Regularising Knowledge Transfer by Meta Functional Learning

Pan Li[1] , **Yanwei Fu**[2*] and **Shaogang Gong**[1]
[1]Queen Mary University of London
[2]School of Data Science, and MOE Frontiers Center for Brain Science, Fudan University
{pan.li, s.gong}@qmul.ac.uk, yanweifu@fudan.edu.cn

Abstract

Machine learning classifiers' capability is largely dependent on the scale of available training data and limited by the model overfitting in data-scarce learning tasks. To address this problem, this work proposes a novel Meta Functional Learning (MFL) by meta-learning a generalisable functional model from data-rich tasks whilst simultaneously regularising knowledge transfer to data-scarce tasks. The MFL computes meta-knowledge on functional regularisation generalisable to different learning tasks by which functional training on limited labelled data promotes more discriminative functions to be learned. Moreover, we adopt an Iterative Update strategy on MFL (MFL-IU). This improves knowledge transfer regularisation from MFL by progressively learning the functional regularisation in knowledge transfer. Experiments on three Few-Shot Learning (FSL) benchmarks (miniImageNet, CIFAR-FS and CUB) show that meta functional learning for regularisation knowledge transfer can benefit improving FSL classifiers.

1 Introduction

The success of current deep architectures benefits a great deal on representation learning, in the sense of learning "big models" of richer representations for many tasks. Recent developments on self-supervised learning, or models trained on very large-scale data [Devlin *et al.*, 2018; Brown *et al.*, 2020], seem to suggest that powerful and universal representations could be learned for all tasks in all domains.

Given a universal feature extractor, can a good classifier for a particular task be effectively learned from only a few labelled examples of that task? Having a good universal representation does not guarantee fitting generalisable hypotheses of different individual tasks from a few labelled samples. For a Few-Shot Learning (FSL) task, many researchers had devoted their efforts in addressing the severe overfitting problem resulting in inferior classification accuracy and generalisation on novel categories [Ravi and Larochelle, 2016; Ye *et al.*, 2020]. Typical FSL settings [Chen *et al.*, 2019] assume that given a large amount of labelled data on source/base tasks, and few labelled data on target/novel tasks, a FSL algorithm can learn good hypotheses on novel tasks. Moreover, one may further consider Cross-Domain Few-Shot Learning (CD-FSL) when the source and target tasks are from significantly different semantic domains [Tseng *et al.*, 2020].

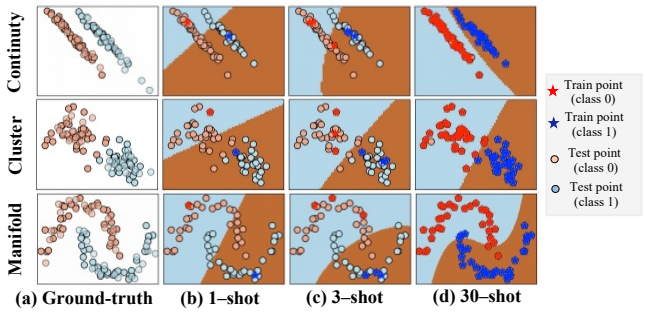

Figure 1: The illustration of hypotheses learned with k-shot data on binary classification tasks with continuity, cluster and manifold distributions. Plots(a) are the ground-truth data distributions, and (b-d) represent the hypotheses learned with 1/3/30-shot data. Without efficient data training, the hypotheses (plots(b)) fail to learn the ground-truth data distributions, whilst the hypotheses (plots(c-d)) are progressively capable to learn them with the increased regularisation knowledge deriving from the labelled data. Best viewed in color.

Given a learned representation from richly labelled data, we consider that the underlying data distribution should follow the *continuity*, *cluster*, and *manifold* assumptions, as in Semi-Supervised Learning (SSL) [Chapelle *et al.*, 2009]. Figure 1 illustrates this phenomenon from both SSL and supervised learning. Hypotheses learned from larger amount of examples (richer) are favoured than those trained by fewer examples. Moreover, good hypotheses should prefer geometrically simpler decision-boundaries and encourage points in the same cluster to have the same label. This should be a general principle for task-agnostic patterns of a hypothesis.

In a hypothesis/function space, we aim to learn gradually task-agnostic patterns of change in fitting hypotheses to training data from few to many labelled examples. In particular, the latent knowledge of task-agnostic patterns of change in a hypothesis fitting process is to be learned as a *functional*, estimated from a family of richly labelled data on source tasks that simultaneously satisfies new hypothe-

∗ Yanwei Fu is the corresponding author.

ses of the same/similar family of functional generalisable to learning new target tasks. To that end, we introduce a meta-learning strategy to learn this functional, called *Meta Functional Learning* (MFL).

