Universal Representation Learning from Multiple Domains for Few-shot Classification

Wei-Hong Li, Xialei Liu*, and Hakan Bilen
VICO Group, University of Edinburgh, United Kingdom
groups.inf.ed.ac.uk/vico/research/URL

Abstract

In this paper, we look at the problem of few-shot image classification that aims to learn a classifier for previously unseen classes and domains from few labeled samples. Recent methods use various adaptation strategies for aligning their visual representations to new domains or select the relevant ones from multiple domain-specific feature extractors. In this work, we present URL, which learns a single set of universal visual representations by distilling knowledge of multiple domain-specific networks after co-aligning their features with the help of adapters and centered kernel alignment. We show that the universal representations can be further refined for previously unseen domains by an efficient adaptation step in a similar spirit to distance learning methods. We rigorously evaluate our model in the recent Meta-Dataset benchmark and demonstrate that it significantly outperforms the previous methods while being more efficient.

1. Introduction

As deep neural networks progress to dramatically improve results in most of standard computer vision tasks, there is a growing community interest for more ambitious goals. One of them is to improve the data efficiency of the standard supervised methods that rely on large amount of expensive and time-consuming hand-labeled data. Just like the human intelligence is capable of learning concepts from few labeled samples, *few-shot learning* [24, 33] aims at adapting a classifier to accommodate new classes not seen in training, given a few labeled samples from these classes.

Earlier works in few-shot learning focus on evaluating their methods in homogeneous learning tasks, *e.g*. Omniglot [25], miniImageNet [53], tieredImageNet [43], where both the meta-train and meta-test examples are sampled from a single data distribution (or dataset). Recently, the interest of the community has shifted to a more realistic and challenging experimental setting, where the goal is to learn few-shot models that can generalize not only within a single data distribution but also to previously unseen data distributions. To this end, Triantafillou *et al*. [52] propose a new heterogeneous benchmark, Meta-Dataset that consists of ten datasets from different domains for meta-training and meta-test. While, initially two domains were kept as unseen domains, later three more unseen domains are included to meta-test the generalization ability of learned models.

While the few-shot methods [14, 48, 49, 53], which were proposed before Meta-Dataset was available, can be directly applied to this new benchmark with minor modifications, they fail to cope with domain gap between train and test datasets and thus obtain subpar performance on Meta-Dataset. Recently several few-shot learning methods are

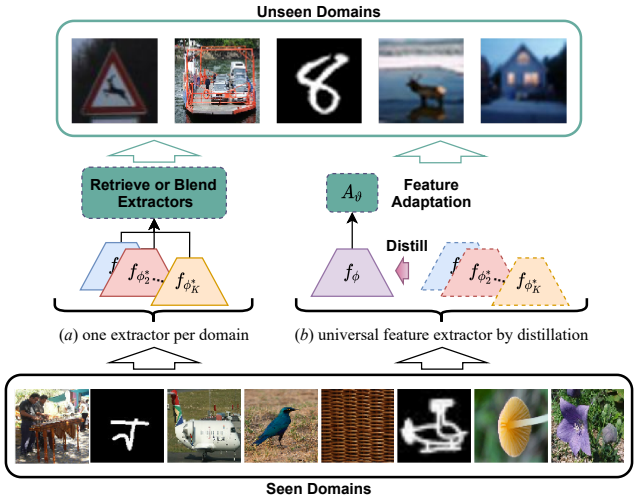

Figure 1. **URL – Universal Representation Learning**. Unlike the previous methods [13, 29] (illustrated in (a)) that learn K feature extractors $\{f_{\phi_\tau^*}\}_\tau^K$, one for each domain, and retrieve or combine their features for the target task during meta-test stage, our method (illustrated in (b)) learns a single universal feature extractor f_ϕ that is distilled from from multiple feature extractors $\{f_{\phi_\tau^*}\}_\tau^K$. In meta-test stage, we use a linear transformation A_ϑ to further refine the universal representations to unseen domains.

*Xialei Liu is the corresponding author.

proposed to address this challenge, which can be coarsely grouped into two categories, adaptation [2, 44] and feature selection based methods [13, 29]. CNAPS [44] consists of an adaptation network that modulates the parameters of both a feature extractor and classifier for new categories by encoding the data distribution of few training samples. Simple CNAPS [2] extends CNAPS by replacing its parametric classifier with a non-parametric classifier based on Mahalanobis distance and shows that adapting the classifier from few samples is not necessary for good performance. SUR [13] and URT [29] further show that adaptation for the feature extractor can also be replaced by a feature selection mechanism. In particular, both [13, 29] learn a separate deep network for each training dataset in an offline stage, employ them to extract multiple features for each image, and then select the optimal set of features either based on a similarity measure [13] or on an attention mechanism [29]. Despite their good performance, SUR and URT are computationally expensive and require multiple forward passes through multiple networks during inference time.

In this work, we propose an efficient and high performance few-shot method, called *URL* based on *U*niversal *R*epresentation *L*earning. Like [13, 29], our method builds on multi-domain representations that are learned in an offline stage. However, we learn a single set of universal representations (a single feature extractor) over multiple domains which has a fixed computational cost regardless of the number of domains at inference unlike them. Similar to the adaptation based techniques [2, 44], our method further employs a simple adaptation strategy to learn the domain specific representations from few samples (see Fig. 1).

In particular, we propose to *distill* knowledge from multiple domains to a single model, which can efficiently leverage useful information from multiple diverse domains. Learning multi-domain representations is a challenging task and requires to leverage commonalities in the domains while minimizing interference (negative transfer [8, 41, 56]) between them. To mitigate this, we align the intermediate representations of our multi-domain network with the ones of the domain-specific networks after carefully aligning each space by using small task-specific adapters and Centered Kernel Alignment (CKA) [22]. Finally, inspired from the use of Mahalanobis distance in [2], we adapt the learned multi-domain features into the new task by mapping them into a task-specific space. However, unlike [2], we *learn* the parameters of this mapping via adaptation in a discriminative way. We rigorously evaluate our method in Meta-Dataset benchmark and show that our method outperforms the state-of-the-art methods significantly in both previously seen and unseen domain generalization.

2. Related Work

Meta-learning based few-shot classification. Meta-learning approaches for few-shot learning that allow for end-to-end training of few-shot classifiers be broadly divided into two groups, metric-based and optimization-based approaches. The key idea in the former is to map raw images to vector representations and use nearest neighbor classifiers with different distance functions by learning discriminative feature spaces with Siamese networks [21], producing a weighted nearest neighbor classifier [53], representing each class with the average of the samples in the support set [48]. The latter focuses on learning models that can quickly adapt to new tasks from few samples in support. The successful methods include MAML [14] that poses learning to learn problem in a bi-level optimization where the weights of the network are modeled as a function of the initial network weights, Reptile [35] that alleviates the expensive second order derivative computation in MAML by a first order approximation, MAML++ [1] that introduces multiple speed and stability improvements over MAML.

