Transferable Meta Learning Across Domains

Bingyi Kang & Jiashi Feng
Department of Electrical and Computer Engineering
National University of Singapore
kang@u.nus.edu, elefjia@nus.edu.sg

Abstract

Meta learning algorithms are effective at obtaining meta models with the capability of solving new tasks quickly. However, they critically require sufficient tasks for meta model training and the resulted model can only solve new tasks similar to the training ones. These limitations make them suffer performance decline in presence of insufficiency of training tasks in target domains and task heterogeneity—the source (model training) tasks presents different characteristics from target (model application) tasks. To overcome these two significant limitations of existing meta learning algorithms, we introduce the cross-domain meta learning framework and propose a new transferable meta learning (TML) algorithm. TML performs meta task adaptation jointly with meta model learning, which effectively narrows divergence between source and target tasks and enables transferring source meta-knowledge to solve target tasks. Thus, the resulted transferable meta model can solve new learning tasks in new domains quickly. We apply the proposed TML to cross-domain few-shot classification problems and evaluate its performance on multiple benchmarks. It performs significantly better and faster than well-established meta learning algorithms and fine-tuned domain-adapted models.

1 Introduction

Meta learning aims at obtaining a model that can capture common characteristics across different learning tasks, such that the learned model can adapt to new tasks quickly. Recently, various meta learning methods (Hariharan & Girshick, 2016; Koch et al., 2015; Lake et al., 2013; Ravi & Larochelle, 2016; Santoro et al., 2016b; Vinyals et al., 2016) have been developed to solve multiple challenging problems, *e.g.*, few-shot classification (Fei-Fei et al., 2006), and achieved promising performance. Those methods devise different approaches to train a meta model that can be applied to new tasks via simple fine-tuning.In contrast to conventional supervised learning methods that suffer poor generalization performance, meta learning methods explicitly optimize the model generalization ability to new tasks and therefore achieve better performance.

Under the standard meta learning paradigm, the meta model is trained on a meta-training dataset consisting of sufficient training tasks and evaluated on another dataset with novel tasks. However, existing meta learning methods usually assume the training and test tasks have similar characteristics. For instance, for few-shot classification tasks, the samples of different tasks are usually from splits of the *same* dataset (Finn et al., 2017; Hariharan & Girshick, 2016; Vinyals et al., 2016). This actually deviates from the real world scenarios where a pre-trained meta model usually needs to be applicable to heterogeneous tasks in different domains. Moreover, constructing the meta-training set demands sufficiently many labeled examples, which are usually not available in practice considering the "few-shot" nature of meta learning problems. Existing meta learning methods generally ignore such discrepancy between the traditional meta learning paradigm and realistic application scenarios, leading to poor generalization ability of the obtained model to new tasks in new domains.

To extend applicability of meta learning methods, we propose a new framework to utilize data from another (source) domain to construct the meta-training set and aim to develop a new meta learning algorithm to learn a meta model from the source domain that can be directly applied to target domains, without requiring further meta-training. We term this new framework as the cross-domain meta learning. See Fig. 1 for an exam-

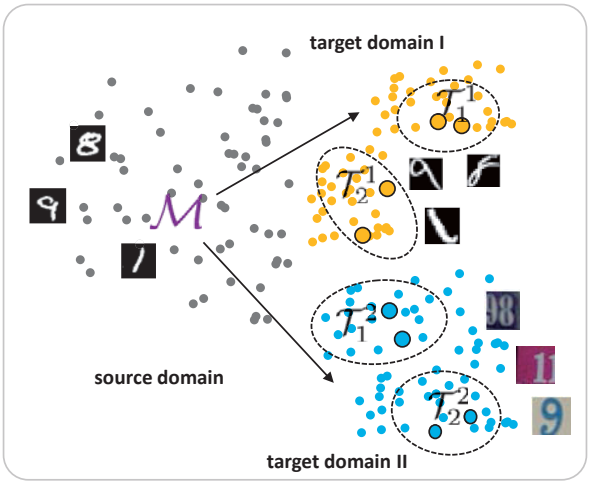

Figure 1: Illustration of the cross-domain meta learning. The pre-trained meta learning model $\mathcal{M}$ in the source domain is applied to solve new learning tasks $\mathcal{T}_1^1, \ldots, \mathcal{T}_2^2$ in different target domains. The target domains have too few labeled samples to construct sufficient meta-training tasks. Our proposed TML algorithm solves this problem by learning a transferable meta model which can be directly applied to target tasks.

ple. Developing cross-domain meta learning algorithms is difficult due to the scarcity of meta-training examples in target domains and task heterogeneity caused by domain shift. As far as we know, none of existing meta learning methods can deal with these challenging issues.