Essentially, our MFL learns a functional regularisation on how to best fit new hypotheses on scarcely labelled novel tasks according to how to best fit hypotheses on richly labelled base tasks, thus imposing penalties (constraints) on excessive optimisation (overfit) in fitting the novel hypotheses. Particularly, given the task of learning a novel hypothesis from scarcely labelled data, our functional encourages *a process* of learning the hypothesis by approximating the learning process of richly labelled data, from which it favours to satisfy the underlying data distribution principles of continuity, cluster, and manifold. The functional from MFL captures model learning ***regularisation knowledge*** from source data and transfers it to guide the FSL of novel tasks. Our approach to knowledge transfer as learning regularisation (*how to learn*) differs fundamentally to other existing methods of knowledge transfer on *what to learn*, e.g., representations in FSL. Figure 1 illustrates our idea of MFL that learns a task-agnostic, transferable and generalisable functional, a function in the function space, to remit the overfitting problem in hypothesis optimisation given scarcely labelled data.

Specifically, we explore a meta-learning paradigm to learn a functional of meta-knowledge for learning process regularisation. That is, MFL first samples many functional episodes to craft a functional set of function pairs trained on few/many labelled data with a base classifier, e.g., a Logistic Regression (LR). MFL then minimises the distances between its predicted functions and target functions, achieving the meta-knowledge learning/transfer through functional regularisation. We formulate an iterative update strategy to connect a sequence of basic module blocks, forming an overall model by Iterative Update (MFL-IU). Our contributions are: (1) We formulate knowledge transfer in few-shot learning as a problem of transfer learning regularisation (how to learn) rather than knowledge transfer in representation (what to learn). This problem is solved by meta functional learning. (2) We introduce an iterative update strategy for meta functional learning that aims to gradually improve the classifier's learning ability through the transfer of functional regularisation. (3) We apply the meta functional learning to both the standard few-shot learning and the cross-domain few-shot learning problems. We provide comprehensive experiments on miniImageNet, CIFAR-FS and CUB to validate the effectiveness of MFL and MFL-IU in improving FSL by minimising model overfit.

2 Related Work

Model Transformation and Composition. Our investigation on knowledge transfer by functional regularisation is related to previous works on model transformation and composition, in particular, a model regression network with MLP architecture for learning a generic, category agnostic transformation from small-sample models to the underlying large-sample models [Wang and Hebert, 2016]. Subsequently, a MetaModelNet [Wang *et al.*, 2017] was proposed for transferring the model dynamic from head classes to tail classes in long-tail recognition problem. Functional gradient learning [Johnson and Zhang, 2019] was explored to learn the composition of functions and an incremental strategy was adopted for gradually learning a generator network. And MetaReg [Balaji *et al.*, 2018] proposed to explicitly meta-learn a regularization function for domain generalization. Our work is partly inspired by these works but we expand the existing works to a new method of meta functional learning to construct generalisable learning regularisation knowledge capable of guiding 'infant' functions to become 'mature' functions in a process of function update.

Few-Shot Learning. Existing methods for few-shot learning can be divided into several categories. 1) Metric-based methods learn a common feature space where categories can distinguish with each other based on a distance metric, and then infer labels for query data with a nearest neighbor classifier [Snell *et al.*, 2017] or a separate learnable similarity metric [Sung *et al.*, 2018]. 2) Gradient-based methods design the meta-learner as an optimiser that is learned to update model parameters. These approaches aim to learn good initialised parameters for a network so that the classifiers for novel classes can be learned with several gradient update steps on few labelled examples [Finn *et al.*, 2017; Ravi and Larochelle, 2016]. 3) Weight generation methods learn to generate classification weights for novel classes. A typical generation method directly predicts the classification weights from the activation statistics of their categories [Gidaris and Komodakis, 2018; Qi *et al.*, 2018]. Besides, some work try to generate better classification weights with denoising auto-encoders for weights reconstruction [Gidaris and Komodakis, 2019] or looking into the mutual information between generated weights and support/query data [Guo and Cheung, 2020]. Different from existing work to generate weights from the activations of a feature extractor, we aim to investigate the function learning update dynamics (a functional) which is not limited to backbone training strategies.

3 Methodology

Problem Definition. In the transfer learning scenario, we consider a large-scale labelled source/base image-label pair set $D_{src} = \{\mathbf{I}_j, y_j\}_{j=1}^{M}$, $y_j \in \mathcal{C}_{base}$, and a small labelled novel/target image set $D_{nov} = \{\mathbf{I}_j, y_j\}_{j=1}^{N}$, $y_j \in \mathcal{C}_{nov}$, from a base $\mathcal{C}_{base}$ and a novel category $\mathcal{C}_{nov}$ respectively. On D_{src}, we learn a representation function $\psi : \mathbf{I}_j \rightarrow x_j$, and then we learn a classifier $f_\phi : \psi(\mathbf{I}_j) \rightarrow y_j$, where ϕ is the parameter of f. A common practice in deep learning is optimising end-to-end ψ and f by formulating a multi-class classification problem over D_{src} with a cross-entropy loss. We employ this process here to compute a feature representation ψ.