Transfer learning based few-shot classification. There are also simple yet effective methods [6, 7, 11] that first learn a neural network on all the available training data and transfer it to few-shot tasks in test time. Baseline++ [6] only updates a parametric classifier with cosine distance, while Meta-Baseline [7] fine-tunes entire network with a nearest-centroid cosine similarity and a scale parameter. Dhillon *et al*. [11] explore fine-tuning in a transductive setting, where the query set is assumed to be available at the same time.

Cross-domain few-shot classification. Recent few-shot techniques [5, 13, 29, 44] focus on few-shot learning that generalizes to unseen domains at test time in the recently proposed Meta-Dataset [52]. CNAPS [44] adapts the parameters of feature encoder and classifier by conditioning them on current input task via FiLM layers [39] which is further extended in Simple CNAPS [2] adopts a non-parametric classifier using a simple class-covariance-based distance metric, namely the Mahalanobis distance. In contrast SUR [13] stores the domain-specific knowledge by learning an independent feature extractor for each domain, and automatically selects the most relevant representations for a new task by linearly combining features from domain-specific features. URT [29] instead meta-learns the feature selection mechanism for new tasks by using Transformer layers. Like SUR and URT, our method uses multi-domain features but in a more efficient way, by learning a single network over multiple domains. Our method requires significantly less network capacity and compute load than theirs. In addition, similar to Simple CNAPS [2], we map our features to a task-specific space before applying the nearest neighbor classifier but we learn the parameters of this mapping from each support set.

Knowledge distillation. Our work is related to knowledge distillation (KD) methods [17, 27, 30, 40, 45, 50] that dis-

tills the knowledge of an ensemble of large teacher models to a small student neural network at the classifier [17] and intermediate layers [45]. Born-Again Neural Networks [15] uses KD proposes to consecutively distill knowledge from an identical teacher network to a student network, which is further applied to few-shot learning in [51] and multi-task learning in [10]. Most similar to our work, Li and Bilen [27] apply knowledge distillation to align features of a student multi-task network to multiple single-task learning networks by introducing task-specific adapters. While we use task-specific adapters to align the features across multiple networks like [27], we apply the alignment to a more challenging setting of multi-domain learning where there are substantial gap between different domains unlike their method that is shown to work in multi-task learning where multiple tasks are sampled from a single data distribution. To this end, we incorporate a more effective feature matching loss inspired from [22] to align features in presence of large domain gap.

Universal representation. A representation that works equally well in multiple domain, termed *universal representation*, is introduced in [3]. To learn a universal representation in multiple domains, SUR [13] and URT [29] propose to learn an independent model for each domain and learn to retrieve or blend appropriate models for a new task in few-shot classification. Alternatively, [3, 41, 42] propose to learn a single network to perform image classification on very different domains by sharing a large majority of parameters across domains and encoding domain-specific information via normalization layers [3], light-weight residual adapters [41, 42], Feature-wise Linear Modulate (FiLM) [39]. Our method is inspired from these methods, thus we learn universal representations without any domain-specific weights and use them in few-shot learning.

3. Method

In this section, we describe the problem setting, introduce our method in two parts, multi-domain feature learning and feature adaptation.

3.1. Few-shot task formulation

Few-shot classification aims at learning to classify samples from a small training set with only few samples for each class. The task contains two sets of images: a support set $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|\mathcal{S}|}$ that contains $|\mathcal{S}|$ image and label pairs respectively that define the classification task and a query set $\mathcal{Q} = \{(\boldsymbol{x}_j)\}_{i=1}^{|\mathcal{Q}|}$ that contains $|\mathcal{Q}|$ samples to be classified. In words, we would like to learn a classifier on the support set that can accurately predict the labels of the query set.

As in [13, 29], we solve this problem in two steps: i) a meta-training step where a learning algorithm receives a large dataset $\mathcal{D}_b$ and outputs a general feature extractor f, ii) a meta-test step where the target tasks $(\mathcal{S}, \mathcal{Q})$ are sampled from another large dataset $\mathcal{D}_t$ by taking the subsets of the dataset to build $\mathcal{S}$ and $\mathcal{Q}$. Note that $\mathcal{D}_b$ and $\mathcal{D}_t$ contain mutually exclusive classes.

3.2. Learning universal representations

Our focus is to learn few-shot image classification that generalizes not only within previously seen visual domains but also to unseen ones. As it is challenging to obtain the domain-specific knowledge from only few samples in a previously unseen domain, inspired by [3, 41], we hypothesize that using multi-domain (or universal) representations is the key to the success of cross-domain generalization. To this end, we propose learning a multi-domain network that works well for all the domain-specific tasks simultaneously and use this network as a feature extractor for the target tasks.

Let assume that $\mathcal{D}_b$ consists of K subdatasets, each sampled from a different domain. One potential solution is train a multi-domain network by jointly optimizing its parameters over the images from all K domains (subdatasets):

$$\min_{\phi,\psi_\tau} \sum_{\tau=1}^{K} \frac{1}{|\mathcal{D}_\tau|} \sum_{\boldsymbol{x},y \in \mathcal{D}_\tau} \ell(h_{\psi_\tau} \circ f_\phi(\boldsymbol{x}), y), \tag{1}$$

where ℓ is cross-entropy loss, f is a multi-domain feature extractor that takes an image as input and outputs a d dimensional feature and is parameterized by a single set of parameters ϕ which is shared across K domains. h is a domain-specific classifier that takes in $f_\phi(\boldsymbol{x})$ and outputs a probability vector over the target categories and it is parameterized by ψ_τ. While minimizing Eq. (1) results in a multi-domain feature extractor f, several previous works report that this optimization is problematic due to the interference between the different tasks [8, 56], varying dataset sizes and difficulty [20, 27] and often leads to subpar results compared to individual single-domain networks.

