference between implicit class representation and explicit class representation is in the way w is determined, *i.e,* $\mathfrak{A}_{exp}$ can be seen as a regularized version of $\mathfrak{A}_{imp}$ which ground w with imprinting. A number of studies on few-shot learning support that explicit representation is more effective when predicting results directly from few samples [36]. However on many-shot data, implicit representation demonstrates a more general representation power. There seems to be a trade-off between few-shot and many-shot regime, *i.e,* the hyper-parameter k is entangled from training to testing.

Table 1
four different variants of training agenda for phase one

II	$V_{imp}(V_{imp}(\mathfrak{A};\mathbb{D}_{\text{aux}});\mathbb{D}_{\text{m}})$
IE	$V_{exp}(V_{imp}(\mathfrak{A};\mathbb{D}_{\text{aux}});\mathbb{D}_{\text{m}})$
EE	$V_{exp}(V_{exp}(\mathfrak{A};\mathbb{D}_{\text{aux}});\mathbb{D}_{\text{m}})$
EI	$V_{imp}(V_{exp}(\mathfrak{A};\mathbb{D}_{\text{aux}});\mathbb{D}_{\text{m}})$

We conduct the following theoretical analysis to help understand this relationship. We use a special case where a binary classification with two classes a and b being considered, and the objective for optimization can be expressed as

$$J = \sum_{x\in\mathbb{D}_q, y=a} \left[-\|f_\Phi(x) - \mathbf{c}_a\|^2 + \|f_\Phi(x) - \mathbf{c}_b\|^2) \right]. \quad (23)$$

Let $\alpha = \|f_\Phi(x) - \mathbf{c}_a\|^2 - \|f_\Phi(x) - \mathbf{c}_b\|^2$, the expected accuracy is $\mathrm{E}[\alpha > 0]$, from one-sided Chebyshev's inequality, follows that:

$$\mathrm{E}[\alpha > 0] \geq \frac{\mathrm{E}[\alpha]^2}{\mathrm{E}[\alpha]^2 + \mathrm{D}^2[\alpha]}. \quad (24)$$

the lower bound of Eq.24 is proved to have the following:

Theorem 1. *Use* $\boldsymbol{\mu}_c \triangleq \mathbb{E}_{\mathbf{x}}[f_\Phi(\mathbf{x}) \mid Y(\mathbf{x}) = c]$ *and* $\boldsymbol{\mu} \triangleq \mathbb{E}_c\left[\boldsymbol{\mu}_c\right]$ *to denote the expected average of the output feature,* $\Sigma_c \triangleq \mathbb{E}_{\mathbf{x}}\left[\left(f_\Phi(\mathbf{x}) - \boldsymbol{\mu}_c\right)^2 \mid Y(\mathbf{x}) = c\right]$ *and* $\Sigma \triangleq \mathbb{E}_c\left[\left(\boldsymbol{\mu}_c - \boldsymbol{\mu}\right)^2\right]$ *to denote the variance of the class identity under different expectation. The lower bound of the generalization error follows that*

$$\mathrm{E}[\alpha > 0] \geq \frac{4\,\mathrm{Tr}\,(\Sigma)^2}{4\,\mathrm{Tr}\,(\Sigma)^2 + 8(1+\frac{1}{k})\,\mathrm{Tr}\left(\Sigma_c\left(\left(1+\frac{1}{k}\right)\Sigma_c + 2\Sigma\right)\right)}. \quad (25)$$

when k is small, which few-shot learning stipulates, k serves as a trade-off factor between Σ and Σ_c, *i.e*, $\left(1+\frac{1}{k}\right)\Sigma_c + 2\Sigma$. It can be interpreted that a specific k is able to control the lower bound of generalization in that the denominator of Eq.25 is minimized when k fits a specific configuration of Σ and Σ_c. This relationship is locked during training where Σ and Σ_c are formed. When using Implicit Metric-models during training, k amounts to the total number of intra-class samples which is many-shot. It suggests that an explicit class representation directed from k samples, are able to maximize the generalization performance of the pre-training. Thus we adopt 'EE' as the combination of pre-training in this paper and we will validate this choice using empirical evaluation. The phase one learning of our approach is written as:

$$\mathfrak{A} \mapsto V_{exp}(V_{exp}(\mathfrak{A};\mathbb{D}_{aux});\mathbb{D}_m). \quad (26)$$

As we will demonstrate in the experiments, the dual metric-based learning strategy, significantly boosts the performance, we refer to this technique as Metric-based Infusion (MI). Model obtained by simply applying phase one is referred to as $\mathrm{T}^1\mathrm{L}$ in this paper whose variants are able to work with naive incorporation of novel classes which are evaluated in the ablation study.

4.4. T^2L: Phase Two

Conventional Transfer Learning techniques adapt knowledge extracted from source domains and find its way into the target domain. Both the target domain and source domain have a sufficient amount of data is a general assumption made by most of the previous approaches [1, 57]. This assumption only holds for $\mathbb{D}_{\text{aux}}$ and $\mathbb{D}_{\text{m}}$, which are many-shot.

In phase two, the target task is few-shot where only the support set are attainable during training. Previous adaptation approaches either do not extend to few-shot, in which case parameter imprinting is a naive measure which can be used to incorporate $\mathbb{D}_{\text{s}}$, or do they have many-shot and shallow adaptation. However, the essence of transfer learning lies in deep adaptation to the end task in order to achieve ideal performance. Moreover the ultimate goal of transfer learning is to solve a multitude of visual learning tasks by obtaining a completely adapted feature extractor. Therefore we propose Extended Adaptation. Clearly, adapting to $\mathbb{D}_{\text{s}}$ requires a V_{imp} training scheme which optimize the backbone architecture and the class parameter w at the same time:

$$\mathfrak{A} \mapsto V_{imp}(\mathfrak{A};\mathbb{D}_{\text{s}}). \quad (27)$$

However, the extremely low sample size is problematic, *i.e.,* even a few steps of fine-tuning on such small samples would be prone to over-fitting and causes deterioration of performance in generalization. As stated in the introduction, the fundamental reason for over-fitting under few-shot, is due to misguided optimization which inclines to emphasize noise. Conversely, suppressing noise means strict amplifying of signals. Unlike approaches introduced in explainable-learning literatures [58], where the classification result y is directly used as guidance for lower-layer analysis, we can't directly use supervision information y here in that $\mathbb{D}_{\text{s}}$ is too small therefore bias-prone. Instead the amplifying of l2-norm is adopted as a more general guidance of better signal-noise ratio since the features are normalized.

$$L_{sel}(\Phi) = -\sum_j \sum_k \left|\left|f_\Phi(x_j)_k\right|\right|^2, \quad (28)$$

where k is the index of output channels. This loss function is used to identify the importance of each layer of the activation in the same principle as CAAM [34], except that the original paper is solely intended for the last layer. The reason why we use CAAM instead of class-aware methods is that when it's shifting to novel classes, the information embedded in the pre-trained network becomes class-agnostic. But since the domain distance is small towards $\mathbb{D}_{\text{m}}$, larger branches of the model are still domain relevant and potentially discriminative. We intent to include them in the reconfigured model, and let the final optimization procedure to tune their weights. We use back propagation of L_{sel} to

Algorithm 1 BinarySearch.