Functional Learning. Our goal is to learn to fit a *functional regularisation,* $\mathcal{T} : f_\phi \rightarrow f_{\tilde{\phi}}$. Specially, the input $f_\phi(\psi(\mathbf{I}))$ is a classifier fitted by few labelled samples, and $\mathcal{T}(f_\phi)$ aims at approximating the corresponding function $f_{\tilde{\phi}}$ with regularisation knowledge learned from many labelled examples. We use ϕ and $\tilde{\phi}$ to denote the parameters learned by few and many labelled examples.

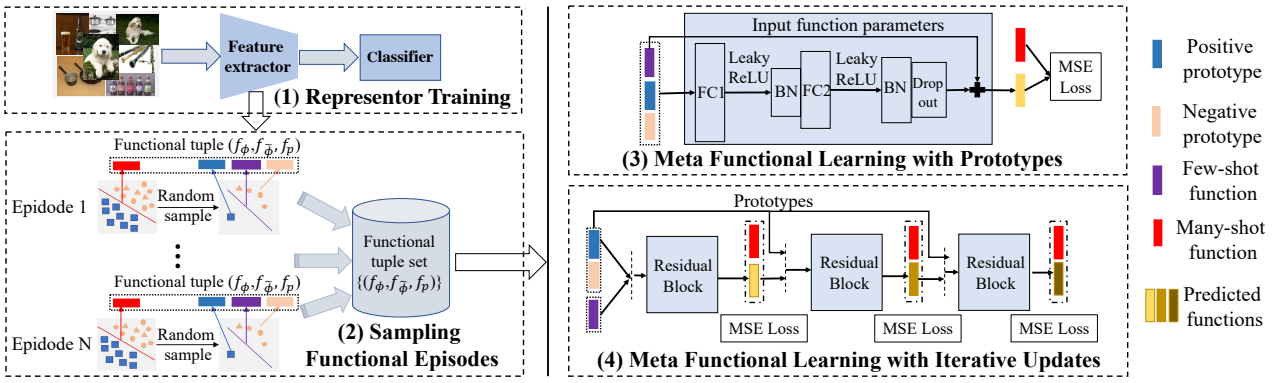

Figure 2: The overall model design for Meta Functional Learning (MFL). Plots (1-3) and (1, 2, 4) depict the process of MFL with prototypes and MFL with iterative updates, respectively. Each residual block in plot (4) represents a network architecture in the blue block of plot (3).

3.1 Meta Learning Task-Agnostic Functional

Rather than directly regressing $\mathcal{T}$ by a crafted functional set, we adopt a meta-learning strategy here. In principle, such a strategy helps cover a distribution of related tasks, sampled by episodes, and thus mimicking the predicting future functions from different domains. Our insight is that: *despite the data may be different between the source and target domains, the underlying patterns of change in fitting hypotheses to training data from few to many labelled examples, should be in principle, the same, or similar at least.* The functional $\mathcal{T}$ learned to represent such meta-knowledge of model convergence in one domain, could be generalisably applied to a novel domain. To that end, $\mathcal{T}$ should be learned in a task-agnostic manner.

Learning Task-Agnostic Knowledge Transfer. Our empirical study (in Fig. 1) shows the task-agnostic knowledge, i.e. the meta-knowledge of *functional regularisation*, extracted from a family of source tasks, could potentially be utilised to improve the generalisation of new tasks from that family. Particularly, to learn a regularisation knowledge transfer, we adopt the meta-learning strategy to learn the functional $\mathcal{T}$ over multiple learning episodes of the source tasks, sampled from base categories $\mathcal{C}_{base}$. Then the learned functional $\mathcal{T}$ is generalised and applied to tasks in target dataset D_{nov}. Our Meta Functional Learning (MFL) scheme is specified in Fig. 2: (1) The representator ψ is firstly trained on D_{src}; then (2) we sample the functional episodes to train $\mathcal{T}$; and finally, (3) we learn $\mathcal{T}$ by prototypes and functional episodes or (4) learn $\mathcal{T}_x$ with x iterative updates.

3.2 Sampling Functional Episodes

Rather than directly sample episodes from source data, we compute the functional episodes to support the learning of $\mathcal{T}$. Given the trained ψ, the goal of this step is to craft the paired functional set $\mathcal{F}_\mathcal{T} = \left\{\mathcal{F}_\mathcal{T}^{(b)}\right\}$ on D_{src} and the class $b \in \mathcal{C}_{base}$; and we denote $\mathcal{F}_\mathcal{T}^{(b)} = \left\{\left(f_\phi^{(b)}, f_{\tilde{\phi}}^{(b)}, f_p^{(b)}\right)\right\}$, where $f_\phi^{(b)}$ and $f_{\tilde{\phi}}^{(b)}$ are the classifiers of class b, trained by few and many examples, respectively; and $f_p^{(b)}$ represent the prototypes of the positive class b and other negative classes, computed by few labelled examples which are used for training $f_\phi^{(b)}$.