To address the above challenges, we propose a novel transferable meta learning (TML) algorithm, which provides a meta model capable of fast adapting to new learning tasks in different domains via a few simple gradient descent fine-tuning. Inspired by state-of-the-art meta learning methods (Finn et al., 2017; Vinyals et al., 2016), TML introduces a new learning scheme. It first organizes available training data, very few of which are from target domains, to form a collection of cross-domain meta learning tasks. Taking these tasks as training examples, TML explicitly optimizes the capability of "learning to fast adapt" of the meta model. By taking sensitiveness of model parameters to different tasks and domain shift as the joint learning objective, TML effectively trains the meta model to learn task representations robust to domain-shift, enables cross-domain meta-knowledge transfer and makes the model fast adaptable to novel target tasks of different characteristics from the source ones.

TML trains a meta model in two alternating phases. In *meta learning*, TML optimizes the meta model to minimize the loss over all its task-specific fine-tuned models, *i.e.*, minimizing the meta loss. In *meta adaptation*, TML reinforces the meta model by adapting task representations to minimize domain divergence and thus facilitates meta-knowledge transfer across heterogeneous tasks. We use a domain discriminative loss for measuring domain divergence. Through these two phases, TML effectively minimizes the source domain meta loss and domain divergence jointly, which together serve as an accurate surrogate for learning to minimize the meta loss in the target domain. Therefore, a meta model trained by TML is readily applicable to solving new tasks in target domains.

We apply and evaluate TML for cross-domain few-shot learning problems, on multiple datasets with various domain shift issues. The results demonstrate that TML surpasses fine-tuning based methods and other meta learning models significantly in terms of few-shot classification accuracy and adaptation speed. To our best knowledge, TML is the first one that considers the illness of current meta learning frameworks and explicitly pursues generalization across heterogeneous tasks in different domains. It substantially extends existing meta learning algorithms and mitigates the gap between meta learning frameworks and realistic application scenarios.

2 Related Work

Meta Learning Recently, some meta learning (Koch et al., 2015; Ravi & Larochelle, 2016; Santoro et al., 2016b; Snell et al., 2017; Vinyals et al., 2016; Wang & Hebert, 2016) works are developed to solve the few-shot learning problems. A meta model is usually trained over a set of similar tasks to capture generalizable properties across tasks, such that it can fast adapt to new similar tasks. Several different strategies for designing meta-learning algorithms are adopted (Andrychowicz et al., 2016; Ravi & Larochelle, 2016). For instance, "learning to compare" aims to learn a comparison metric that can be used to find the most similar labeled sample for each unlabeled input (Koch et al., 2015; Mishra et al., 2017; Vinyals et al., 2016). Some meta learning methods adopt external memory to augment the model (Munkhdalai & Yu, 2017; Santoro et al., 2016a). For example, (Santoro et al., 2016a) builds a meta model upon a Neural Turing Machine (Graves et al., 2014), which encodes and writes labeled examples into the memory and retrieve relevant information from the memory for classifying an unlabeled sample. Differently, Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017) tries to find a proper intermediate model, which can be fine-tuned for several steps to produce a task-specific model given very few samples. However, all these existing models assume that training and testing tasks have similar characteristics, and suffer performance decline in presence of task heterogeneity. Moreover, they all require suffi-

ciently many training tasks. Our proposed TML algorithm is the first one that tries to achieve fast adaptation to new meta learning tasks in presence of varying in the task characteristics in applied domains.

Few-shot Learning Few-shot learning (Fei-Fei et al., 2006; Hariharan & Girshick, 2016; Lake et al., 2013) is proposed to learn to recognize new categories with few examples. (Fei-Fei et al., 2006) provides a solution based on Bayesian inference over a pre-trained model to capture general knowledge from previously learned categories, whose generalization ability however is limited by heavy dependency on the relation between previously seen and new objects. Recently, (Hariharan & Girshick, 2016; Luo et al., 2017) propose to transfer intra-class features from base classes to new classes. This method achieves good performance on new examples while maintaining the accuracy on original training classes. But all these conventional few-shot learning methods require retraining the model from scratch when applied to new few-shot learning tasks with randomly assigned labels, thus are incapable of fast adapting to multiple new tasks.

Domain Adaptation Many works have been developed for domain adaptation learning (Ganin et al., 2016; Hoffman et al., 2013; Liu & Tuzel, 2016; Motiian et al., 2017a,b; Tzeng et al., 2015, 2017). Maximum Mean Discrepancy (MMD) Tzeng et al. (2014) measures the distribution difference between the source and target domains by computing norm of the mean feature difference between two domains. (Long et al., 2015; Sun & Saenko, 2016) have shown that combining MMD with popular deep learning models is effective. More recently, Generative Adversarial Network (GAN) (Goodfellow et al., 2014) based models have achieved remarkable success, *e.g*. Adversarial Discriminative Domain Adaptation (ADDA) (Tzeng et al., 2017) and CoGAN (Liu & Tuzel, 2016). ADDA adapts a well-learned source CNN by learning a target CNN that maps target-domain images into a feature space, where they are indistinguishable from the source feature space by the GAN discriminator. However, existing domain adaptation methods cannot be applied to solve meta learning tasks.

3 The Proposed Algorithm

Meta learning aims to learn a meta model that captures generalizable properties across tasks, such that the model can adapt to solving new similar tasks quickly. In this work, we consider the few-shot classification tasks in particular, which are widely adopted for evaluating meta learning methods. We first define the problem of meta learning for few-shot classification. Then we elaborate on our target problem, cross-domain meta learning, and our proposed TML algorithm.