Require: ρ denotes the hyper-parameters controls the global parameter selection rate; $\mathbb{D}_s$ denotes the support set of the meta-testing stage; ϵ is the error tolerance; $lower$ is the starting lower-bound, $upper$ is the starting upper-bound.

Calculate selection loss $L_{sel}(\Phi; \mathbb{D}_s)$ using Eq.28;
Back-propagate L_{sel} and calculate z and α respectively;
Let $\omega \leftarrow upper$
while $\omega - lower > \epsilon$ **do**
  Select $\boldsymbol{\Phi} \leftarrow \{\boldsymbol{\Phi}_{j,k} \in \Phi | \alpha_{j,k} > \omega\}$;
  if $|\boldsymbol{\Phi}| \,/\, |\Phi| > \rho$ **then**
    $\omega \leftarrow (lower + \omega)/2$;
  else
    $lower \leftarrow (lower + \omega)/2$
  end if
end while
return ω as the determined threshold.

Algorithm 2 Pseudo code for phase two.

Require: n, ρ denotes the hyper-parameters controls the training epoch and the global parameter selection rate; γ denotes the learning rate; w denotes the weight vector of the classification layer; $\mathbb{D}_s$ denotes the support set of the meta-testing stage.

for i in $range(0, n)$ **do**
  if $mod(i, 10) = 0$ **then**
    $\Omega \leftarrow \text{BinarySearch}(\rho, lower, upper, \epsilon)$;
    Select $\boldsymbol{\Phi} \leftarrow \{\boldsymbol{\Phi}_{j,k} \in \Phi | \alpha_{j,k} > \Omega\}$;
  end if
  Calculate classification loss $L_{imp}(\Phi, w; \mathbb{D}_s)$ using Eq.18;
  Update the reconfigured model
    $\boldsymbol{\Phi}^{t+1} \leftarrow \boldsymbol{\Phi}^t + \gamma \Delta_{\boldsymbol{\Phi}} L_{imp}(\Phi, w; \mathbb{D}_s)$;
  Update classification weight
    $w^{t+1} \leftarrow w^t + \gamma \Delta_w L_{imp}(\Phi, w; \mathbb{D}_s)$;
end for
return (Φ, w) representing the trained classification model.

calculate the channel-wise importance score for each layer of the activation:

$$z_{k,l} = \frac{1}{WH} \sum_{i}^{W} \sum_{j}^{H} \frac{\partial L_{sel}}{\partial A_{ij}^{k,l}} . \tag{29}$$

$A^{k,l}$ is the matrix denoting the activation map of channel k in the l-th layer. From a VC-dimension stand point, a neural network's VC-dimension is roughly linear to the number of connecting weights. Therefore when the sample size is fixed, one way to alleviate over-fitting is to reduce parameter size. In light of that, the reconfiguration of the model should serve two purposes: first, it is to highlight the activations with more important scores during the tuning. Second, the parameter-size of the model should be largely reduced. It is straight forward to obtain the parameter-wise importance score using the channel-wise one:

$$\alpha_{j,k} = z_{k,l} \cdot z_{j,l+1}. \tag{30}$$

A threshold Ω is used for excluding a subset of Φ where $\alpha_{j,k} < \Omega$, so that the DoF of the model, which is measured by a ratio ρ, can be controlled. It can be seen the Ω is monotonically increasing *w.r.t* ρ, but a direct mapping from the later to the former is not clear. In this case a binary search strategy can be employed to approximate this relationship to a certain error tolerance ϵ within the range of $(lower, upper)$:

$$\Omega \leftarrow \text{BinarySearch}(\rho, lower, upper, \epsilon) . \tag{31}$$

The pseudo code of Eq.31 is listed in Algorithm 1. The mechanism of this search is to find the smallest Ω that makes $|\boldsymbol{\Phi}| \,/\, |\Phi| > \rho$ with sub-linear number of iterations. The tolerance ϵ is typically set at $1e-5$. Once Ω is obtained, the network can be reconfigured with

$$\boldsymbol{\Phi} := \{\boldsymbol{\Phi}_{j,k} \in \Phi | \alpha_{j,k} > \Omega\} . \tag{32}$$

Instead of updating Φ, a subset $\boldsymbol{\Phi}$ is updated in exchange while keeping the rest part of the parameter fixed. The final optimization procedure is conducted on loss function

$$L_{imp}(\boldsymbol{\Phi}, w^{t_0}; \mathbb{D}_s) \tag{33}$$

where w^{t_0} is initialized with the averaged class prototypes [30]. On the right of Fig.3 is the training process of phase two, corresponding pseudo code is listed in Algorithm 2. In this refined fine-tuning procedure, the training epochs can be as large as 200 by which more representative features will be discovered based on as few as one sample per-category but over-fitting is largely avoided. We refer to it as Feature Rediscovery (FR).

4.5. Absence of Meta-training Set

The framework we proposed here relies on a generic visual database working in tandem with a fine-grained meta dataset. However arguably, in most realistic situations, it will not be possible to establish such a meta dataset since it is pointed out data collection and annotation in fine-grained domains are both time-consuming and expensive [20, 21, 22]. Instead it is only safe to assume a generic visual database be available for most tasks. It would have very wide domain coverage but not class-wise coverage especially for the fine-grained categories. Thus Unseen categories will be encountered during the meta-testing stage which fits the definition of few-shot learning.

In this case, previous approaches would usually resort to domain generalization techniques [25, 59] and using one-phase training [36, 60]. But we believe the proposed two-phase transfer learning framework and the idea of EAD would be helpful in such situations considering the concept of 'in-domain' dataset is not a binary term when it comes to novel-classes. *E.g.,* depending on how it is defined, but domain distance measured between base and novel classes would hardly be zero [61].

But the thing about fine-grained domain is that is would probably overlap with a subset of the general domain. In order to apply the proposed framework, a subset of the base domain can be extracted and forming two disjoint training sets, each of which substitutes $\mathbb{D}_{\text{aux}}$ and $\mathbb{D}_{\text{m}}$ respectively. A

commonly seen approach to measure domain distance, the $\mathcal{H}$-divergence [62], is adopted. It is a practical replacement of the variational A-distance.

$$d_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) = 2\Big(1 - 2 \min_{h' \in \mathcal{H}} err(h')\Big) \tag{34}$$

where

$$err(h') = \frac{1}{N} \sum_{s \in \mathcal{D}_s \cup \mathcal{D}_t} |h'(s) - \mathbb{1}_{s \in \mathcal{D}_s}| \,. \tag{35}$$

The domain distance can be simply measured by the error produced by separating the two domains with an hypothesis h out of a hypothesis space $\mathcal{H}$ which minimizes the error. Here, we use simple multi-dimensional hyperplane as the hypothesis space and sigmoid loss function to find the optimal hypothesis. We want to construct a pseudo meta-training set that is as close as possible to the novel training set and as distant as possible to the generic dataset. Note that there will be combinatorially many choices to construct the meta-training set. Instead we select out classes one at a time base on the $\mathcal{H}$-divergence to the formed meta-training set and regroup until certain number of classes been included (which can be cross validated). We call this $\mathcal{H}$-based domain extraction. We will show in the experimental section that this technique allows the two-phase training framework to work in the absence of a directly given in-domain meta-training set and demonstrates advantages against prior arts.