The sampled functional episodes include different classes in $\mathcal{C}_{base}$. This will help our meta functional learning algorithm to learn task-agnostic functional $\mathcal{T}$. Specifically, for class b ($b \in \mathcal{C}_{base}$), we compute functional tuple set $\mathcal{F}_\mathcal{T}^{(b)} = \left\{\left(f_\phi^{(b)}, f_{\tilde{\phi}}^{(b)}, f_p^{(b)}\right)\right\}$. For each tuple, $f_{\tilde{\phi}}^{(b)}$ is trained by the set of positive examples $\{\psi(\mathbf{I}_j), y_j = b\}$, *i.e.*, all images in class b, and negative examples $\{\psi(\mathbf{I}_j), y_j \neq b\}$ by randomly sampling from other classes. To obtain the set of tuples, this process is randomly repeated for M_l times. To compute $f_\phi^{(b)}$, we sample s samples and $k \times s$ samples from class b and other classes. For each $f_{\tilde{\phi}}^{(b)}$, we randomly sample samples M_f times and use different hyper-parameters to train the classifiers $f_\phi^{(b)}$ for increasing their diversity.

Remark. First, we adopt the $f_\phi^{(b)}$ by a vanilla binary classifier for class b, and the generalised multi-class scenario (one *vs.* all setting) is also extended in the experiments. We utilise the Logistic Regression (LR) classifiers here, and $f_\phi^{(b)}$ and $f_{\tilde{\phi}}^{(b)}$ are the corresponding vectors of LR parameters. Second, rather than directly learning $\mathcal{T} : f_\phi \rightarrow f_{\tilde{\phi}}$, our MFL learns an extended form with prototypes, i.e. $\mathcal{T} : (f_\phi, f_p) \rightarrow f_{\tilde{\phi}}$, where f_p is a vector by concatenating the positive and negative prototypes, which are computed by averaging the embeddings of samples from corresponding classes. Essentially, f_p is very important to our MFL, as it provides important category-related prototypes, to help $\mathcal{T}$ better learn the category agnostic knowledge in the meta training episodes.

3.3 Meta Functional Learning with Prototypes

Given the functional sets $\mathcal{F}_\mathcal{T}$, we design a meta functional learning mechanism to learn the functional regularisation $\mathcal{T}$. For any given class b, the objective of our MFL is to approximate the ground-truth output $f_{\tilde{\phi}}^{(b)} = \mathcal{T}\left(f_\phi^{(b)}, f_p^{(b)}\right)$. We introduce Mean Square Error (MSE) to measure the difference of parameter vectors $\left(f_\phi, f_{\tilde{\phi}}, f_p\right)$ as,

$$l_\tau = \mathbf{E}_{\left(f_\phi, f_{\tilde{\phi}}, f_p\right) \sim \mathcal{F}_\mathcal{T}} \left\| f_{\tilde{\phi}} - \mathcal{T}\left(f_\phi, f_p\right) \right\|^2 \qquad (1)$$

Algorithm 1 Meta Functional Learning (MFL).

Require: Embeddings $\Psi_{src} = \{\psi(\mathbf{I}_j), y_j \in \mathcal{C}_{base}\}$ of D_{src}; Classifier f_c; Sampling time M_l, M_f; Hyper-parameter set H; Shot number s, $s \times k$; Train epochs T;
Ensure: Functional set $\mathcal{F}_{\mathcal{T}}$; Functional regularisation $\mathcal{T}$;

1: **// Sampling Functional Episodes**
2: $\mathcal{F}_{\mathcal{T}} = \Phi$; $\mathcal{F}_{\mathcal{T}}^{(b)} = \Phi, b \in \mathcal{C}_{base}$;
3: **for** all $b \in \mathcal{C}_{base}$ **do**
4: Sample episode $\mathcal{E}_l = \{\psi(\mathbf{I}_j^i), y_j = b\}_{i=1}^{N_b} \bigcup \{\psi(\mathbf{I}_j^i), y_j \neq b\}_{i=1}^{2\times N_b}$ from Ψ_{src} and train $f_{\tilde{\phi}}^b$ on $\mathcal{E}_l$;
5: Randomly sample sub-episode $\mathcal{E}_f$ including $s(s \times k)$ $\psi(\mathcal{I}_j)$ with $y_j = (\neq) b$ from $\mathcal{E}_l$ and train $f_{\phi}^{(b)}$ on $\mathcal{E}_f$;
6: Compute $f_p^{(b)}$ including the prototypes of $\psi(\mathcal{I}_j)$ with $y_j = b$ and $y_j \neq b$ in $\mathcal{E}_f$;
7: $\mathcal{F}_{\mathcal{T}}^{(b)} = \mathcal{F}_{\mathcal{T}}^{(b)} \bigcup (f_{\phi}^{(b)}, f_{\tilde{\phi}}^{(b)}, f_p^{(b)})$;
8: Repeat line 6-7 using f_c with h in H;
9: Repeat line 5-8 for M_f times;
10: Repeat line 4-9 for M_l times;
11: $\mathcal{F}_{\mathcal{T}} = \mathcal{F}_{\mathcal{T}} \bigcup \mathcal{F}_{\mathcal{T}}^{(b)}$
12: **end for**
13: **// Meta Function learning**
14: **while** $t < T$ **do**
15: Randomly split mini-batches with size n from $\mathcal{F}_{\mathcal{T}}$;
16: **for** each mini-batch **do**
17: Predict functions $\mathcal{T}(f_\phi, f_p)$ with $\mathcal{T}$;
18: Compute the loss in Eq. 1;
19: Update the parameters of $\mathcal{T}$;
20: **end for**
21: **end while**