3.1 Problem Definition

Let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ be the label space. We are interested in meta learning for the N-category k-shot learning tasks, where only a small number of k annotated samples per category (*e.g*., $k \leq 5$) are available for training a classification model within each task.

Let $f_\theta(\cdot)$: $\mathcal{X} \to \mathcal{Y}$ denote the meta learning model with learnable parameter θ which is optimized to solve the following few-shot learning tasks:

$$\mathcal{T} \triangleq \{\underbrace{(x_1, y_1), \ldots, (x_{Nk}, y_{Nk})}_{\text{training samples}}, \underbrace{(\boldsymbol{x}_t, \boldsymbol{y}_t)}_{\text{test samples}}, f_\theta, \ell\}. \quad (1)$$

More specifically, each task is to learn a specific classification model $f_{\theta'}(\cdot)$ from only Nk training samples such that the following task-specific classification loss on test samples $(\boldsymbol{x}_t, \boldsymbol{y}_t)$ can be minimized:

$$\mathcal{L}_\mathcal{T}(f_{\theta'}) \triangleq \ell(f_{\theta'}(\boldsymbol{x}_t), \boldsymbol{y}_t), \quad (2)$$

where ℓ is the classification loss function.

Existing meta learning approaches generally learn a meta model $f_\theta(\cdot)$ through meta-training over a collection of tasks $\mathcal{T} \sim p(\mathcal{T})$ with similar distribution. In meta training, the meta model parameter θ is learned to minimize the *meta loss* computed from all the training tasks:

$$\theta = \arg\min_\theta \sum_{i=1}^{m} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i}), \mathcal{T}_i \sim p(\mathcal{T}), \quad (3)$$

where θ'_i is derived from the meta model θ through task-specific adaptation, *e.g*., by fine-tuning θ on training samples of task $\mathcal{T}_i$. When there are sufficiently many meta-training tasks $\mathcal{T}_i$ from the same distribution $p(\mathcal{T})$, *i.e*., m is sufficiently large, one can reliably obtain a well-performing meta model that can solve new task $\mathcal{T}_t \sim p(\mathcal{T})$ with satisfactory task-specific loss.

However, in many realistic few-shot learning problems, only a few labeled data are available which are not sufficient to form many tasks for performing meta-training in Eqn. (3). Thus the resulted meta model would suffer from insufficient meta-training and would not generalize well to new tasks. In this work, we propose to address this problem by constructing a meta-training set from another (source) domain of different characteristics where rich labeled data are available. Despite being promising and fitting realistic scenarios better, such a method brings a cross-domain meta learning problem as defined below. This problem is challenging to existing meta learning

methods as they usually assume the meta-training and meta-test tasks are from the same distribution. We aim to solve the following problem in this work, whose solution would also bring significant practical benefits in extending application of meta learning models to other heterogeneous tasks in different domains.

Definition 1 (Cross-domain Meta Learning)**.** *Suppose there are two different datasets of $\mathcal{X} \times \mathcal{Y}$ with domain shift in $\mathcal{X}$, called source data D_S and target data D_T, and D_T only provides very few labeled samples. We aim to learn a meta model $f_\theta(\cdot)$ by leveraging the sufficiently many source data D_S and their formed meta-training tasks $\mathcal{T}_i \in p_S(\mathcal{T})$, such that the model can generalize well to new tasks $\mathcal{T}_t \in p_T(\mathcal{T})$ in target dataset D_T with small task loss. Here $p_T(\mathcal{T})$ is different from $p_S(\mathcal{T})$ in terms of label space $\mathcal{Y}$ and data distribution $\mathcal{X}$.*

Directly training a meta model via minimizing the meta loss (Eqn. (3)) in target dataset D_T is infeasible due to limited labeled data and consequently insufficient meta-training tasks. On the other hand, although the source domain data is enough for training a meta model, directly applying it to the target domain will suffer poor generalization performance due to domain shift (verified by experiments in Sec. 4). To address this problem, we aim to fully utilize cross-domain knowledge to learn a powerful meta model f_θ, which is well prepared for fast adaptation to new few-shot learning tasks in target dataset D_T. To this end, we develop the transferable meta learning algorithm in this work.

3.2 Model-Agnostic Meta-Learning (MAML)

We develop our transferable meta learning (TML) algorithm from the state-of-the-art MAML algorithm (Finn et al., 2017). While we extend MAML here, our proposed idea is applicable to other meta-learning methods.

MAML solves above few-shot learning problems by learning the parameter θ such that f_θ can solve a new task rapidly via several gradient descent steps on few-shot task-related training examples. To this end, MAML forms a set of training tasks $\mathcal{T}=\{\mathcal{T}_1, \ldots, \mathcal{T}_m\}$, where each task instantiates an N-category k-shot classification problem $\mathcal{T}_i = \{(x_1^{(i)}, y_1^{(i)}), \ldots, (x_{Nk}^{(i)}, y_{Nk}^{(i)}), (\boldsymbol{x}_t^{(i)}, \boldsymbol{y}_t^{(i)}), f_\theta, \ell\}$ as in Eqn. (1).