5. Experiments

5.1. Datasets and Evaluation Protocols

First we describe the bucket of datasets and the evaluation protocols we use in this work. There are *three* fine-grained datsets which are used to investigate the performance of the proposed models: CUB Birds [4] which contains 200 categories of bird species and a total of 11,788 images, Stanford Dogs [5] which contains 120 categories of dogs and a total of 20,580 images, Stanford Cars [7] which contains 196 categories of car make models and a total of 16,185 images. On top of that, *two* generic visual databases ImageNet [39] and Places [40] are used as the auxiliary dataset for our experiment.

There are *two* experimental protocols exist in the FSFG field that considers a transfer learning setting. We empirically evaluate the proposed algorithm under both settings in comparison with previous works and demonstrate the effectiveness as well as the generalization capability of our approach.

- **Exp.A** [46]: The three datasets CUB, Dogs, Cars are randomly split into a meta-training set and a meta-testing set by category and evaluated respectively. The number of classes in each separation are summarized in Tab.2. There is no validation set reserved for this split. Accordingly, the training and validation split from the original split is used which allows the experiments to cross-validate its hyper-parameters. In this setting, a transfer learning based on **Places** is considered. The results reported on CUB-B, Dogs-B, Cars-B are 50-way, 30-way, 49-way respectively.
- **Exp.B** [24]: A experimental setting recently well received in the community [48, 50, 51]. The three datasets are split into a meta-training set, a validation-set and a meta-testing set by category and evaluated respectively. The number of classes in each separation are summarized in Tab.2. The validation-sets are used to cross-validate the hyper-parameters then the meta-testing set is used to evaluate the performance. In this setting, all results are evaluated on a 5-way 1-shot and 5-way 5-shot bases.

Table 2
Category split for three datasets. C_{total} denotes the total number of categories in a dataset, C_{m} the number of categories in $\mathbb{D}_{\mathrm{m}}$ and C_{s} the number of categories in $\mathbb{D}_{\mathrm{s}}$.

#	CUB-A	Dogs-A	Cars-A	CUB-B	Dogs-B	Cars-B
C_{total}	200	120	196	200	120	196
C_{m}	150	90	147	120	70	130
C_{s}	50	30	49	50	30	49

Exp.A evaluate the performance in larger scales (100-way and 50-way) than the usual 5-way settings. Both of the results are averaged over 200 episodes, with 15 queries each.

5.2. Implementation and Hyper-parameter Settings

5.2.1. Implementation details

For most of our experiments, a standard ResNet18 is used for the backbone, except for when we analyse generalization on deeper backbones, where a deeper Resnet50 is adopted. Input images are resized to 224×224, standard data-augmentation including random crop, flip and color jitter is used during training. For optimizer, SGD is adopted for all experiments. For $\mathbb{D}_{\mathrm{aux}}$, model is trained with an initial learning rate of 0.1, with nesterov momentum 0.9, scheduler step size 30, decay rate 0.1, a total 90 epochs. For $\mathbb{D}_{\mathrm{m}}$, an initial learning rate of 0.01 is used, scheduler step size 100, decay rate 0.1, a total 300 epochs, each epoch consists of 100 episodes. For $\mathbb{D}_{\mathrm{s}}$, model is trained with a fixed learning rate of 0.005, with momentum 0.4, damping 0.2, batch-size is set equal to test n-way.

5.2.2. Hyper-parameter settings

there are following hyper-parameters that are specific to our algorithm. Training epochs n, n', n'': n is the training epoch of V, and it is validated on the validation set of dataset $\mathbb{D}_{\mathrm{m}}$, which is the target domain of $\mathrm{T}^1\mathrm{L}$. n_1 is the training epoch of V', and it is also validated on the validation set of $\mathbb{D}_{\mathrm{m}}$. n_2 is the training epoch of V'' where the dataset is $\mathbb{D}_{\mathrm{s}}$ which has no validation set, therefore, we cross validate this hyper-parameter on the validation set of $\mathbb{D}_{\mathrm{m}}$, and then keep it fixed. Weight reconfiguration parameter ρ, which is the global ratio of weight selection in phase two of $\mathrm{T}^2\mathrm{L}$. It

is cross-validated on the validation set of $\mathbb{D}_m$, we empirically find that within certain range, the choice for this hyper-parameters does not have significant impact on the accuracy. Further more, ρ is cross-validated and appears to have very stable optimums across all 4 dataset we tested, therefore we set the hyper-parameter value of 0.15 to obtain the results. We will do further analysis on this hyper-parameter along with the over-fitting analysis in Section 5.6.

5.2.3. Running time

Our method introduces additional computation cost in the second phase of the training, while phase one and inference time is uninfluenced (as compared to the corresponding baseline algorithms [11, 36]) which are largely depending on the total size of the data pool. Phase two includes a model reconfiguration time T_1 and a deep adaptation time T_2. Here we report T_1 and T_2 as well as the inference time T_i for 15 samples. The results are averaged over 100 trials on a single nvidia V100 GPU.

	T_1	T_2	T_i
Exp.A	431ms	2083ms	125ms
Exp.B	889ms	4740ms	131ms

It can be seen that a relatively long time will be required on each new coming set of $\mathbb{D}_s$ comparing to the rapid inference time. However this cost is very affordable because in reality it typically does not need to frequently re-train on a new set of novel classes.

5.3. Classification Accuracy

There are a number of works [11, 23, 24] addressed the problem setting of FSFG, however most of these works does not consider transferring knowledge from related domains. As a result, these works usually went with training on limited data of $\mathbb{D}_m$, expecting it to generalize well on $\mathbb{D}_s$. This branch of works usually test the model on 5-way miniature scenarios. They provide general insight to the problem of few-shot learning. Another branch of works focus on the more realistic settings, where a larger classifier is considered, as well as the availability of generic visual databases. In order to gain more insight into the area of FSFG, we think it's necessary to compare algorithms from both of the branches. Therefore, in this work, we re-implement some of the state of art generic few-shot learning algorithms under the transfer learning setting.

- **Imprinting** [30] is a non-meta transfer learning approach proposed for FSFG. It uses centroids of the support-set directly as the segment of weights for novel classes.
- **Baseline** [11] is a traditional softmax-based linear classifier with fine-tuning of the linear weights which is empirically proved by Chen *et al.* to be competitive on few-shot learning.
- **Baseline++**[11] is an improved version of Baseline [11] that substitutes linear layer, instead uses cosine distance as the output of the classifier.
- **Prototypical Networks** [36] is one of the simplest yet most competitive metric-based approach which adopt the squared euclidean distance as the metric and the centroid of the feature vector as the prototype.
- **Matching Networks** [35] leverages a full context embedding module to maintain all samples from the support-set and uses the cosine distance as the metric.
- **FEAT** [60] is a generic few-shot learning model which uses attention-based feature level transformation layer to obtain better prototypes for the support-set.