Model Implementation. The functional $\mathcal{T}$ is implemented as a deep network, with the model architecture in Fig. 2(3). It consists of a residual block, where the LeakyReLu activation function is used to learn the nonlinear mapping from fully connection layers. We employ BatchNorm and dropout to improve the generalisation of $\mathcal{T}$. The skip connection is used to keep the scale of classifiers' parameters and avoid the degradation of learning. The pseudo-codes of sampling functional episodes and MFL are shown in Alg. 1.

MFL with Iterative Updates. The model in Fig. 2(3) can be updated by multiple blocks. Specifically, as illustrated in Fig. 2(4), we enable MFL by a sequence of blocks, i.e. MFL with Iterative Updates (MFL-IU). And MFL-IUx represents the output of xth basic block, i.e. $\mathcal{T}_x\left(\mathcal{T}_{x-1}\cdots\left(\mathcal{T}_1\left(f_\phi, f_p\right)\right)\right)$. A simple version is MFL-IU1 by only using one block for MFL. The training process of MFL-IU is illustrated in Alg. 2.

Algorithm 2 MFL with Iterative Updates (MFL-IU).

Require: Functional set $\mathcal{F}_{\mathcal{T}}$; Iterations X; Train epochs T;
Ensure: Functional regularisation $\mathcal{T} = \{\mathcal{T}_1, ..., \mathcal{T}_X\}$;

while $t < T$ **do**
  Randomly split mini-batches with size n from $\mathcal{F}_{\mathcal{T}}$;
  for each mini-batch **do**
    for $x < X$ **do**
      Predict functions $\mathcal{T}_x(f_\phi, f_p)$ with $\mathcal{T}_x$;
      Compute the loss in Eq. 1;
      Update the parameters of $\mathcal{T}_x$;
    end for
  end for
end while

4 Experiments

To evaluate the effectiveness of MFL, we tested MFL on two data-scarce learning problems: N-way K-shot classification, i.e. a task aiming to discriminate between N classes with K labelled samples of each class, by (1) standard FSL and (2) Cross-Domain FSL (CD-FSL). In particular, we adopted a binary classifier as a vanilla classifier and generalised it to multi-way classification scenari with one *vs.* all manner. We first evaluated MFL on basic 2-way FSL tasks and then investigated whether the learning pattern of MFL can be generalised to multi-way FSL tasks. Furthermore, the experiments on CD-FSL were carried out for learning tasks with different shot numbers to investigate the model generalisation capacity to multi-shot FSL tasks.

Datasets. We employed three FSL datasets: 1) *miniImageNet* is a subset of the ILSVRC-12 dataset and contains 100 classes with 600 images per class. We followed the split in [Ravi and Larochelle, 2016] and used 64, 16 and 20 classes as base, validation and novel set. 2) *CIFAR-FS* is a dataset with lower-resolution images, and it contains 100 classes with 600 instances in each class. Following the split in [Bertinetto *et al.*, 2019], we used 64 classes to construct the base set, 16 and 20 for validation and novel set. 3) *CUB* is a fine-grained dataset which consists of 200 bird categories with 11788 images in total. We used 100, 50 and 50 classes for base, validation and novel set with the previous setting in [Hilliard *et al.*, 2018], and we conducted all experiments with the cropped images provided in [Triantafillou *et al.*, 2017].

Implementation. We used Conv4 as the backbone for learning a feature representation. The architecture of this Conv4 network is provided by [Snell *et al.*, 2017] and it contains four convolutional blocks. Each block comprises a 64-filter 3×3 convolution, batch normalization layer, a ReLU nonlinearity and a 2×2 max-pooling layer. For training the network, we randomly split the images from base classes into (90%, 10%) partitions as (train, validation) sets. We trained the backbone over 120 epochs. The batch size and learning rate are set as 64 and 0.01. For training MFL/MFL-IU, we employed BatchNorm (0.1), dropout (0.9) and LeakyReLU (0.01), and the parameters for the first and second fully connected layers are 6000 and 1601 respectively. Moreover, we trained MFL/MFL-IU over 50 epochs with batch size (256) and learning rate (0.01). We adopted the Logistic Regression (LR) function as the base binary classifier and the parameters for computing functional set are $M_l = 5, M_s = 100$, $k = \{1, 2, 3, 4\}$ and $H = 1e\{-2, -1, 0, 1, 2\}$. Specifically, we set $s = \{1, 2, 3, 4, 5\}$ to construct functional tuple sets for s-shot learning scenarios in FSL. In all experiments, we selected the best model by evaluating them on a validation set and evaluated all methods with 600 episodes randomly selected from the novel classes in the corresponding dataset.