MAML fine-tunes the meta model f_θ to a particular task $\mathcal{T}_i$ by gradient descent:

$$\theta_i' \leftarrow \theta - \alpha \nabla \mathcal{L}_{\mathcal{T}_i}(f_\theta) \tag{4}$$

where $\mathcal{L}_{\mathcal{T}_i}(f_\theta) = \frac{1}{Nk} \sum_{j=1}^{Nk} \ell(f_\theta(x_j^{(i)}), y_j^{(i)})$ is the task-related training loss and α is a universal learning rate.

By treating each task as a training example, MAML optimizes meta model parameter θ such that the total loss for the task-wise fine-tuned parameter θ_i' over testing samples $(\boldsymbol{x}_t^{(i)}, \boldsymbol{y}_t^{(i)})$ can be minimized:

$$\min_\theta \sum_{i=1}^{m} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'}) = \sum_{i=1}^{m} \mathcal{L}_{\mathcal{T}_i}(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)}),$$

where $\mathcal{L}_{\mathcal{T}_i}(f_{\theta'}) = \ell(f_{\theta'}(\boldsymbol{x}_t^{(i)}), \boldsymbol{y}_t^{(i)})$, *i.e.*, classification loss on the reserved testing samples. The meta parameter θ is then updated by gradient descent $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{i=1}^{m} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$. The trained meta model f_θ can be applied directly to a new similar N-category k-shot learning task through gradient descent fine-tuning in Eqn. (4) and performs remarkably well.

MAML inspires us in two aspects for solving few-shot learning problems. First, instead of training a single model on all available training data at once (which is a common practice in most few-shot learning methods (Fei-Fei et al., 2006; Hariharan & Girshick, 2016; Lake et al., 2013)), we should construct learning tasks exactly matching the testing case for model training, which is a more suitable learning scheme for obtaining strong generalization ability from few examples. Second, compared with optimizing the classification accuracy, optimizing the model adaptive capability to new tasks better fits the nature of few-shot learning.

Although MAML provides promising solutions to few-shot learning, its performance highly depends on the similarity of training and testing tasks. It cannot handle discrepancies among tasks—in particular the domain shift between source and target data we aim to address—and suffers performance decline.

3.3 Transferable Meta Learning Algorithm

Our proposed Transferable Meta Learning (TML) algorithm solves cross-domain meta learning problems by learning a meta model from the source data D_S along with a few *unlabeled* target data, which can fast adapt to various few-shot classification tasks in target data D_T. Beyond existing meta-learning algorithms (like MAML), TML entails the meta model with two-fold fast adaptation capability. First, the model can learn from few training examples fast through simple fine-tuning, solving the few-shot learning tasks. Second, the model can adapt to tasks in different domains, addressing the task heterogeneity issues caused by domain shift.

TML is developed following a simple intuition: the loss function computed in the source domain is expected to be a good indicator of the target loss when both tasks are similar. The main idea of TML is to learn

a meta model that is capable of adapting to new tasks fast and meanwhile learning domain-invariant representations such that source tasks can provide useful meta-knowledge for training models in target domains. To this end, we develop a new learning scheme and propose a novel meta model architecture. The meta model $f_{\varphi,\theta}$ learned by TML includes two components, a domain-invariant representation learner f_φ parameterized by φ and a meta-classifier f_θ with parameters θ, as illustrated in Fig. 2. TML optimizes these two components jointly such that the domain divergence can be reduced in a way favorable for few-shot learning and facilitate cross-domain meta-knowledge transfer.

TML Learning Scheme We apply TML to train a meta model on a collection of source training tasks, with a new learning scheme suiting cross-domain meta learning. For notational simplicity, we use $(\boldsymbol{x}_S, \boldsymbol{y}_S)$ and $\boldsymbol{x}_T$ to collectively denote source data and unlabeled target data respectively, and let $(\boldsymbol{x}_{t,S}, \boldsymbol{y}_{t,S})$ denote another source sample reserved for evaluating the loss in Eqn. (2). Then, we define a cross-domain few-shot learning task for TML as

$$\mathcal{T}_i \triangleq \{(\boldsymbol{x}_S^{(i)}, \boldsymbol{y}_S^{(i)}), (\boldsymbol{x}_{t,S}^{(i)}, \boldsymbol{y}_{t,S}^{(i)}), \boldsymbol{x}_T^{(i)}, f_{\varphi,\theta}, \ell\}, \quad (5)$$

which includes two model training steps. First, fine-tune the meta-classifier θ and representation learner φ with few-shot source training samples $(\boldsymbol{x}_S^{(i)}, \boldsymbol{y}_S^{(i)})$ by gradient descent:

$$\varphi_i', \theta_i' \leftarrow (\varphi, \theta) - \alpha \nabla_{\varphi,\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\varphi,\theta}), \quad (6)$$

and evaluate classification loss on $(\boldsymbol{x}_{t,S}^{(i)}, \boldsymbol{y}_{t,S}^{(i)})$ based on ℓ as in Eqn. (2). Second, optimize representation learner φ to minimize distribution divergence (see below) between source data $\boldsymbol{x}_S^{(i)}$ and target data $\boldsymbol{x}_T^{(i)}$. This new task formulation distinguishes TML from existing meta learning algorithms. TML explicitly meta-learns both few-shot classification and task adaptation.