And the following methods specialized for fine-grained domain are compared:

- **Piece-wise Mapping** [46] adopts a bilinear encoder and piecewise classifiers mapping module which was tailored for FSGF. The performance supersedes all predecessors but the dual feature extractor and the bilinear pooling resulted in much larger computation consumptions.
- **PABN** [48] targets at FSFG by proposing a low-rank pairwise alignment bilinear network to capture the nuanced differences between images for learning an effective distance metric.
- **BSNet [51]** proposes to learn more discriminative and less similarity-biased features from few shots of fine-grained images which showed improvements.
- **FOT [50]** proposes foreground object transformation module, in order to improve generalization on FSFG tasks with data augmentation.

For **Exp.A**, as shown in Tab.3, our method out-performs previous state-of-the-art model PCM [46] on all three datasets tested. For generic algorithms we re-implemented on this setting, it can be seen both the classical algorithms [11, 35, 36] and the more sophisticated algorithms such as [60] which leads the performance on general domains such as miniImagenet, comes up short in the fine-grained domains. Moreover, improvements can be observed in 1, 2, 5-shot settings across all three datasets, where it is maximized in 1-shot settings. The reason why the performance is more sensitive to FR in 1-shot settings, is likely due to the absence of intra-class data in 1-shot settings where inter-class difference is more crucial. On the contrary, when more inter-class samples are given, it becomes over-powering so that inter-class differences carry less weights.

Fig.4 are some randomly selected samples and their four nearest neighbors from **Exp.A** to illustrate the results of our algorithm both before and after the EAD is conducted. It can be seen that phase two re-orders the similarity score for most of the samples and it has the ability to identify distinguishing features across exemplars which explains why it is statistically better than T^1L.

As shown in Tab.4, it can be seen for the three datasets we tested, general-domain models (the first block) are significantly inferior than specialized models (second and third

Table 3
The top-1 accuracy with 95% confidence interval measured across all novel classes of the split from [46] on three different fine-grained datasets, CUB, Dogs and Cars respectively, under settings of **Exp.A**, averaged over 200 episodes 15 queries each. Highest average accuracy in each column is marked in bold. † indicates our re-implementation of the original algorithm base on ResNet18 and pre-trained on Places dataset. ‡ indicates the results reported in [46]. Note that results reported by [46] employs a double Alexnet. '-' denotes results not reported by previous works.

	CUB			Dogs			Cars		
N-shot	1	2	5	1	2	5	1	2	5
Imprint [30] †	32.23±1.26	42.69±1.33	53.82±1.61	31.87±1.36	42.19±1.58	53.18±1.86	30.26±1.55	41.78±1.58	52.10±1.83
Baseline [11] †	31.86±1.61	42.62±1.80	54.52±1.71	30.26±1.56	41.63±1.61	54.52±1.69	29.93±1.75	40.69±2.06	53.32±2.12
Baseline++ [11] †	34.26±1.72	44.58±1.96	53.57±2.31	33.21±1.55	43.15±2.01	52.72±1.71	32.22±1.70	42.92±1.86	51.97±1.81
Protonet [36] ‡	35.16±1.09	46.09±1.31	57.12±1.76	35.56±1.23	45.10±1.71	55.73±1.62	34.24±1.82	44.18±2.13	53.89±2.28
Matchingnet [35] ‡	37.16±2.06	45.16±2.13	55.82±2.56	34.11±2.01	43.92±2.16	53.12±2.46	32.86±2.31	43.32±2.04	52.92±2.16
FEAT [60]†	39.02±1.11	49.15±1.41	59.34±1.71	35.57±1.22	46.31±1.86	57.70±1.58	37.94±1.82	48.01±1.92	55.22±2.01
Piecewise Map [46]	42.10±1.96	-	**62.48±1.21**	28.78±2.33	-	46.92±2.00	29.63±2.38	-	52.28±1.46
T^{1}L-IE(ours)	37.26±1.56	47.69±2.02	57.52±2.57	35.26±1.76	46.39±1.72	56.42±2.06	34.16±2.11	45.69±1.66	57.92±2.51
+FR(ours)	40.12±1.96	49.42±2.12	59.21±2.74	36.46±1.76	47.73±1.71	57.08±2.18	36.05±2.36	36.52±2.10	58.71±2.19
T^{1}L-EE(ours)	40.26±2.17	49.69±1.91	60.52±2.31	37.36±1.74	49.39±1.86	59.42±2.31	35.26±2.08	46.29±2.36	58.52±2.46
***+FR*(ours)**	**43.42±2.01**	**51.32±2.56**	**62.01±2.27**	**38.96±1.90**	**50.73±2.12**	**61.08±2.56**	**37.05±1.86**	**47.70±2.09**	**60.12±1.92**

Table 4
Top-1 accuracy with 95% confidence intervals on **Exp.B**, measured across 5-way novel classes, averaged over 600 episodes 15 queries each. † indicates the results we re-implement under the same backbone and settings. ‡ indicates the results reported in [50]. The first block are generic few-shot learning methods. The second block are methods specialised in fine-grained domains. The third block are model variants using the proposed technique.

	CUB		Dogs		Cars	
N-shot	1	5	1	5	1	5
Baseline [11]‡	45.97±0.74	67.09±0.71	34.42±0.59	51.95±0.64	35.60±0.65	53.21±0.79
Baseline++ [11]‡	61.08±0.84	79.28±0.68	42.01±0.73	62.52±0.72	46.64±0.80	65.29±0.73
MatchingNet [35]‡	57.78±0.91	72.44±0.74	42.88±0.77	58.03±0.75	41.26±0.80	62.77±0.79
ProtoNet [36]‡	44.53±0.83	75.28±0.70	37.32±0.75	59.09±0.71	30.46±0.64	61.89±0.77
FEAT [60]†	55.60±0.89	77.64±0.68	45.41±0.76	63.51±0.62	52.84±0.80	68.65±0.44
PABN [48]	63.56±0.79	75.23±0.59	45.64±0.74	58.97±0.63	53.39±0.72	66.56±0.64
BSNet(R&C) [51]	65.89±1.00	80.99±0.63	51.06±0.94	68.60±0.73	54.12±0.96	73.47±0.75
FOT [50]	67.46±0.68	83.19±0.43	49.32±0.74	68.18±0.69	54.55±0.73	73.69±0.65
T^{1}L (ours)	67.28±1.19	81.43±0.92	50.10±1.11	67.73±1.2	53.30±1.11	71.90±1.51
***+FR*(ours)**	**71.04**±1.21	**83.44**±0.94	**52.12**±1.14	**70.83**±1.09	**56.80**±1.23	**74.10**±1.65

block). Our method out-performs state-of-the-art on the benchmark tested in **Exp.B**. It also shows that even without the further adaptation phase, T^{1}L based solely on Metric-based Infusion is alight under the best performance reported by previous works such as PABN [48], BSNet(R&C) [51] and FOT [50] under this setting which validate that dual metric-based learning is effective on this setting but still shows limitation when the model is not deeply adapted. The improvements from T^{1}L to T^{2}L can be observed across three datasets both 1-shot and 5-shot, especially for the CUB dataset which are around 3.8% and 2.1% at 1-shot and 5-shot cases respectively. It even out-performs the best performing previous work FOT which utilizes complex background extraction and deep data-augmentation techniques which introduces substantial computation burden. It shows deep adaptation could be the bottleneck for previously models.