Dataset	Methods	2-way	3-way	4-way	5-way	10-way	20-way
miniImageNet	Baseline†	70.09±1.13	55.74±0.99	46.33±0.79	40.41±0.68	26.50±0.38	16.09±0.21
	ProtoNet†	73.76±1.34	59.34±1.14	51.24±0.95	45.22±0.81	29.04±0.44	18.09±0.23
	MAML†	73.56±1.38	62.21±1.16	52.44±0.94	48.29±0.83	31.41±0.47	-
	Vanilla LR	72.86±1.13	59.51±0.93	51.05±0.83	46.18±0.77	31.04±0.44	21.09±0.24
	MetaModelNet‡	76.34±1.36	62.54±1.14	53.51±0.97	47.99±0.85	31.02±0.46	19.23±0.24
	MFL (Ours)	76.50±1.17	63.12±0.99	54.53±0.86	49.01±0.80	33.26±0.45	22.40±0.26
	MFL-IU3 (Ours)	**78.25±1.24**	**65.77±1.00**	**56.60±0.89**	**51.17±0.83**	**34.74±0.46**	**23.45±0.26**
CIFAR-FS	Baseline†	72.66±1.14	59.44±1.06	50.77±0.85	46.16±0.77	32.46±0.46	22.04±0.26
	ProtoNet†	73.36±1.13	60.45±1.20	51.87±1.01	47.04±0.91	31.41±0.51	20.48±0.25
	MAML†	75.82±1.35	63.06±1.23	56.82±1.03	50.15±0.94	39.52±0.60	-
	Vanilla LR	76.53±1.16	64.12±1.02	56.62±0.92	51.48±0.82	38.67±0.49	28.27±0.28
	MetaModelNet‡	79.37±1.25	67.96±1.23	60.11±1.11	55.26±1.02	39.48±0.61	27.09±0.31
	MFL (Ours)	80.50±1.15	69.57±1.05	61.86±0.98	57.05±0.88	43.15±0.54	31.20±0.29
	MFL-IU3 (Ours)	**82.17±1.18**	**72.30±1.11**	**64.72±1.05**	**60.11±0.94**	**45.46±0.60**	**32.95±0.30**

Table 1: Few-Shot Learning Evaluation: Comparison to Vanilla LR and prior work on miniImageNet and CIFAR-FS with Conv4 backbone. Mean accuracies (%) with 95% confidence intervals results are reported on N-way 1-shot FSL. $(\cdot)^\dagger$ represent the experimental results with the released codes and $(\cdot)^\ddagger$ are our re-implemented results with the corresponding paper. **Bold:** the best scores.

4.1 Meta Functional Learning

Competitors. We compared our methods against existing models for N-way 1-shot FSL tasks from three perspectives: 1) Comparison with the base classifier: We used Logistic Regression (LR) as a typical classifier. As in Tab.1, the Vanilla LR represents a naive LR classifier trained on labelled data, while MFL and MFL-IU3 are the predicted functions with our MFL and MFL-IU3, respectively. 2) Comparison with typical FSL methods: Baseline [Chen *et al.*, 2019] ProtoNet [Snell *et al.*, 2017], and MAML [Finn *et al.*, 2017]; 3) Comparison with a model transformation method: MetaModelNet [Wang *et al.*, 2017]. Since no official result is provided on these comparison methods in N-way classification FSL, we re-ran the released code in [Chen *et al.*, 2019] for evaluating FSL methods and evaluated MetaModelNet with our re-implemented model following [Wang *et al.*, 2017].

Results and Analysis. Table 1 shows the comparative results on miniImageNet and CIFAR-FS. We can see that: (1) Our methods can effectively transfer the regularisation knowledge to benefit the naive functions, i.e. Vanilla LR, yielding more robust and accurate functions with significant performance improvement on 2/3/4/5/10/20-way 1-shot FSL; (2) Our methods significantly outperform three typical FSL methods, achieving the potentially smooth and discriminative hypotheses on a fixed ; (3) MetaModelNet can improve the performance of the Vanilla LR in low-way (1-5 way) FSL tasks, while the improvement in higher way (10/20 way) FSL tasks is limited. In contrast, our methods performed well in all N-way 1-shot FSL tasks. This verifies that our methods are more robust and generalisable to multi-way FSL tasks. Moreover, MFL and MFL-IU3 are both effective in improving the performance of the Vanilla LR. MFL-IU3 performed better than MFL due to the benefit from the progressive increasing functional regularisation knowledge provided by the iterative update strategy.