When applying the meta model $f_{\varphi,\theta}$ to few-shot learning tasks in target domain $\mathcal{D}_T$, we apply gradient descent to fine-tune the model parameters φ and θ over the few labeled target data, following Eqn. (6).

TML Algorithm We explain how TML trains a meta model on the training tasks $\{\mathcal{T}_i\}$ constructed as above, which alternates between two optimization sub-procedures, as illustrated in Fig. 2.

The first is *meta learning* step, where TML tries to learn a domain-specific meta-classifier θ and representation learner φ such that fine-tuning over them can minimize

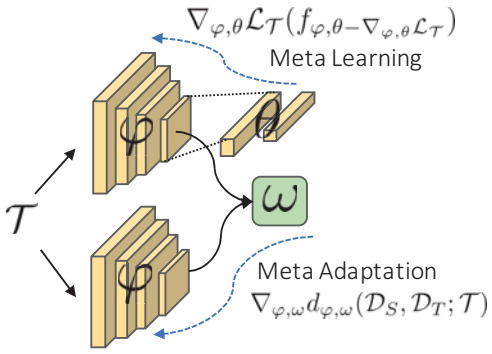

Figure 2: TML for meta model training. TML performs meta learning and task adaptation jointly to optimize the meta model (consisting of representation learner φ and classifier θ) and the discriminative model ω.

the loss over source test data:

$$\min_{\varphi,\theta} \sum_{\mathcal{T}_i \in \mathcal{T}} \mathcal{L}_{\mathcal{T}_i}(f_{\varphi_i',\theta_i'}) = \sum_{\mathcal{T}_i \in \mathcal{T}} \mathcal{L}_{\mathcal{T}_i}(f_{(\varphi - \alpha\nabla_\varphi \mathcal{L}_{\mathcal{T}_i}(f_{\varphi,\theta})),(\theta - \alpha\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_{\varphi,\theta}))})$$

where the inner loss $\mathcal{L}_{\mathcal{T}_i}(f_{\varphi,\theta})$ is the total cross-entropy loss over the training samples $(\boldsymbol{x}_S^{(i)}, \boldsymbol{y}_S^{(i)})$ in task i:

$$\mathcal{L}_{\mathcal{T}_i}(f_{\varphi,\theta}) = \sum_{(x_j,y_j)\in(\boldsymbol{x}_S^{(i)},\boldsymbol{y}_S^{(i)})} y_j \log f_{\varphi,\theta}(x_j) + (1-y_j)\log(1-f_{\varphi,\theta}(x_j)). \quad (7)$$

The outer meta loss $\mathcal{L}_{\mathcal{T}_i}(f_{\varphi_i',\theta_i'})$ is the cross-entropy loss defined on the task-specific testing samples $(\boldsymbol{x}_{t,S}^{(i)}, \boldsymbol{y}_{t,S}^{(i)}) \in \mathcal{T}_i$ for the fine-tuned model after one gradient descent step $f_{\varphi',\theta'}$:

$$\mathcal{L}_{\mathcal{T}_i}(f_{\varphi_i',\theta_i'}) = \sum_{(x_{t,S},y_{t,S})\in\mathcal{T}_i} y_{t,S} \log f_{\varphi_i',\theta_i'}(x_{t,S}) + (1-y_{t,S})\log(1-f_{\varphi_i',\theta_i'}(x_{t,S})).$$

The involved meta model parameters are updated by gradient descent:

$$\varphi \leftarrow \varphi - \beta\nabla_\varphi \sum_{\mathcal{T}_i\in\mathcal{T}} \mathcal{L}_{\mathcal{T}_i}(f_{\varphi_i',\theta_i'}),$$
$$\theta \leftarrow \theta - \beta\nabla_\theta \sum_{\mathcal{T}_i\in\mathcal{T}} \mathcal{L}_{\mathcal{T}_i}(f_{\varphi_i',\theta_i'}).$$

This meta learning step is similar to MAML but it decouples the representation learner φ and classifier θ for developing the following meta adaptation learning.

The second step in TML is *meta adaptation* with a target to make the meta model fast adaptable to the target

domain $\mathcal{D}_T$ which is different from the source $\mathcal{D}_S$ used for extensive meta model training. In particular, TML trains the representation learner f_φ in this step such that it can be adapted through fine-tuning to minimize the distribution divergence between $\mathcal{D}_T$ and $\mathcal{D}_S$, to alleviate domain-shift issues when applying the meta-classier f_θ. Specifically, we use a domain adversarial discriminative loss to measure divergence between domains $\mathcal{D}_S$ and $\mathcal{D}_T$, in the feature space produced by the representation learner f_φ, inspired by (Ganin et al., 2016).