Motivated by these challenges, we propose a two stage procedure to learn multi-domain representations, inspired by the previous distillation methods [17, 27]. To this end, we first train domain-specific deep networks where each consists of a specific feature extractor $f_{\phi_\tau^*}$ and classifier $h_{\psi_\tau^*}$ with parameters ϕ_τ^* and ψ_τ^* respectively, similarly to [13, 29]. However, instead of using K domain-specific feature extractors and selecting the most relevant ones like them, we propose to learn a single multi-domain network that performs well in K domains by distilling the knowledge of K pretrained feature extractors. This has two key advantages over [13, 29]. First using a single feature extractor, which has the same capacity with each domain-specific one, is significantly more efficient in terms of run-time and number of parameters in the meta-test stage. Second learning to find the most relevant features for a given support and query set in [29] is not trivial and may also suffer from overfitting to the small number of datasets in the training set, while the

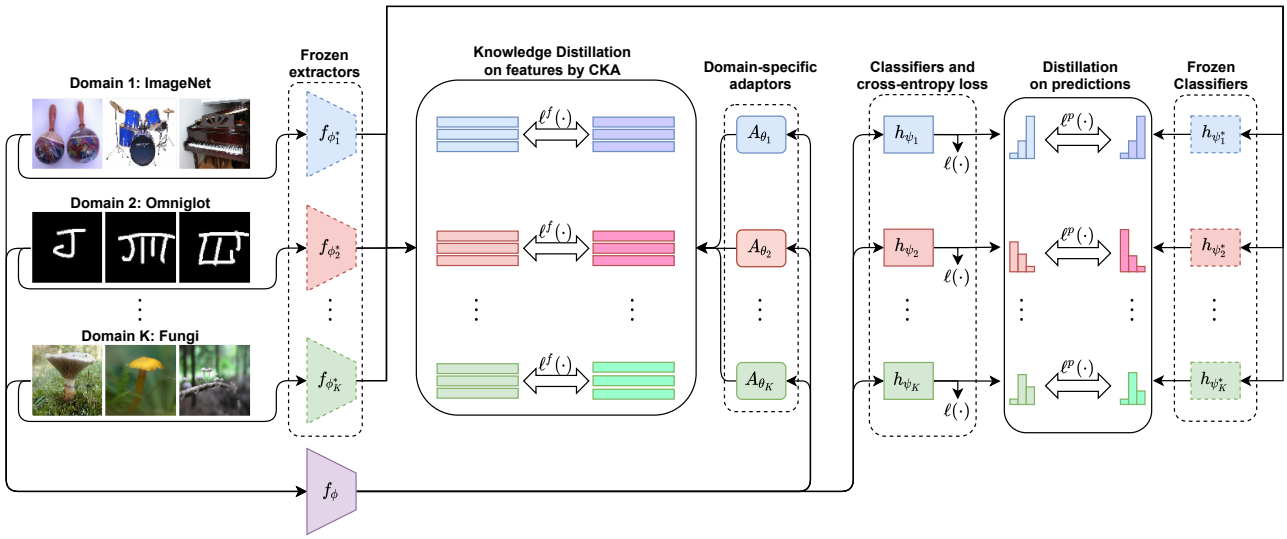

Figure 2. **Training pipeline for universal representation learning**. Given training images from K different domains, we first train K domain-specific networks $f_{\phi_1^*}, \ldots, f_{\phi_K^*}$ and their classifiers $h_{\psi_1^*}, \ldots, h_{\psi_K^*}$, freeze their weights and distill their knowledge to our multi-domain network by matching their features and predictions through two loss functions ℓ^f and ℓ^p respectively. As matching multiple features is challenging, we co-align all the features by using light-weight adaptors $A_{\theta_1}, A_{\theta_2}, \ldots, A_{\theta_K}$ and centered kernel alignment.

multi-domain representations, by definition, automatically contain the required information from the relevant domains.

In the second stage, we freeze the pretrained weights of the domain-specific feature extractors $f_{\phi_\tau^*}$ and transfer their knowledge into the multi-domain model at train time. Knowledge distillation can be performed at the prediction [17] and feature level [27, 45] by minimizing the distance between (i) the predictions of the multi-domain and corresponding single-domain network, and also between (ii) the multi-domain and single-domain features respectively for given training samples. While Kullback-Leibler (KL) divergence is the standard choice for the predictions in [17], matching the multi-domain features to multiple single-domain ones simultaneously is an ill-posed problem, as the features from different domain-specific extractors for a given image $\boldsymbol{x}$ are not necessarily aligned and can vary significantly. To this end, as in [27], we propose to map each domain specific feature into a common space by using adaptors $A_{\theta_\tau} \in \mathbb{R}^{d \times d}$ with parameters θ_τ and jointly train them along with the parameters of the multi-domain network:

$$\min_{\phi, \psi_\tau, \theta_\tau} \sum_{\tau=1}^{K} \frac{1}{|\mathcal{D}_\tau|} \sum_{\boldsymbol{x}, y \in \mathcal{D}_\tau} \Big(\ell(h_{\psi_\tau} \circ f_\phi(\boldsymbol{x}), y) + \lambda_\tau^p \ell^p(h_{\psi_\tau} \circ f_\phi(\boldsymbol{x}), h_{\psi_\tau^*} \circ f_{\phi_\tau^*}(\boldsymbol{x})) + \lambda_\tau^f \ell^f(A_{\theta_\tau} \circ f_\phi(\boldsymbol{x}), f_{\phi_\tau^*}(\boldsymbol{x})) \Big) \tag{2}$$

where ℓ^p is KL divergence, ℓ^f is a distance function in the feature space, λ_τ^p and λ_τ^f are their domain-specific weights for task τ. We illustrate this key idea in Fig. 2. In words, the multi-domain network is optimized to match the domain-specific features up to a transformation (*i.e.* A_{θ_τ}) and predict the ground-truth classes y_τ.

While Li and Bilen [27] show that L2 distance is effective to match the features across multi-task and task-specific networks, which are trained for different tasks on a single domain, here we argue that learning to match features that are trained on substantially diverse domains require a more complex distance function to model non-linear correlations between the representations. To this end, inspired by [22], we propose to adopt the Centered Kernel Alignment (CKA) [22] similarity index with the Radial Basis Function (RBF) kernel that is shown to be capable of encoding meaningful non-linear similarities between representations of higher dimension than the number of data points.

Next we briefly describe CKA. Given a set of images $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, let $\mathbf{M} = [A_{\theta_\tau} \circ f_\phi(\boldsymbol{x}_1), \ldots, A_{\theta_\tau} \circ f_\phi(\boldsymbol{x}_n)]^\top \in \mathrm{R}^{n \times d}$ and $\mathbf{Y} = [f_{\phi_\tau^*}(\boldsymbol{x}_1), \ldots, f_{\phi_\tau^*}(\boldsymbol{x}_n)]^\top \in \mathrm{R}^{n \times d}$ denote the features that are computed by the multi-domain network adapted by A_{θ_τ} and domain-specific networks respectively for a given set of images $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. We first compute the RBF kernel matrices $\mathbf{P}$ and $\mathbf{T}$ of $\mathbf{M}$ and $\mathbf{Y}$ respectively and then use two kernel matrices $\mathbf{P}$ and $\mathbf{T}$ to measure CKA similarity between $\mathbf{M}$ and $\mathbf{Y}$:

$$\mathrm{CKA}(\mathbf{M}, \mathbf{Y}) = \mathrm{tr}(\mathbf{PHTH}) / \sqrt{\mathrm{tr}(\mathbf{PHPH})\mathrm{tr}(\mathbf{THTH})}, \tag{3}$$

where $\mathrm{tr}(\cdot)$ and $\mathbf{H}$ denote the trace of a matrix and centering matrix $\mathbf{H}_n = \mathbf{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ respectively. The loss $\ell^f(\mathbf{M}, \mathbf{Y})$ can be derived as $\ell^f(\mathbf{M}, \mathbf{Y}) = 1 - \mathrm{CKA}(\mathbf{M}, \mathbf{Y})$ as *dissimilarity* between the multi-domain and domain-specific features. As the original CKA similarity requires the computation of the kernel matrices over the whole datasets, which is not scalable to large datasets, we follow [34] and com-

pute them over each minibatch in our training. We refer to [22, 34] for more details.