4.2 Learning Cross-Domain

We employed MFL and MFL-IU3 on a more challenging task, CD-FSL. We followed the miniImageNet → CUB setting in [Chen *et al.*, 2019], where D_{src} and D_{nov} are the images from the base classes of miniImageNet and the novel classes of CUB, respectively. We adopted the same competitors in section 4.1 and carried out experiments on CD-FSL by using 5-way K-shot settings to investigate model effectiveness on different shot learning scenarios. Table 2 shows the results with the following observations: (1) By directly using the learned representation trained on miniImageNet, the three existing FSL methods give inferior performance on CD-FSL. (2) MetaModelNet, the model transformation method, improved the Vanilla LR on FSL but failed on CD-FSL, resulting in a poorer transformed classifier than Vanilla LR. (3) Our methods are able to improve the Vanilla LR by transferring the regularisation knowledge in model learning across domains, yielding a more accurate classifier with 1%-3% increase of classification accuracy on 5-way K-shot CD-FSL.

Dataset	**miniImageNet → CUB**				
#shot	1	2	3	4	5
Baseline	36.46	45.21	50.94	55.76	58.52
ProtoNet	41.06	50.88	52.73	60.01	60.38
MAML	42.22	49.33	55.07	57.72	58.08
Vanilla LR	42.42	51.92	58.42	62.74	66.21
MetaModelNet	36.54	42.40	45.95	50.81	52.52
MFL (Ours)	44.04	53.10	59.15	63.54	66.94
MFL-IU3 (Ours)	**45.17**	**53.69**	**60.34**	**64.50**	**67.71**

Table 2: Cross-Domain Few-Shot Learning Evaluation: Mean accuracies (%) of our methods and the competitors with Conv4 backbone on 5-way K-shot tasks under the cross-domain scenario.

4.3 Visualisation

To validate our hypothesis, i.e. the regularisation knowledge transfer with MFL, we adopted T-SNE [Maaten and Hinton, 2008] to visualise the classification results of Vanilla LR and MFL-IU3 on 2-way 1-shot tasks from the novel classes of miniImageNet. Specifically, we showed three typical data distributions, i.e. continuity, cluster and manifold, for com-

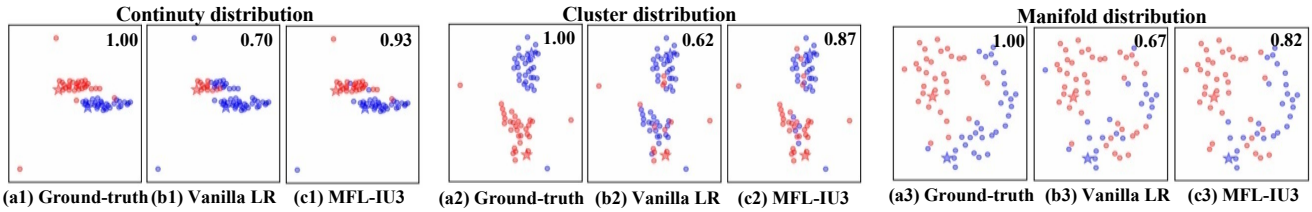

Figure 3: The T-SNE visualisation of 2-way 1-shot FSL tasks from miniImageNet. Plots (a1-3) depict the data distributions with ground-truth labels, while plots (b1-3) and (c1-3) are the classification results of Vanilla LR and MFL-IU3, respectively. The red/blue stars and round points represent the class(0/1) train and test data, while numbers in plots are the classification accuracies of corresponding methods.

Backbone	Conv4		ResNet-12	
#shot	1	5	1	5
Vanilla SVM	46.0	62.36	58.25	74.26
MFL (Ours)	48.98	64.32	58.95	74.81
MFL-IU3 (Ours)	**50.77**	**65.42**	**59.45**	**75.19**

Table 3: Mean accuracies (%) of Vanilla SVM and SVM with MFL and MFL-IU3 on 5-way 1/5shot tasks from miniImageNet.

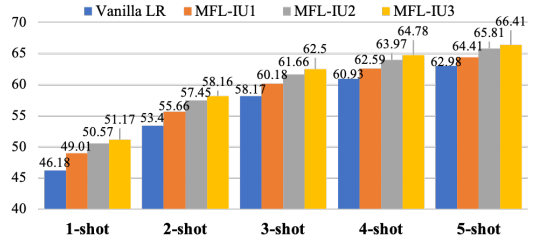

Figure 4: Mean accuracies (%) of Vanilla LR and MFL-IUx on 5-way K-shot tasks from miniImageNet with Conv4 backbone.

prehensively describing the regularisation behaviors with the learned functional regularisation knowledge. Figure 3 shows: (1) In a specific feature space, the data distributions fit the characters of continuity, cluster or manifold (Fig. 3(a1-3)); (2) The few-shot classifiers easily overfit to the labelled data, resulting in hypotheses lacking of regularisation and inferior classification results (Fig. 3(b1-3)); (3) Our MFL-IU3 can remit this limitation via imposing the functional regularisation knowledge into classifiers, achieving more reasonable hypotheses with superior classification results (Fig. 3(c1-3)).