5.4. No Meta-training Set

In the absence of well established 'in-domain' meta-training set, we compare methods from two categories. One category is the generic FSL models which are designed to apply for general few-shot learning problems [11, 36]. The other one is Cross-domain Few-shot models [25, 59], which are designed to bridge the gap between base and novel classes under significant domain shift.

For the first comparison, we integrate it into **Exp.A** and Imagenet train split becomes the only source of training data, the results will be detailed along with the ablation study in

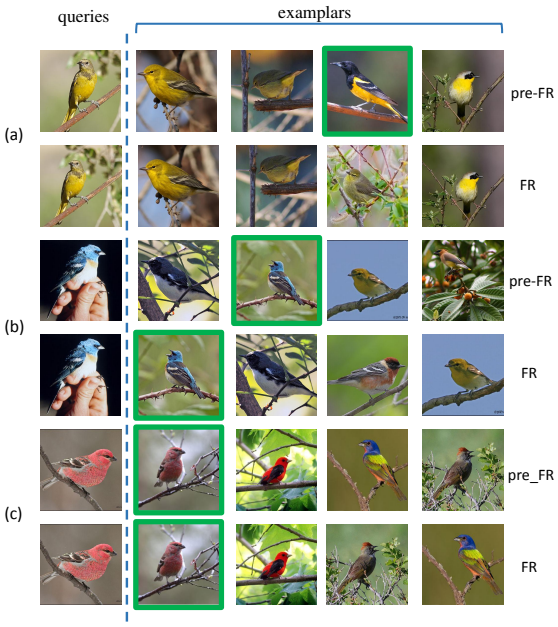


Figure 4: Some random selected category to demonstrate the classification results in Exp.A from the test set of the CUB birds dataset. Leftmost column is the query samples. Column 2-5 is the 4 nearest neighbours ordered by their similarity to the exemplar. For each group from a to c, the first row are the results before adaptation, the second row are the results after. Solid border indicates the ground truth if it appears in the list.

Section.5.5.

For the second comparison, we use the experimental setting of previous publications where miniImageNet is adopted as the base dataset and the validation split of CUB and Cars as the novel dataset. Then the results are obtained under 5-way bases. This experimental setting is with two considerations: firstly, we want to see how the portion of the obtained meta-training set effects the final results down to as few as 5 classes. Secondly, we want to compare results with some of the prior arts which are usually tested under 5-way settings, using the above datasets.

The results obtained across different C_{m} are shown in Figure 5. The results are compared with two recently proposed one-phase methods, namely FwT [25] and ATA [59]. It can be seen that the performance of our method peaked when $C_{\mathrm{m}} = 20$, which surpasses the best one-phase method ATA. This accounts for evidence of our method being potentially useful in more realistic situations where $\mathbb{D}_{\mathrm{m}}$ is not directly provided.

5.5. Ablation Study

There are a number of different ablated versions of the proposed algorithm we want to do further analysis. Generally there are different design choices in both stages of T^{2}L.

For phase one, as aforementioned, there are 4 different combination of training agenda can be employed by stage one. We propose in this paper that 'EE' is the best choice for end tasks that is under 1, 2, 5-shots. For phase two, the key for this stage is the model reconfiguration procedure. We ablate our model with random parameter selection under different global probability, namely 0.1, 0.3, 1.0. Note that selection with probability 1.0 is equivalent to all parameters enabled. On top of that, we want investigate a version of our algorithm where $\mathbb{D}_{\mathrm{aux}}$ and $\mathbb{D}_{\mathrm{m}}$ are used in a multi-task fashion, which proves that the successive training assumption we

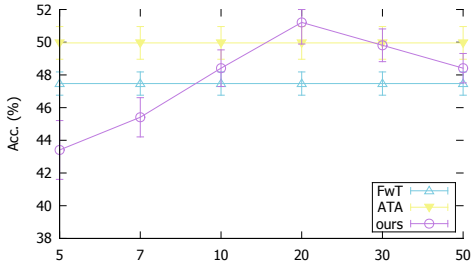


Figure 5: Accuracy with 95% confidence intervals on more realistic situation where $\mathbb{D}_{\mathrm{m}}$ is not directly provided but extracted based on domain distance with varied C_{m} sizes. Comparing methods are FwT [25] and ATA [59] which are one-phase based and whose performances are obtained on the entire training set all the time.

Table 5
Ablation study results using CUB and Dogs dataset from **Exp.A** with 95% confidence interval. T^{1}L has five different variants, II, IE, EI, EE, MT where MT is the multi-task version. For phase two adaptation we test random selection and FR, with different hyper-parameters, but all based on T^{1}L-EE models. In phase two we report results after 200 epochs and EE trained model to make the results non-trivial.

N-shot	1	2	5	1	2	5
T^{1}L-II	35.16±1.69	44.33±2.01	57.52±2.12	33.12±1.42	43.30±1.89	54.02±1.73
T^{1}L-IE	37.26±1.72	47.69±2.29	57.52±2.36	35.26±1.39	46.39±2.07	56.42±2.29
T^{1}L-EI	36.21±2.09	45.12±1.99	57.76±2.30	34.67±1.68	45.54±2.20	55.47±2.02
T^{1}L-EE	40.26±2.03	49.69±2.32	60.11±2.54	37.36±1.90	49.39±2.12	59.42±2.27
T^{1}L-MT	31.86±1.12	41.45±1.69	53.68±1.90	29.26±1.22	40.68±1.79	52.50±2.21
random0.1	40.16±2.09	49.45±1.80	60.28±1.83	38.23±1.84	49.69±2.01	59.12±2.03
random0.3	39.86±1.95	48.15±1.90	59.63±1.84	37.66±1.57	47.92±1.76	58.52±2.12
random1.0	37.16±2.03	45.92±1.98	56.72±2.02	35.20±1.65	44.59±1.67	55.23±2.01
FR$^-$0.15	38.01±2.01	47.45±1.98	59.86±1.92	35.21±1.67	46.87±1.98	59.82±2.21
FR0.3	41.96±2.12	50.01±1.94	61.68±1.80	37.16±1.93	49.69±2.02	61.32±2.12
FR0.15	**43.42±2.01**	**51.32±2.56**	**62.01±2.27**	**38.96±1.90**	**50.73±2.12**	**61.08±2.56**

Table 6
Impact of auxiliary dataset, using CUB and Dogs from **Exp.A** with 95% confidence interval. Four groups divided horizontally indicate results obtained with no auxiliary data, Places, ImageNet and no meta-training set respectively.

	N-shot	1	2	5	1	2	5
w/o	T^{1}L-EE	28.16±1.89	37.33±2.01	47.52±2.10	26.92±1.78	38.31±1.67	49.05±1.65
	T^{2}L-EE	31.22±1.89	39.25±2.09	49.20±2.12	28.06±1.57	39.80±1.95	51.43±2.03
Places	T^{1}L-EE	40.26±1.79	49.69±1.98	60.11±2.12	37.36±1.92	49.39±2.09	59.42±2.12
	T^{2}L-EE	43.42±2.01	51.32±2.56	62.01±2.27	38.96±1.90	50.73±2.12	61.08±2.56
Imgnet	T^{1}L-EE	50.89±2.16	60.02±2.24	70.50±2.69	45.08±1.99	55.46±2.12	65.41±2.40
	T^{2}L-EE	**54.10±2.10**	**62.25±2.20**	**71.87±2.42**	**46.12±1.92**	**56.66±1.81**	**66.92±1.78**
no-meta	*no*-extract	29.56±1.19	37.62±1.28	48.30±1.42	26.31±1.12	35.39±1.29	42.41±1.32
	$\mathcal{H}$-extract	**38.91±1.71**	**49.37±1.56**	**57.01±1.37**	**33.32±1.50**	**42.73±1.42**	**53.12±1.36**

made is valid.