Formally, we use $\mathcal{U}_S$ to denote the source feature space derived by passing source data $\boldsymbol{x}_S$ through the representation learner f_φ. The target feature space $\mathcal{U}_T \leftarrow \boldsymbol{x}_T : f_\varphi$ is derived similarly. A domain discriminator D_ω parameterized by ω is trained on tasks $\mathcal{T}_i$ to distinguish whether a sample x is from $\mathcal{U}_S$ or $\mathcal{U}_T$:

$$\omega = \arg\max \log D_\omega(f_\varphi(\boldsymbol{x}_S)) + \log(1 - D_\omega(f_\varphi(\boldsymbol{x}_T)),$$

where we give label 1 to the data from source domain and 0 otherwise. We use the negative cross-entropy loss of D_ω as a measure over the domain divergence—a larger discriminative loss means the samples are indistinguishable w.r.t. domain shift, indicating a small domain divergence. The domain divergence is calculated as below, dependent on the meta model parameter φ and discriminator ω:

$$\begin{aligned} d_{\varphi,\omega}(\mathcal{D}_S, \mathcal{D}_T) := &- \mathbb{E}_{x_S \in \mathcal{D}_S} \left[\log D_\omega(f_\varphi(x_S))\right] \\ &- \mathbb{E}_{x_T \in \mathcal{D}_T} \left[\log(1 - D_\omega(f_\varphi(x_T)))\right]. \end{aligned}$$

In meta adaptation, TML learns φ and ω jointly to minimize the domain divergence derived from samples provided in training tasks $\mathcal{T}_i$, *i.e.*,

$$\begin{aligned} d_{\varphi,\omega}(\mathcal{D}_S, \mathcal{D}_T; \mathcal{T}_i) := &- \sum_{x_S \in \boldsymbol{x}_S^{(i)}} \left[\log D_\omega(f_\varphi(x_S))\right] \\ &- \sum_{x_T \in \boldsymbol{x}_T^{(i)}} \left[\log(1 - D_\omega(f_\varphi(x_T)))\right]. \end{aligned}$$

In all, the learning objective for TML is

$$\begin{aligned} &\min_{\theta,\varphi} \max_{\omega} \sum_{\mathcal{T}_i \in \mathcal{T}} \mathcal{L}_{\mathcal{T}_i}(f_{\varphi_i', \theta_i'}) - d_{\varphi,\omega}(\mathcal{D}_S, \mathcal{D}_T; \mathcal{T}_i), \\ &\text{s.t.} \quad \varphi_i', \theta_i' \leftarrow (\varphi, \theta) - \alpha \nabla_{\varphi,\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\varphi,\theta}). \end{aligned}$$

TML updates meta model parameters φ, θ and discriminator ω by gradient descent. Details for our proposed TML algorithm are summarized in Alg. 1. The output meta model has following attractive advantages. First, it is well prepared for fast adapting to new tasks and domains through gradient based fine-tuning. Second, its representation learner maps the heterogeneous-domain data into a common space with minimized domain divergence such that model can effectively transfer meta-knowledge for few-shot learning across domains. In the experiments, we also verify that TML is superior to the approach that performs domain adaptation and meta learning separately.

Algorithm 1: Transferable Meta Learning

Input: Source domain data D_S, target domain data D_T, task set $\mathcal{T}$, learning rates α, β, γ, max iteration I

Output: Representation learner φ, classifier θ, domain discriminator ω

Randomly initialize θ, φ, ω

for $i = 1, \ldots, I$ **do**

- Sample task $\mathcal{T}_i \in \mathcal{T}$.
- Estimate $\nabla \mathcal{L}_{\mathcal{T}_i}(f_{\varphi,\theta})$ using task provided training samples $(\boldsymbol{x}, \boldsymbol{y})$ based on Eqn. (7)
- Fine-tune the parameters φ, θ as Eqn. (6): $(\varphi_i', \theta_i') = (\varphi, \theta) - \alpha \nabla_{(\varphi,\theta)} \mathcal{L}_{\mathcal{T}_i}(f_{\varphi,\theta})$
- Estimate meta learning gradient w.r.t. (φ, θ) on the testing examples in task $\mathcal{T}_i$ by: $(\Delta_\varphi^{cls}, \Delta_\theta^{cls}) = \nabla_{(\varphi,\theta)} \sum_{\mathcal{T}_i \in \mathcal{T}} \mathcal{L}_{\mathcal{T}_i^S}(f_{\varphi_i', \theta_i'})$
- Sample source data $\boldsymbol{x}_S \in D_S$, target data $\boldsymbol{x}_T \in D_T$
- Estimate meta adaptation gradient w.r.t. φ based on $(\boldsymbol{x}_S, \boldsymbol{x}_T)$ by: $\Delta_\varphi^{adpt} = \nabla_\varphi d_{\varphi,\omega}(\mathcal{D}_S, \mathcal{D}_T)$
- Update model parameters:
  - $\varphi \leftarrow \varphi - \beta \Delta_\varphi^{cls} - \gamma \Delta_\varphi^{adpt}$
  - $\theta \leftarrow \theta - \beta \Delta_\theta^{cls}$
  - $\omega \leftarrow \omega - \gamma \nabla_\omega d_{\varphi,\omega}(\mathcal{D}_S, \mathcal{D}_T)$