3.3. Adapting multi-domain features

During meta-test, given a support set $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|\mathcal{S}|}$ of a new learning task, we use the multi-domain model to extract features $\{f_\phi(\boldsymbol{x}_i)\}_{i=1}^{|\mathcal{S}|}$ and adapt them to the target task. To this end, we apply a linear transformation $A_\vartheta : \mathrm{R}^d \rightarrow \mathrm{R}^d$ with learnable parameters ϑ to the computed features, *i.e.* $\{\boldsymbol{z}_i\}_{i=1}^{|\mathcal{S}|} = \{A_\vartheta \circ f_\phi(\boldsymbol{x}_i)\}_{i=1}^{|\mathcal{S}|}$ where $\vartheta \in \mathrm{R}^{d \times d}$. Then we follow a similar pipeline to the one in [13, 32, 48] to build a centroid classifier by averaging the embeddings belonging to this class:

$$\boldsymbol{c}_j = \frac{1}{|\mathcal{S}_j|} \sum_{\boldsymbol{z}_i \in \mathcal{S}_j} \boldsymbol{z}_i, \mathcal{S}_j = \{\boldsymbol{z}_k : y_k = j\}, j = 1, \ldots, C \tag{4}$$

where C is the number of classes in the support set. Next we estimate the likelihood of a support sample $\boldsymbol{z}$ by:

$$p(y = l|\boldsymbol{z}) = \frac{\exp(-d(\boldsymbol{z}, \boldsymbol{c}_l))}{\sum_{j=1}^{C} \exp(-d(\boldsymbol{z}, \boldsymbol{c}_j))}, \tag{5}$$

where $d(\boldsymbol{z}, \boldsymbol{c}_l)$ is the negative cosine similarity. We then optimize ϑ to minimize the following objective on the support set $\mathcal{S}$:

$$\min_{\vartheta} \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{x}_i, y_i \in \mathcal{S}} log(p(y = y_i|\boldsymbol{x}_i)). \tag{6}$$

Solving Eq. (6) for ϑ results in high intra-class and low inter-class similarity in the adapted space. We then use ϑ and Eq. (5) to predict the label of the query sample from $\mathcal{Q}$ by picking the closest centroid $\boldsymbol{c}_j$. Our meta-test pipeline is illustrated Fig. 3.

Discussion. Simple CNAPS [2] uses the (squared) Mahalanobis distance between the features of class centroid and a query image, $d(\boldsymbol{z}, \boldsymbol{c}) = \frac{1}{2}(f_\phi(\boldsymbol{x}) - \boldsymbol{c}')^\top \mathbf{Q}^{-1}(f_\phi(\boldsymbol{x}) - \boldsymbol{c}')$ where $\mathbf{Q}$ is a covariance matrix specific to the task and class and $\boldsymbol{c}'$ is the class centroid in the feature space (before the adaptation). The authors show that considering the class covariance enables better adaptation of the feature extractor to the target task. Our adaptation strategy can be seen as a generalization of the Mahalanobis distance computation. Assuming that $\mathbf{Q}^{-1}$ can be decomposed into a product of a lower triangular matrix and its conjugate transpose, *i.e.* $\mathbf{Q}^{-1} = \mathbf{L}\mathbf{L}^\top$, one can first pre-transform the features by multiplication, *i.e.* $\boldsymbol{z} = \mathbf{L}^\top f_\phi(\boldsymbol{x})$ and then compute the distance between these features and centroids. Similarly, we apply a linear transformation to the features but unlike [2], we learn its parameters ϑ by optimizing Eq. (6).

4. Experiments

Here we first describe the benchmarks, implementation details and competing methods. Then we rigorously compare our method to the state-of-the-art few-shot classification methods, study each proposed component in an ablation and also analyze our method qualitatively. Finally we evaluate our method in a global retrieval task to further evaluate the learned feature representations in few-shot classification task.

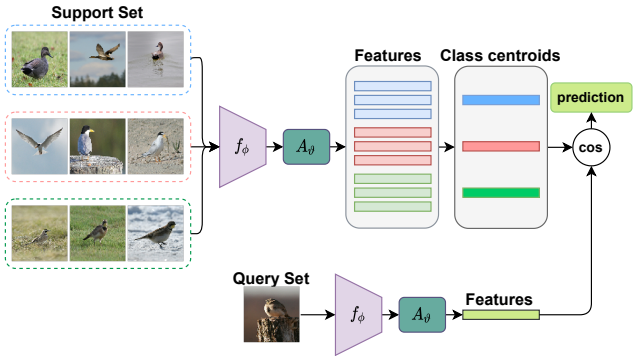

Figure 3. **Feature adaptation procedure in meta-test**. Given a support set and query image, our method learns to map their features to a task-specific space through a linear transformation A_ϑ and assign the query image to the nearest class center.

4.1. Experimental setup

Dataset. Meta-Dataset [52] is a few-shot classification benchmark that initially consisted of ten image datasets: ILSVRC_2012 [46] (ImageNet), Omniglot [25], FGVC-Aircraft [31] (Aircraft), CUB-200-2011 [54] (Birds), Describable Textures [9] (DTD), QuickDraw [19], FGVCx Fungi [4] (Fungi), VGG Flower [36] (Flower), Traffic Signs [18] and MSCOCO [28] then further expanded with MNIST [26], CIFAR-10 [23] and CIFAR-100 [23]. We follow the standard procedure and use the first eight datasets for meta-training, in which each dataset is further divided into train, validation and test set with disjoint classes. The evaluation within these datasets is used to measure the generalization ability in the seen domains. The rest five datasets are reserved as unseen domain for meta-test for measuring the cross-domain generalization ability.