Tab.5 presents results for ablation study on **Exp.A** (the same setting as results reported in Tab.3). Note that some of the training schemes over-fit from the first epoch, therefore their best performing model is the initialization configuration. Here we unanimously use 200 epochs to make the results non-trivial. It can be seen that Metric-based Infusion is beneficial to 1, 2 and 5-shot scenarios where its sensitivity is in descending order, which is a phenomenon reported by most meta-learning based algorithms, *i.e.*, meta-learning are capable of rendering the model to be more sensitive to few-shot data, but its effectiveness will decrease when the shots get larger. For the effectiveness of phase two, improvements can be observed in all experiments reported in Tab.5 which again indicates that FR is most sensitive in 1-shot scenarios and remains helpful through out all settings evaluated. The performance of T^1L-MT, the multi-task version where both datasets are trained simultaneously are inferior than the proposed approach, which validates the hypothesize that different datasets should train successively base on their relative domain-distance to the target task.

Next, we analyse the impact of different auxiliary datasets since our performance is partly attributed to the pre-training on auxiliary data. Tab.6 presents results obtained based on **Exp.A** with different settings of auxiliary data, *i.e.*, without $\mathbb{D}_{\text{aux}}$, with Places and with ImageNet respectively. It can be shown that even the auxiliary data are from distant domains, thanks to their huge volume and diversity, the performances are boosted on the end task by a large margin, *e.g.*, 22.73% for *1-shot* CUB in the case of ImageNet. For Places dataset, the gap narrows to 11.10%. The reason is partly due to that places dataset are from a more distant domain of CUB. Also even with a commensurate number of total images, the places dataset affords much less classes and a large inter class diversity, which is proved to be a negative factor for few-shot learning by previous works [63].

The last block of Tab.6 labeled 'no-meta' presents results obtained in the absence of no meta-training set. With the previously introduced $\mathcal{H}$-based domain extraction technique, we are able to implement our model with a complete two-phase training procedure, which can be seen significantly out-perform the one-phase version of the same algorithm which uses the Imagenet train split all at once. It can be seen that even in the absence of meta-training set, it out-performs the version where only in-domain meta-training set is used.

5.6. Over-fitting Tendency

For phase two of T^2L, we proposed an adaptive reconfiguration technique to facilitate FR fine-tuning. The number of epochs for this procedure is fixed to 200 through out all our experiment. This value is not meticulously tuned for each setting, but is shown to generally work well across all datasets during the validation. We want to further analyse here to see how different hyper-parameter choices of ρ influence the training loss, as well as the generalization performance, after which we can have a general understanding about how the above variations affect the over-fitting tendency, and how does the model reconfiguration alleviates the problem we emphasized in Section 1.

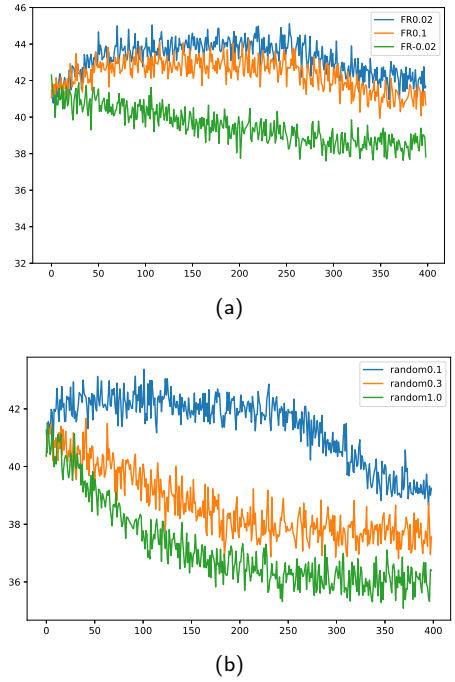


Figure 6: Overfitting analysis on CUB-B. Results are recorded at the end of each epoch, and averaged over 10 episodes each sampled 20 queries. In group (a), "FR#" indicates the results obtained by the proposed Feature Rediscovery algorithm with ρ set to #, while "FR$^-$#" is the reverse version where the least important features are considered. In group (b), "Random#" indicates random weight selection with a probability of #.

It can be seen from Fig.6 that even random enabling of parameters can avoid over-fitting when only 10% of the weights are chosen. However random version does not have observable positive impact on the results. When the selection rate gets higher, the over-fitting gets more severe and appears more pre-maturely during training.

For our proposed approach, when the hyper-parameter ρ is set to 0.15. When ρ is set to 0.3, 30% of the weights get enabled, in which case the model still shows improvement but less than the former setting and stronger tendency to over-fit when the training continues. On the other hand, when ρ is set at 0.15 but reverse the direction of the importance value, we refer to as FR$^-$0.15. Even though the global selection rate is similar to FR0.15, the model demonstrates poor performance even worse than the random version, and over-fitting tendency very prematurely, which suggests fine-tuning the least important set of weights has strictly negative impact on the performance which proves the validity of the importance score. It also coincide with the theory that the over-fitting is not simply dictated by the number of parameters and degree of freedom, but also on the intrinsic quality of the model.

Table 7
The generalization performance of the proposed algorithm on a deeper backbone which is ResNet50 with 95% confidence interval, using CUB and Dogs from **Exp.A**.

N-shot	1	2	5	1	2	5
T^1L-IE	39.76±1.40	50.02±1.65	59.78±1.87	37.86±1.23	48.95±1.49	58.02±1.76
+FR	42.32±1.95	51.42±1.86	60.21±1.80	38.46±1.28	49.73±1.49	59.56±1.68
T^1L-EE	42.65±1.98	51.29±2.09	60.52±2.29	38.36±1.68	50.39±1.96	59.42±2.02
+FR	**45.20**±2.06	**53.04**±1.87	**62.71**±1.68	**39.96**±1.37	**51.30**±1.62	**60.78**±1.98

5.7. Deeper Backbones

To investigate the generalization performance of our algorithm on deeper network architectures, we conduct ablation study on Resnet50. In this experiment, we only compare the performance of our algorithm's different ablated version, because most previous works didn't report performance on Resnet50 backbone, and re-implement them can be restricted by GPU memory. For instance, Relation Networks [37] with a Resnet50 backbone, could take up more than 100GB gpu memory, since we need to train it 100-way 5-shot, in order to achieve a reasonable performance.

Tab. 7 shows the performance of the proposed approach on CUB dataset. It can be seen that 'EE' training agenda for phase one produce better results than any other ablated version of the algorithm. And feature rediscovery procedure in phase two improves the performance of the learned model for a various ablated versions of the algorithms. Note that the hyper-parameters n and ρ are exactly the same with that of ResNet18, which is set to 200, 0.15 respectively.