end

4 Experiments

Datasets We first conduct experiments on learning a meta few-shot classification model with TML across three digits datasets, *i.e.*, MNIST (LeCun et al., 1998), USPS (Le Cun et al., 1989) and SVHN (Netzer et al., 2011). Each dataset contains 10 categories of digit images with varying characteristics. Then we evaluate TML on the office dataset (Saenko et al., 2010), which is a more challenging benchmark with more complex image contents and more significant domain shift. The dataset contains 31 classes of office supplies from three distinct domains, *i.e.*, Amazon, DSLR and Webcam.

Experiment Settings All experiments follow the standard N-way K-shot protocol (Vinyals et al., 2016) for few-shot learning, which is widely used for evaluating meta learning algorithms. Under this protocol, samples from one dataset are split and re-organized into multiple tasks. Each task provides N selected classes with K labeled training instances per class. Each task requires training a few-shot learning model on the provided la-

beled samples. See Sec. 3.1 for the formal definition of the task. We form the learning tasks for TML in the way described in Sec. 3.3. Note the labels for the N selected classes are randomly assigned under the few-shot learning protocol. The purpose is to evaluate whether the model indeed gains the capability of learning to recognize from few examples, instead of memorizing training examples from all the tasks. In our implementation, both the N classes and K samples are randomly selected from the whole dataset. Performance of a few-shot learning model is measured by the classification accuracy on another $K \times N$ unseen samples.

As we are interested in the cross-domain setting, in the experiments we train a model with access to the source data but evaluate its performance on tasks from the specified target domain. For instance, under the cross-domain setting of "MNIST $\Rightarrow$ USPS", we train a model on MNIST data and evaluate its performance in USPS few-shot learning tasks.

Baselines Since the problem of cross-domain few-shot classification is new, few valid methods are available for solving it. Here we compare TML with following strong baselines, which leverage the state-of-the-art meta learning algorithms and domain adaptation methods, for obtaining the meta model. 1) Train a standard supervised-learning classifier on the source dataset, and fine-tune it for each specific target task. Comparing with this baseline aims to verify effectiveness of TML in training a meta few-shot learning model with strong generalization ability from few samples, compared with standard supervised learning methods. 2) The state-of-the-art meta learning algorithm, MAML (Finn et al., 2017). In particular, we use MAML to train the meta model in the source domain following the few-shot learning protocols and directly apply MAML to solve target tasks. We aim to demonstrate the advantage of TML over MAML in handling cross-domain few-shot learning problems. 3) The "oracle" MAML. It is trained using the full target datasets and provides performance upper bound for all the cross-domain trained models. 4) MAML+ADDA. Concretely, apply state-of-the-art domain adaptation method ADDA (Tzeng et al., 2017) to align target domain samples with the source domain at first, and then apply MAML on the adapted sample representations. For this baseline, the domain adaptation is blind and unaware of few-shot learning tasks. Comparing TML with it verifies the benefits of end-to-end meta-adaptation and meta-learning in TML.

Implementation Details The meta model in TML consists of two components, a meta representation learner and a meta classifier. The former contains four cascade convolutional units, while the latter is built with a linear transformation layer followed by a softmax layer, following similar architectures in (Finn et al., 2017; Vinyals et al., 2016). The architecture of the convolutional units varies along with the number of input image channels. For gray-scale images (*e.g.*, digit images), each convolutional unit is composed of 1) 3×3 2D convolution with 64 filters and stride 2×2, 2) batch normalization (Ioffe & Szegedy, 2015), and 3) ReLU nonlinear activation function. After the convolutions, a mean pooling layer is used to transform multiple 2D feature maps into a linear feature vector. For color images (*e.g.*, office images), the convolutional unit changes to 1) 3×3 2-D convolution with 32 filters and stride 1, 2) batch normalization, 3) 2×2 max pooling layer, and 4) ReLU nonlinearity. Then the feature maps are simply flatten to produce a feature vector for classification. The domain discriminator consists of 3 fully connected layers. Each of the two hidden layers has 500 neurons and is followed by a ReLU activation function. One single unit in the output layer is used to indicate the input is from the source or target domain.

To train all network models, we adopt Adam (Kingma & Ba, 2014) as the optimizer. The meta learning rate β is set as 0.001, while the adaptation learning rate γ is set as 2×10^{-4}. The update (or fine-tuning) learning rate α is fixed as 0.4 for training on gray images and 0.01 for color images. The task batch size is 32 for gray images and 4 for color image due to GPU memory limitation. All models are trained on a single GeForce GTX TITAN Black GPU with 12G memory.