Implementation details. We use PyTorch [38] library to implement our method. In all experiments we build our method on ResNet-18 [16] backbone for both single-domain and multi-domain networks. In the multi-domain network, we share all the layers but the last classifier across the domains. For training single-domain models, we strictly follow the training protocol in [13], use a SGD optimizer with a momentum and the cosine annealing learning scheduler with the same hyperparameters. For our multi-domain network, we use the same optimizer and scheduler as before, train it for 240,000 iterations. We set λ^f and λ^p to 4 for ImageNet and 1 for other datasets and use early-stopping based on cross-validation over the validations sets of 8 training

Test Dataset	Proto-MAML [52]	BOHB-E [47]	CNAPS [44]	Simple CNAPS [2]	SUR [13]	URT [29]	Best SDL	MDL	Ours
ImageNet	46.5 ± 1.1	51.9 ± 1.1	50.8 ± 1.1	58.4 ± 1.1	56.2 ± 1.0	56.8 ± 1.1	55.8 ± 1.0	53.4 ± 1.1	$\mathbf{58.8 \pm 1.1}$
Omniglot	82.7 ± 1.0	67.6 ± 1.2	91.7 ± 0.5	91.6 ± 0.6	94.1 ± 0.4	94.2 ± 0.4	93.2 ± 0.5	93.8 ± 0.4	$\mathbf{94.5 \pm 0.4}$
Aircraft	75.2 ± 0.8	54.1 ± 0.9	83.7 ± 0.6	82.0 ± 0.7	85.5 ± 0.5	85.8 ± 0.5	85.7 ± 0.5	86.6 ± 0.5	$\mathbf{89.4 \pm 0.4}$
Birds	69.9 ± 1.0	70.7 ± 0.9	73.6 ± 0.9	74.8 ± 0.9	71.0 ± 1.0	76.2 ± 0.8	71.2 ± 0.9	78.5 ± 0.8	$\mathbf{80.7 \pm 0.8}$
Textures	68.2 ± 0.8	68.3 ± 0.8	59.5 ± 0.7	68.8 ± 0.9	71.0 ± 0.8	71.6 ± 0.7	73.0 ± 0.6	71.4 ± 0.7	$\mathbf{77.2 \pm 0.7}$
Quick Draw	66.8 ± 0.9	50.3 ± 1.0	74.7 ± 0.8	76.5 ± 0.8	81.8 ± 0.6	82.4 ± 0.6	$\mathbf{82.8 \pm 0.6}$	81.5 ± 0.6	82.5 ± 0.6
Fungi	42.0 ± 1.2	41.4 ± 1.1	50.2 ± 1.1	46.6 ± 1.0	64.3 ± 0.9	64.0 ± 1.0	65.8 ± 0.9	61.9 ± 1.0	$\mathbf{68.1 \pm 0.9}$
VGG Flower	88.7 ± 0.7	87.3 ± 0.6	88.9 ± 0.5	90.5 ± 0.5	82.9 ± 0.8	87.9 ± 0.6	87.0 ± 0.6	88.7 ± 0.6	$\mathbf{92.0 \pm 0.5}$
Traffic Sign	52.4 ± 1.1	51.8 ± 1.0	56.5 ± 1.1	57.2 ± 1.0	51.0 ± 1.1	48.2 ± 1.1	47.4 ± 1.1	51.0 ± 1.0	$\mathbf{63.3 \pm 1.1}$
MSCOCO	41.7 ± 1.1	48.0 ± 1.0	39.4 ± 1.0	48.9 ± 1.1	52.0 ± 1.1	51.5 ± 1.1	53.5 ± 1.0	49.6 ± 1.1	$\mathbf{57.3 \pm 1.0}$
MNIST	-	-	-	94.6 ± 0.4	94.3 ± 0.4	90.6 ± 0.5	89.8 ± 0.5	94.4 ± 0.3	$\mathbf{94.7 \pm 0.4}$
CIFAR-10	-	-	-	$\mathbf{74.9 \pm 0.7}$	66.5 ± 0.9	67.0 ± 0.8	67.3 ± 0.8	66.7 ± 0.8	74.2 ± 0.8
CIFAR-100	-	-	-	61.3 ± 1.1	56.9 ± 1.1	57.3 ± 1.0	56.6 ± 0.9	53.6 ± 1.0	$\mathbf{63.5 \pm 1.0}$
Average Rank	7.8	8.1	6.6	5.2	5.0	4.4	4.8	4.6	1.3

Table 1. **Comparison to baselines and state-of-the-art methods on Meta-Dataset**. Mean accuracy, 95% confidence interval are reported. The first eight datasets are seen during training and the last five datasets are unseen and used for test only. Average rank is computed according to first 10 datasets as some methods do not report results on last three datasets.

datasets. We refer to supplementary for more details.

Baselines and compared methods. First we compare our method to our own baselines, i) the best single-domain model (Best SDL) where we use each single-domain network as the feature extractor and test it for few-shot classification in each dataset and pick the best performing model (see supplementary for the complete results). This involves evaluating 8 single-domain networks on 13 datasets, serves a very competitive baseline, ii) the vanilla multi-domain learning baseline (MDL) that is learning by optimizing Eq. (1) without the proposed distillation method. As an additional baseline, we include the best performing method in [52], *i.e*. Proto-MAML [52], and as well as the state-of-the-art methods, OHB-E [47], CNAPS [44], SUR [13], URT [29], and the Simple CNAPS [2][1]. For evaluation, we follow the standard protocol in [52], randomly sample 600 tasks for each dataset, and report average accuracy and 95% confidence score in all experiments. We reproduce results by training and evaluating SUR [13], URT [29], and Simple CNAPS [2] using their code for fair comparison as recommended by Meta-Dataset.

4.2. Results

As in Meta-Dataset [52], we sample each task with varying number of ways and shots and report the results in Table 1. Our method outperforms the state-of-the-art methods in seven out of eight seen datasets and four out of five unseen datasets. We also compute average rank as recommended in [52], our method ranks 1.3 in average and the state-of-the-art methods SUR and URT rank 5.0 and 4.4, respectively. More specifically, we obtain significantly better results than the second best approach on Aircraft (+2.8), Birds (+2.1), Texture (+4.2), and VGG Flower (+1.5) for seen domains and Traffic Sign (+6.1)[2] and MSCOCO (+3.8). The results show that jointly learning a single set of representations provides better generalization ability than fusing the ones from multiple single-domain feature extractors as done in SUR and URT. Notably, our method requires less parameters and computations to run during inference than SUR and URT, as it runs only one universal network to extract features, while both SUR and URT need to pass the query set to multiple single-domain network.

We also see that our method outperforms two strong baselines, Best SDL and MDL in all datasets except in Quick-Draw. This indicates that i) universal representations are superior to the single-domain ones while generalizing to new tasks in both seen and unseen domains, while requiring significantly less number of parameters (1 vs 8 neural networks), ii) our distillation strategy is essential to obtain good multi-domain representations. While MDL outperforms the best SDL in certain domains by transferring representations across them, its performance is lower in other domains than SDL, possibly due to negative transfer across the significantly diverse domains. Surprisingly, MDL achieves the third best in average rank, indicating the benefit of multi-domain representations.