5.8. Visualization

Fig.7 visualizes the percentage of weights been selected by our approach on each layer when ρ is set to 0.15. Note that this hyper-parameter can be seen as a global ratio for the parameter reconfiguration. But when it comes to each layer of the feature extractor locally, the selection rate varies. It can be seen for most convolution layers of the model, the probability of the weight been chosen for fine-tuning is between 0.1 and 0.2. Compared to random selection with a probability of 0.1, in which case the network over-fitting is large alleviated, our algorithm brings performance improvement which random selection fails to (Tab.5).

Fig.8 are the Class Activation Map [58] visualization on five randomly selected samples from the validation set. The hotter area indicates the image area that have bigger impact on the output of the last convolution layer. It can be seen the hot map is generally around the same area but more compact which suggests that focus of the model has subtly shifted after the Feature Rediscovery.

Fig.9 presents some qualitative results by visualizing the feature extractor learning by our approach in the 2D space. The dots with the same color denote the features generated from different validation samples of the same category in N. As shown in the figure, the features generated by exhibit better category-separability and more centralized intra-category aggregation.

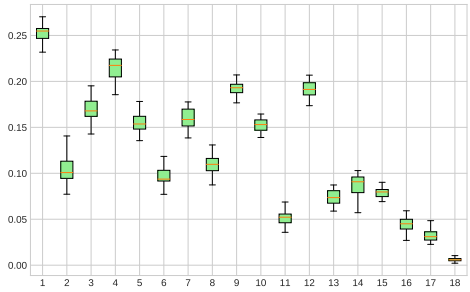


Figure 7: Visualization of the local selection rate in each layer of ResNet18 with a global selection rate set to 0.15. It can be seen that the average ratio of been selected by FR algorithm is between 0.1 and 0.2, except for the first and last layer.

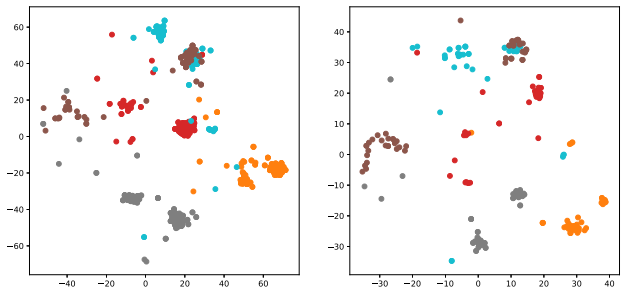
Figure 8: Class Activation Map visualization of our model for random samples from Exp.A. The first row are the original image. The second and third rows are the heatmap generated by T^1L and T^2L respectively. It can be seen the focus of the model has shifted after the Feature Rediscovery.

6. Discussion

Leveraging generic visual database for FSFG is a simple and intuitive idea which dates back to [30]. But the properties of fine-grained domain are not well explored. We contribute to this idea by proposing a two-phase learning framework and $\mathcal{H}-based$ domain extraction technique which allow fine-grained classification to excel on generic training data.

Another of our key novelties lies in phase two, where the model is fully adapted to few-shot data. Meta-learning intend to optimize the learning process in order to accommodate few-shot data. In concept, the outer loop and inner loop of meta-learning can be reckon as a two stage learning process. However their adaptation procedures are proven to be shallow [33]. We try to look at this problem from a different angle which is motivated by the correlation between parameter space and vc-dimension. We found that a downsized parameter space can effectively suppress over-fitting. And moreover an adaptive model reconfiguration strategy can be employed to have very positive effect.

In the future, it appears promising to use transfer learning techniques and train the model in an incremental fashion



(a) Pre-adaptation (b) Post-adaptation

Figure 9: Visualization of the category classifiers generated by T^1L and T^2L in 2D space by t-SNE. Each dot denotes a generated classifier and different colors represent different categories. This visualization is based on CUB Birds. For each category, thirty samples are shown, each of which is obtained via the feature embedding module.

ultimately improving generalization on fine-grained novel classes. On the other hand, adaptive model reconfiguration can be used as a technique to enhance certain capability of the model, which is worth further exploring. In addition, for the problem of training on very few samples, we only explored refinement on one factor which is the model's configuration. For future work, other aspects of the training process, such as the loss function itself could potentially be revamped by tailoring to the few-shot domain.

7. Conclusion

In this paper, we addressed the practical and challenging few-shot fine-grained visual categorization problem. We argue that a direction for improving generalization performance is to explore Extended Adaptation. To this end, we propose Tran-transfer Learning, which achieves EAD by learning three levels of datasets consecutively. Base on our proposal that metric-based models can be generalized and have implicit class representations with which explicit representation can be used inter-changeably, the chain of knowledge transfer is complete by composing of two learning phases concatenated together. The key novelty of our work lies in how to adapt the model to the few-shot task which is extremely low on data and prone to over-fitting. By dissecting the roots of this problem, we propose a CAAM-guided model reconfiguration strategy by back-propagating most relevant features which works adaptively with the training process. The over-fitting is largely mitigated and a further improvement of performance up to 3.2% can be achieved. Through comprehensive experiments on three popular fine-grained image datasets and two different transfer learning experimental settings, our method exhibits promising results.

References

[1] Cui, Y., Song, Y., Sun, C., Howard, A., Belongie, S.. Large scale fine-grained categorization and domain-specific transfer learning. In: CVPR. 2018, p. 4109–4118.

[2] Increasingly specialized generative adversarial network for fine-grained visual categorization. Knowledge-Based Systems 2021;232:107480.

[3] Song, K., Wei, X., Shu, X., Song, R., Lu, J.. Bi-modal progressive mask attention for fine-grained recognition. IEEE TIP 2020;29:7006–7018.

[4] Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.. The caltech-ucsd birds-200-2011 dataset. Tech. Rep.; 2011.

[5] Khosla, A., Jayadevaprakash, N., Yao, B., Li, F.F.. Novel dataset for fine-grained image categorization: Stanford dogs. In: Proc. CVPR Workshop on FGVC; vol. 2. 2011,.

[6] Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., et al. The inaturalist species classification and detection dataset. In: CVPR. 2018, p. 1–10.

[7] Krause, J., Stark, M., Deng, J., Fei-Fei, L.. 3d object representations for fine-grained categorization. In: IEEE ICCV Workshops. 2013, p. 554–561.

[8] He, K., Zhang, X., Ren, S., Sun, J.. Deep residual learning for image recognition. In: CVPR. 2016, p. 770–778.

[9] Krizhevsky, A., Sutskever, I., Hinton, G.E.. Imagenet classification with deep convolutional neural networks. In: NeurIPS. 2012, p. 7–15.

[10] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. Going deeper with convolutions. In: CVPR. 2015, p. 1–9.

[11] Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C.F., Huang, J.B.. A closer look at few-shot classification. In: ICLR. 2019,.

[12] Sun, Q., Liu, Y., Chua, T.S., Schiele, B.. Meta-transfer learning for few-shot learning. In: CVPR. 2019, p. 403–412.

[13] Lee, K., Maji, S., Ravichandran, A., Soatto, S.. Meta-learning with differentiable convex optimization. In: CVPR. 2019, p. 10657–10665.

[14] Wu, J., Dong, N., Liu, F., Yang, S., Hu, J.. Feature hallucination via maximum a posteriori for few-shot learning. Knowledge-Based Systems 2021;225:107129.