4.4 Ablation Study

Generalisation on Different Classifiers and Backbones

We conducted experiments to investigate the generalisation ability of MFL on different base classifier and backbones. Specifically, we used linear Support Vector Machine (SVM) as a base classifier, and two backbone networks Conv4 and ResNet12 [Wang *et al.*, 2020] for learning a representation. As in Tab. 3, our methods perform well on different classifiers, i.e. LR and SVM, verifying the generalisation ability of MFL on different classifiers. Besides, our methods show well generalisation ability on different backbones. Noticeable, the improvement on Conv4 is larger than that on ResNet12, we conjecture this may attribute to the shallow architecture of Conv4, yielding less discriminative representation in which the learned vanilla classifiers are easily stuck in the overfitting problem and our MFL can effectively extricate them from this dilemma via the knowledge of functional regularisation.

Effects of Shot Number and Iterative Steps

To demonstrate the effectiveness of the iterative update strategy, we conducted experiments using MFL-IUx with x iterative steps. Besides, we further investigated the generalisation ability of MFL on FSL with different shot number. Figure 4 shows the results of MFL-IUx ($x = 1, 2, 3$) on 5-ways K-shot ($K = 1, 2, 3, 4, 5$) FSL and we can see that: (1) The MFL-IUx with different iterative steps can both boost the classifiers' performance on 1/2/3/4/5-shot FSL; (2) With the increase of iterations, the predicted functions become more accurate, demonstrating the effectiveness of iterative updates; (3) With the number of shot increasing, the improvement on Vanilla LR with MFL-IUx deceases. This suggests that the hypotheses can gradually learn regularisation knowledge with the help of available labelled data, yielding more robust hypotheses where the boosting space with regularisation knowledge is narrow, thus the functional regularisation knowledge in MFL-IUx brings less improvement.

5 Conclusions

In this work, we explored the idea of knowledge transfer by learning a meta functional of regularisation in the model learning function spaces between a richly labelled domain and a scarcely labelled domain. We demonstrate that classifiers with less training data can gradually learn the functional regularisation knowledge from a concurrent learning process on more labelled data. Based on this observation, we consider that this functional regularisation knowledge can be transferred across different domains for model learning tasks when training data is scarce. We formulated the MFL with iterative updates. Extensive experiments on miniImageNet, CIFAR-FS and CUB, show that the transfer of model learning regularisation knowledge is effective in learning more accurate hypotheses (classifiers) given scarcely labelled data.

Acknowledgments

This work was supported by Vision Semantics Limited, the Alan Turing Institute Turing Fellowship, the China Scholarship Council, Innovate UK Industrial Challenge Project on Developing and Commercialising Intelligent Video Analytics Solutions for Public Safety (98111-571149).

References

[Balaji *et al.*, 2018] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in Neural Information Processing Systems*, 31:998–1008, 2018. 2

[Bertinetto *et al.*, 2019] Luca Bertinetto, Joao F Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019. 4

[Brown *et al.*, 2020] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1

[Chapelle *et al.*, 2009] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009. 1

[Chen *et al.*, 2019] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019. 1, 4.1, 4.2

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017. 2, 4.1

[Gidaris and Komodakis, 2018] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 2

[Gidaris and Komodakis, 2019] Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21–30, 2019. 2

[Guo and Cheung, 2020] Yiluan Guo and Ngai-Man Cheung. Attentive weights generation for few shot learning via information maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13499–13508, 2020. 2

[Hilliard *et al.*, 2018] Nathan Hilliard, Lawrence Phillips, Scott Howland, Artëm Yankov, Courtney D Corley, and Nathan O Hodas. Few-shot learning with metric-agnostic conditional embeddings. *arXiv preprint arXiv:1802.04376*, 2018. 4

[Johnson and Zhang, 2019] Rie Johnson and Tong Zhang. A framework of composite functional gradient methods for generative adversarial models. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 2

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 4.3

[Qi *et al.*, 2018] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5822–5830, 2018. 2

[Ravi and Larochelle, 2016] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2016. 1, 2, 4

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017. 2, 4, 4.1

[Sung *et al.*, 2018] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 2

[Triantafillou *et al.*, 2017] Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. In *Advances in Neural Information Processing Systems*, pages 2255–2265, 2017. 4

[Tseng *et al.*, 2020] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. In *International Conference on Learning Representations*, 2020. 1

[Wang and Hebert, 2016] Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *European Conference on Computer Vision*, pages 616–634. Springer, 2016. 2

[Wang *et al.*, 2017] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pages 7029–7039, 2017. 2, 4.1

[Wang *et al.*, 2020] Yikai Wang, Chengming Xu, Chen Liu, Li Zhang, and Yanwei Fu. Instance credibility inference for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12836–12845, 2020. 4.4

[Ye *et al.*, 2020] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020. 1