4.3. Further results

Varying-way five-shot setting. After reporting results in a broad range of varying shots (*e.g*. up to 100 shots in some extreme cases), we further analyze our method for 5-shot setting with varying number of categories. We follow the procedure in [12], sample a varying number of ways in Meta-Dataset as in the standard setting but a fixed number of

[1]Results of Proto-MAML [52], BOHB-E [47], and CNAPS [44] are obtained from Meta-Dataset.

[2]The accuracy of all methods on Traffic Sign is different from the one in the original papers as one bug has been fixed in Meta-Dataset repository. See `https://github.com/google-research/meta-dataset/issues/54` for more details. As mentioned in the Meta-Dataset repository, we further update the evaluation protocol and report the updated results of all methods in the supplementary.

Test Dataset	Varying-Way Five-Shot				Five-Way One-Shot			
	Simple CNAPS [2]	SUR [13]	URT [29]	Ours	Simple CNAPS [2]	SUR [13]	URT [29]	Ours
ImageNet	47.2	46.7	48.6	**49.4**	42.6	40.7	47.4	**49.6**
Omniglot	95.1	95.8	**96.0**	**96.0**	93.1	93.0	95.6	**95.8**
Aircraft	74.6	82.0	81.2	**84.8**	65.8	67.1	77.9	**79.6**
Birds	69.6	62.8	71.2	**76.0**	67.9	59.2	70.9	**74.9**
Textures	57.5	60.2	65.2	**69.1**	42.2	42.5	49.4	**53.6**
Quick Draw	70.9	79.0	**79.2**	78.2	70.5	**79.8**	79.6	79.0
Fungi	50.3	66.5	66.8	**70.0**	58.3	64.8	71.0	**75.2**
VGG Flower	86.5	76.9	82.4	**89.3**	**79.9**	65.0	72.7	**79.9**
Traffic Sign	55.2	44.9	45.1	**57.5**	55.3	44.6	52.6	**57.9**
MSCOCO	49.2	48.1	52.3	**56.1**	48.8	47.8	56.9	**59.1**
MNIST	88.9	**90.1**	86.5	89.7	**80.1**	77.0	75.6	78.7
CIFAR-10	**66.1**	50.3	61.4	66.0	50.3	35.8	47.3	**54.7**
CIFAR-100	53.8	46.4	52.5	**57.0**	53.8	42.9	54.9	**61.8**
Average Rank	3.1	3.0	2.5	1.3	2.8	3.5	2.4	1.2

Table 2. **Results for varying-Way five-Shot and five-Way one-Shot settings**. Mean accuracies are reported and the results with confidence interval are shown in the supplementary.

Test Dataset	L2	COSINE	CKA	KL	CKA + KL
ImageNet	55.7 ± 1.1	57.0 ± 1.1	$\mathbf{59.0 \pm 1.0}$	57.0 ± 1.1	58.8 ± 1.1
Omniglot	94.0 ± 0.4	94.1 ± 0.4	$\mathbf{94.7 \pm 0.4}$	94.5 ± 0.4	94.5 ± 0.4
Aircraft	87.4 ± 0.5	88.3 ± 0.5	88.9 ± 0.5	89.3 ± 0.4	$\mathbf{89.4 \pm 0.4}$
Birds	78.5 ± 0.7	77.5 ± 0.8	80.4 ± 0.7	78.6 ± 0.8	$\mathbf{80.7 \pm 0.8}$
Textures	72.8 ± 0.6	73.2 ± 0.7	74.5 ± 0.7	73.3 ± 0.7	$\mathbf{77.2 \pm 0.7}$
Quick Draw	81.2 ± 0.6	80.8 ± 0.6	81.9 ± 0.6	81.6 ± 0.6	$\mathbf{82.5 \pm 0.6}$
Fungi	65.7 ± 0.9	65.9 ± 0.9	66.4 ± 0.9	67.6 ± 0.9	$\mathbf{68.1 \pm 0.9}$
VGG Flower	87.5 ± 0.6	85.0 ± 0.6	91.3 ± 0.5	89.6 ± 0.5	$\mathbf{92.0 \pm 0.5}$
Traffic Sign	61.6 ± 1.1	59.5 ± 1.1	63.2 ± 1.1	62.5 ± 1.2	$\mathbf{63.3 \pm 1.2}$
MSCOCO	53.4 ± 1.0	53.8 ± 1.1	56.6 ± 1.0	55.6 ± 1.1	$\mathbf{57.3 \pm 1.0}$
MNIST	94.7 ± 0.3	93.2 ± 0.5	94.7 ± 0.4	$\mathbf{95.3 \pm 0.4}$	94.7 ± 0.4
CIFAR-10	71.1 ± 0.8	68.1 ± 0.8	73.8 ± 0.7	72.9 ± 0.8	$\mathbf{74.2 \pm 0.8}$
CIFAR-100	59.1 ± 1.0	58.1 ± 1.0	62.1 ± 1.0	60.8 ± 1.0	$\mathbf{63.6 \pm 1.0}$

Table 3. **Quantitative analysis of knowledge distillation loss functions**. Mean accuracy, 95% confidence interval are reported. COSINE and KL denote negative cosine similarity and KL divergence respectively. All the loss functions are applied to measure the difference between intermediate representations of neural networks except KL, which is applied to network predictions. All results are obtained with feature adaptation during meta-test stage.

shots to form balanced support and query sets and compare our method to the top three performing methods, Simple CNAPS, SUR and URT. As depicted in Table 2, overall performance for all methods decreases in most datasets compared to results in Table 1, as five-shot setting samples much less support images than the standard setting. The ranking of different methods change slightly. The top-2 methods remain the same, while both Simple CNAPS and SUR obtain 3.1 and 3.0 average rank, respectively. SUR performs the best on MNIST, Simple CNAPS outperforms others on CIFAR-10 and URT is top-1 on Quick Draw. Ours still achieves significantly better performance than other methods on the rest ten datasets.