[15] Fu, J., Zheng, H., Mei, T.. Look closer to see better: recurrent attention convolutional neural network for fine-grained image recognition. In: CVPR. 2017,.

[16] Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.. Adversarial discriminative domain adaptation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. IEEE Computer Society; 2017, p. 2962–2971.

[17] Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.J.. A kernel method for the two-sample-problem. In: Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006. MIT Press; 2006, p. 513–520.

[18] Li, W., Duan, L., Xu, D., Tsang, I.W.. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. IEEE Trans Pattern Anal Mach Intell 2014;36(6):1134–1148.

[19] Yan, H., Ding, Y., Li, P., Wang, Q., Xu, Y., Zuo, W.. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. IEEE Computer Society; 2017, p. 945–954.

[20] Byrd, J., Lipton, Z.. What is the effect of importance weighting in deep learning? In: ICML. 2019, p. 872–881.

[21] Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: CVPR. 2020, p. 9719–9728.

[22] Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.. Class-balanced loss based on effective number of samples. In: CVPR. 2019, p. 9268–9277.

[23] Li, X., Yu, L., Fu, C.W., Fang, M., Heng, P.A.. Revisiting metric learning for few-shot image classification. Neurocomputing 2020;.

[24] 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: CVPR. 2019, p. 7260–7268.

[25] Tseng, H.Y., Lee, H.Y., Huang, J.B., Yang, M.H.. Cross-domain few-shot classification via learned feature-wise transformation. In: ICLR. 2020,.

[26] Oreshkin, B., López, P.R., Lacoste, A.. Tadam: Task dependent adaptive metric for improved few-shot learning. In: NeurIPS. 2018, p. 721–731.

[27] Ravi, S., Larochelle, H.. Optimization as a model for few-shot learning. In: ICLR. 2016,.

[28] Finn, C., Abbeel, P., Levine, S.. Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML. 2017, p. 1126–1135.

[29] Nichol, A., Schulman, J.. Reptile: a scalable metalearning algorithm. arXiv preprint arXiv:180302999 2018;2.

[30] Qi, H., Brown, M., Lowe, D.G.. Low-shot learning with imprinted weights. In: CVPR. 2018, p. 5822–5830.

[31] Xia, H., Lan, Y., Song, S., Li, H.. A multi-scale segmentation-to-classification network for tiny microaneurysm detection in fundus images. Knowledge-Based Systems 2021;226:107140.

[32] Thrun, S., Pratt, L.. Learning to learn. Springer Science & Business Media; 2012.

[33] Raghu, A., Raghu, M., Bengio, S., Vinyals, O.. Rapid learning or feature reuse? towards understanding the effectiveness of maml. ICLR 2019;.

[34] Baek, K., Lee, M., Shim, H.. Psynet: Self-supervised approach to object localization using point symmetric transformation. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, NY, USA. 2020, p. 10451–10459.

[35] Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In: NeurIPS. 2016, p. 3630–3638.

[36] Snell, J., Swersky, K., Zemel, R.. Prototypical networks for few-shot learning. In: NeurIPS. 2017, p. 4077–4087.

[37] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.. Learning to compare: Relation network for few-shot learning. In: CVPR. 2018, p. 1199–1208.

[38] Zhang, C., Cai, Y., Lin, G., Shen, C.. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In: CVPR. 2020, p. 12203–12213.

[39] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. ImageNet large scale visual recognition challenge. ICCV 2015;.

[40] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.. Learning deep features for scene recognition using places database. In: NeurIPS; vol. 1. 2014,.

[41] Huh, M., Agrawal, P., Efros, A.A.. What makes imagenet good for transfer learning? In: NeurIPS WorkShop. 2016,.

[42] Yosinski, J., Clune, J., Bengio, Y., Lipson, H.. How transferable are features in deep neural networks? In: NeurIPS. 2014,.

[43] Sun, C., Shrivastava, A., Singh, S., Gupta, A.. Revisiting unreasonable effectiveness of data in deep learning era. In: ICCV. 2017,.

[44] Cai, S., Zuo, W., Zhang, L.. Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization. In: ICCV. 2017,.

[45] Cui, Y., Zhou, F., Lin, Y., Belongie, S.. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In: CVPR. 2016,.

[46] Wei, X.S., Wang, P., Liu, L., Shen, C., Wu, J.. Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples. IEEE TIP 2019;28(12):6116–6125.

[47] W, L., J, X., J, H., L, W., G, Y., Luo, J.. Distribution consistency based covariance metric networks for few-shot learning. In: AAAI. 2019,.

[48] Huang, H., Zhang, J., Zhang, J., Xu, J., Wu, Q.. Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification. IEEE Transactions on Multimedia 2020;.

[49] Sun, X., Xv, H., Dong, J., Zhou, H., Chen, C., Li, Q.. Few-shot learning for domain-specific fine-grained image classification. IEEE TIE 2020;:1–1.

[50] Wang, C., Song, S., Yang, Q., Li, X., Huang, G.. Fine-grained few shot learning with foreground object transformation. Neurocomputing 2021;466:16–26.

[51] Li, X., Wu, J., Sun, Z., Ma, Z., Cao, J., Xue, J.. Bsnet: Bi-similarity network for few-shot fine-grained image classification. IEEE Trans Image Process 2021;30:1318–1331.

[52] Hariharan, B., Girshick, R.. Low-shot visual recognition by shrinking and hallucinating features. In: ICCV. 2017, p. 3018–3027.

[53] 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. 2018, p. 7278–7286.

[54] Chen, Z., Fu, Y., Wang, Y.X., Ma, L., Liu, W., Hebert, M.. Image deformation meta-networks for one-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019, p. 8680–8689.

[55] Pahde, F., Jähnichen, P., Klein, T., Nabi, M.. Cross-modal hallucination for few-shot fine-grained recognition. arXiv preprint arXiv:180605147 2018;.

[56] Tang, L., Wertheimer, D., Hariharan, B.. Revisiting pose-normalization for fine-grained few-shot recognition. In: CVPR. 2020, p. 14352–14361.

[57] Lin, T.Y., RoyChowdhury, A., Maji, S.. Bilinear cnn models for fine-grained visual recognition. In: ICCV. 2015,.

[58] Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., Torralba, A.. Learning deep features for discriminative localization. In: CVPR. IEEE Computer Society; 2016, p. 2921–2929.

[59] Wang, H., Deng, Z.. Cross-domain few-shot classification via adversarial task augmentation. In: Zhou, Z., editor. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021. 2021, p. 1075–1081.

[60] Ye, H., Hu, H., Zhan, D., Sha, F.. Learning embedding adaptation for few-shot learning. In: CVPR. 2020, p. 8680–8689.

[61] Zhang, Y., Deng, B., Tang, H., Zhang, L., Jia, K.. Unsupervised multi-class domain adaptation: Theory, algorithms, and practice. IEEE Transactions on Pattern Analysis and Machine Intelligence 2020;:1–1.

[62] Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.. Analysis of representations for domain adaptation. In: Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006. MIT Press; 2006, p. 137–144.

[63] Sbai, O., Couprie, C., Aubry, M.. Impact of base dataset design on few-shot image classification. In: ECCV; vol. 12361. 2020, p. 597–613.