Results in five-way one-shot setting. Next we test an extremely challenging five-way one-shot setting on Meta-Dataset. For each task, only one image per class is provided in support set. This setting is often used in evaluating different methods in a single domain [25, 43, 53], here we adopt it for multiple domains and report the results in Table 2. Our

Test Dataset	NCC	NCC+MD	LR	SVM	Ours
ImageNet	57.0 ± 1.1	53.9 ± 1.0	56.0 ± 1.1	54.5 ± 1.1	$\mathbf{58.8 \pm 1.1}$
Omniglot	94.4 ± 0.4	93.8 ± 0.5	93.7 ± 0.5	94.3 ± 0.5	$\mathbf{94.5 \pm 0.4}$
Aircraft	88.0 ± 0.5	87.6 ± 0.5	88.3 ± 0.6	87.7 ± 0.5	$\mathbf{89.4 \pm 0.4}$
Birds	80.3 ± 0.7	78.3 ± 0.7	79.7 ± 0.8	78.1 ± 0.8	$\mathbf{80.7 \pm 0.8}$
Textures	74.6 ± 0.7	73.7 ± 0.7	74.7 ± 0.7	73.8 ± 0.8	$\mathbf{77.2 \pm 0.7}$
Quick Draw	81.8 ± 0.6	80.9 ± 0.7	80.0 ± 0.7	80.0 ± 0.6	$\mathbf{82.5 \pm 0.6}$
Fungi	66.2 ± 0.9	57.7 ± 0.9	62.1 ± 0.8	58.5 ± 0.9	$\mathbf{68.1 \pm 0.9}$
VGG Flower	91.5 ± 0.5	89.7 ± 0.6	91.1 ± 0.5	91.4 ± 0.6	$\mathbf{92.0 \pm 0.5}$
Traffic Sign	49.8 ± 1.1	62.2 ± 1.1	59.7 ± 1.1	$\mathbf{65.7 \pm 1.2}$	63.3 ± 1.2
MSCOCO	54.1 ± 1.0	48.5 ± 1.0	51.2 ± 1.1	50.5 ± 1.0	$\mathbf{57.3 \pm 1.0}$
MNIST	91.1 ± 0.4	95.1 ± 0.4	93.5 ± 0.5	$\mathbf{95.4 \pm 0.4}$	94.7 ± 0.4
CIFAR-10	70.6 ± 0.7	68.9 ± 0.8	73.1 ± 0.8	72.0 ± 0.8	$\mathbf{74.2 \pm 0.8}$
CIFAR-100	59.1 ± 1.0	60.0 ± 0.9	60.1 ± 1.1	60.5 ± 1.1	$\mathbf{63.6 \pm 1.0}$

Table 4. **Quantitative analysis of several classifiers that are incorporated to our method during meta-test stage**. NCC, MD, LR, SVM denote nearest center classifier, Mahalanobis distance, logistic regression, support vector machines respectively.

method outperforms the prior work consistently as observed in previous two settings, which validates the importance of good universal representations when limited labeled samples are available in meta-test. Interestingly, Simple CNAPS achieves better rank than SUR in this setting unlike the previous settings.

4.4. Further analysis

Here we conduct ablation studies on different components in our framework by varying the loss function for the distillation, classifier type in meta-test.

Different distillation loss functions. First we study different distillation loss functions, including L2 loss, cosine distance, KL divergence and CKA for learning the multi-domain networks and report their performances in Table 3. While KL divergence is applied to match the logits of single and multi-domain networks as in [17], the other loss functions are used to match the intermediate representations (features that are fed into classifiers) between those models. Among the individual loss functions, the best results are obtained either with CKA or KL divergence loss, while CKA outperforms KL divergence in the most domains. Although the features are first aligned with an adapter, L2 and cosine loss functions are not sufficient to match features from very diverse domains and further aligning features with CKA is crucial. Note that here L2 baselines corresponds to the method of [27]. Finally, combining CKA with KL divergence gives the best performance over the multi-domain models that are trained with the individual loss functions.

Different classifiers in meta-test. Next we evaluate the proposed adaptation strategy with nearest centroid classifier (NCC), described in Section 3.3, to different parametric including Support Vector Machines (SVM), Logistic Regression (LR) as in [51] and non-parametric classifiers including NCC without the adaptive mapping and NCC with Mahalanobis Distance (NCC+MD) in [2] in Table 4. For non-parametric classifiers, NCC performs best in unseen do-

Test Dataset	ImageNet		Omniglot		Aircraft		Birds		Textures		Quick Draw		Fungi		VGG Flower		Traffic Sign		MSCOCO		MNIST		CIFAR-10		CIFAR-100	
Recall@k	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
Sum	22.1	30.3	84.7	91.8	69.7	80.7	45.9	59.7	66.3	78.2	77.4	84.3	31.9	42.9	85.1	92.1	94.6	97.2	62.6	71.2	98.3	99.2	54.0	68.9	27.8	37.4
Concate	20.2	28.0	84.4	91.5	44.3	58.1	35.5	48.8	68.8	78.2	73.0	80.8	30.7	40.4	83.4	91.3	**95.1**	**97.3**	60.7	69.8	**98.7**	**99.3**	49.7	65.3	25.4	34.6
MDL	29.8	39.6	**89.8**	**94.3**	80.3	87.1	63.2	75.9	67.0	77.1	79.5	85.4	40.2	51.7	86.9	93.3	89.5	94.1	63.6	72.6	97.6	98.8	58.9	72.9	31.6	42.0
Simple CNAPS [2]	34.0	43.8	84.9	91.6	70.5	82.5	55.9	70.5	64.8	76.9	75.3	83.0	29.1	39.0	88.1	94.1	79.9	86.9	65.2	73.8	97.5	98.8	**66.2**	**79.3**	33.2	44.2
Ours	**36.1**	**46.2**	89.7	**94.3**	**83.3**	**90.4**	**66.7**	**78.9**	**70.2**	**80.8**	**79.9**	**86.5**	**44.5**	**56.2**	**90.0**	**94.6**	87.9	93.0	**67.4**	**76.3**	97.0	98.4	62.1	76.5	**35.1**	**46.1**

Table 5. **Global retrieval performance on Meta-Dataset**. Here we evaluate our method in a non-episodic retrieval task to further compare the generalization ability of our universal representations.

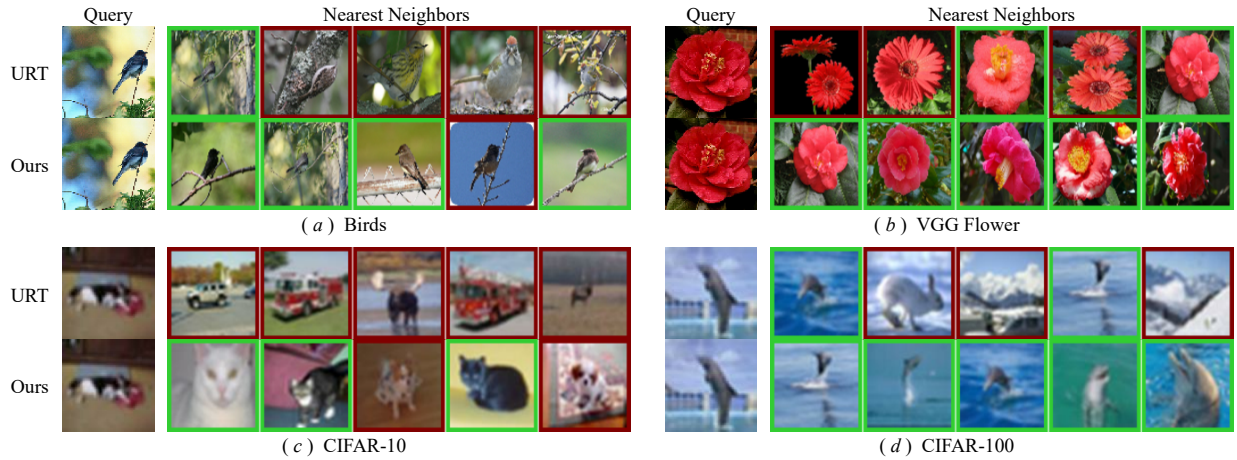

Figure 4. **Qualitative analysis of our method in four datasets**. Green and red colors indicate correct and false predictions respectively.

mains when used with Mahalanobis distance. The parametric classifiers, SVM and LR that are trained on the limited support set obtain very competitive results and outperform the non-parametric ones in most domains. Our method, which combines the benefit of parametric and non-parametric classifiers, outperforms SVM, LR and NCC+MD in most seen datasets, while achieves worse in some unseen domains like Traffic Sign and MNIST.

Qualitative results. We also qualitatively analyze our method and compare it to URT [29] in Fig. 4 by illustrating the nearest neighbors in four different datasets given a query image (see supplementary for more examples). While URT retrieves images with more similar colors, shapes and backgrounds, while our method is able to retrieve semantically similar images and finds more correct neighbors than URT. It again suggests that our method is able to learn more semantically meaningful and general representations.

4.5. Global retrieval

Here we go beyond the few-shot classification experiments and evaluate the generalization ability of our representations that are learned in the multi-domain network in a retrieval task, inspired from metric learning literature [37, 55]. To this end, for each test image, we find the nearest images in entire test set in the feature space and test whether they correspond to the same category. For evaluation metric, we use Recall@k which considers the predictions with one of the k closest neighbors with the same label as positive. In Table 5, we compare our method with Simple CNAPS in Recall@1 and Recall@2 (see supplementary for more results). URT and SUR require adaptation using support set and no such adaptation in retrieval task is possible, we replace them with two baselines that concatenate or sum features from multiple domain-specific networks. Our method achieves the best performance in ten out of thirteen domains with significant gains in Aircraft, Birds, Textures and Fungi. This strongly suggests that our multi-domain representations are the key to the success of our method in the previous few-shot classification tasks.

5. Conclusion

In this work, we demonstrate that learning a single set of universal representations integrated with a feature refining step achieves state-of-the-art performance in the recent Meta-Dataset benchmark. To this end, we propose to optimize the weights of a deep neural network simultaneously over multiple domains by aligning its features with multiple single-domain networks through linear adapters and a loss function that is inspired from CKA. We show that the universal features can be further refined from few examples to unseen tasks by learning a transformation in a similar spirit to distance learning. Our method outperforms the state-of-the-art techniques while using less number of parameters and being more computationally efficient than other multi-domain techniques.

Acknowledgments. HB is supported by the EPSRC programme grant Visual AI EP/T028572/1.

References

[1] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. In *ICLR*, 2019.

[2] Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. Improved few-shot visual classification. In *CVPR*, pages 14493–14502, 2020.

[3] Hakan Bilen and Andrea Vedaldi. Universal representations: The missing link between faces, text, planktons, and cat breeds. *arXiv preprint arXiv:1701.07275*, 2017.

[4] Schroeder Brigit and Cui Yin. Fgvcx fungi classification challenge. *online*, 2018.

[5] John Bronskill, Jonathan Gordon, James Requeima, Sebastian Nowozin, and Richard Turner. Tasknorm: Rethinking batch normalization for meta-learning. In *ICML*, pages 1153–1164, 2020.

[6] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019.

[7] Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. A new meta-baseline for few-shot learning. *arXiv preprint arXiv:2003.04390*, 2020.

[8] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, pages 794–803. PMLR, 2018.

[9] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014.

[10] Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D Manning, and Quoc V Le. Bam! born-again multi-task networks for natural language understanding. In *ACL*, 2019.

[11] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *ICLR*, 2020.

[12] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. In *NeurIPS*, 2020.

[13] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Selecting relevant features from a multi-domain representation for few-shot classification. In *ECCV*, pages 769–786, 2020.

[14] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICLR*, pages 1126–1135, 2017.

[15] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *ICML*, 2018.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning Workshop*, 2014.

[18] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *IJCNN*, pages 1–8. Ieee, 2013.

[19] Jonas Jongejan, Rowley Henry, Kawashima Takashi, Kim Jongmin, and Fox-Gieg Nick. The quick, draw! a.i. experiment. *online*, 2016.

[20] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, 2018.

[21] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[22] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, pages 3519–3529. PMLR, 2019.

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009.

[24] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

[25] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[27] Wei-Hong Li and Hakan Bilen. Knowledge distillation for multi-task learning. In *ECCV Workshop on Imbalance Problems in Computer Vision*, pages 163–176. Springer, 2020.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[29] Lu Liu, William Hamilton, Guodong Long, Jing Jiang, and Hugo Larochelle. A universal representation transformer layer for few-shot image classification. In *ICLR*, 2021.

[30] Jiaqi Ma and Qiaozhu Mei. Graph representation learning via multi-task knowledge distillation. In *NeurIPS GRL Workshop*, 2019.

[31] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[32] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *TPAMI*, 35(11):2624–2637, 2013.

[33] Erik G Miller, Nicholas E Matsakis, and Paul A Viola. Learning from one example through shared densities on transforms. In *CVPR*, volume 1, pages 464–471. IEEE, 2000.

[34] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *ICLR*, 2021.

[35] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[36] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008*

Sixth Indian Conference on Computer Vision, Graphics & Image Processing, pages 722–729. IEEE, 2008.

[37] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, pages 4004–4012, 2016.

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035. Curran Associates, Inc., 2019.

[39] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, volume 32, 2018.

[40] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *ICML*, pages 5142–5151, 2019.

[41] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *NeurIPS*, 2017.

[42] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *CVPR*, pages 8119–8127, 2018.

[43] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018.

[44] James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In *CVPR*, 2019.

[45] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[47] Tonmoy Saikia, Thomas Brox, and Cordelia Schmid. Optimized generic feature learning for few-shot classification across domains. *arXiv preprint arXiv:2001.07926*, 2020.

[48] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

[49] 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 *CVPR*, pages 1199–1208, 2018.

[50] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2020.

[51] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *ECCV*, 2020.

[52] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *ICLR*, 2020.

[53] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, 2016.

[54] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *California Institute of Technology*, 2011.

[55] Lu Yu, Vacit Oguz Yazici, Xialei Liu, Joost van de Weijer, Yongmei Cheng, and Arnau Ramisa. Learning metrics from teachers: Compact networks for image embedding. In *CVPR*, pages 2907–2916, 2019.

[56] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *ICLR*, 2020.