4.1 Results on Digit Datasets

We present results on the digit datasets with multiple cross-domain directions. We first convert all images to gray scale and resize them to 28×28. When building learning tasks, we rotate images class-wise by 90° randomly for data augmentation (Santoro et al., 2016a).

We test our TML in following few-shot settings: 5-way 1-shot and 5-way 5-shot, and four cross-domain directions: MNIST $\Rightarrow$ USPS, MNIST $\Rightarrow$ SVHN, USPS $\Rightarrow$ MNIST and SVHN $\Rightarrow$ USPS. For the fine-tuning baseline, we first train three classifiers of the same architecture as our model on the three full digit datasets (using the training set) individually. Then the learned classifier on source domain is fine-tuned on the other two target domains under the few-shot setting, *i.e.*, fine-tuning the model on a few training samples and evaluating it on the testing samples for each task individually. For effectively preventing over-fitting, we carefully select the fine-tune learning rate which is set as 2×10^{-4}.

Table 1 reports the few-shot classification accuracy averaged over 500 randomly sampled tasks. One can observe that MAML, MMAL+ADDA and TML all surpass the fine-tune baseline by a large margin in all settings,

Table 1: Few-shot classification results for digit images. The left-most column shows the cross-domain direction, where **M**, **U**, **S** denotes MNIST, USPS, SVHN respectively. The results are reported in terms of classification accuracy (%) averaged over 500 tasks. The gray number in parentheses for MAML is the averaged accuracy obtained by training MAML on the full target datasets. "M + A" denotes the MAML+ADDA baseline.

Direction	5-way 1-shot				5-way 5-shot			
	Fine-tune	MAML	M + A	TML	Fine-tune	MAML	M + A	TML
M ⇒ U	39.12	86.33 (97.97)	86.83	**91.70**	63.42	93.43 (97.95)	93.44	**94.43**
M ⇒ S	21.08	22.30 (83.00)	24.03	**29.60**	24.56	28.82 (89.70)	29.43	**29.68**
U ⇒ M	37.48	83.30 (99.30)	81.80	**88.90**	52.99	89.42 (99.59)	**90.03**	90.01
S ⇒ U	23.36	**84.60** (97.97)	82.70	82.67	61.21	87.23 (97.95)	87.86	**89.23**
Direction	10-way 1-shot				10-way 5-shot			
	Fine-tune	MAML	M + A	TML	Fine-tune	MAML	M + A	TML
M ⇒ U	24.50	74.15 (94.60)	75.05	**80.45**	49.50	81.86 (95.85)	82.54	**86.43**
M ⇒ S	11.14	12.52 (67.57)	13.13	**13.55**	13.09	15.36 (84.41)	14.22	**15.44**
U ⇒ M	17.64	53.98 (98.68)	59.05	**65.58**	31.49	73.09 (98.98)	72.96	**75.99**
S ⇒ U	21.46	54.88 (94.60)	59.30	**68.80**	43.23	78.71 (95.85)	78.96	**80.04**

Table 2: Few-shot classification accuracy (in %) of TML in source domain of the digit datasets, averaged over 500 randomly sampled tasks.

	5-way 1-shot	5-way 5-shot
M	99.47	99.42
U	97.40	97.05
S	83.87	89.82

proving the benefits of considering the nature of few-shot learning in model training. More importantly, TML outperforms MAML for almost all settings, by a margin up to 8%. This shows effectiveness of TML in solving domain-shift issues for few-shot learning tasks. The second-best baseline ADDA+MAML also tries to solve domain-shift explicitly by applying ADDA to align domains at first. However its performance is inferior to TML as it conducts domain adaptation blindly which may harm the few-shot learning performance. In contrast, our proposed TML performs meta-adaptation and meta-learning jointly, providing fast adaptation abilities to both new tasks and new domains. Thus it improves ADDA+MAML by up to 9.5%. The superiority of TML over ADDA+MAML becomes more significant when training samples are very limited.

For the U ⇒ M (5-way, 5-shot) setting, TML performs comparably well as ADDA+MAML, where the target domain data are sufficient for ADDA to achieve good domain adaptation. For the S ⇒ U (5-way, 1-shot) setting, MAML performs slightly better than TML. This is because the domain divergence between S and U is large and target data from a single task are limited for TML to perform meta-adaptation sufficiently well.

For better understanding the domain shift challenge to few-shot learning, we also evaluate the oracle MAML

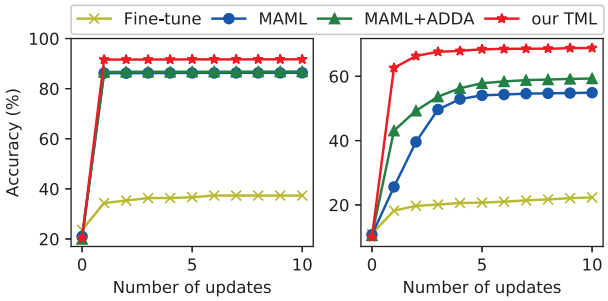

Figure 3: Adaptation speed comparison. Left: MNIST ⇒ USPS, 5-way 1-shot. Right: SVHN ⇒ USPS, 10-way 1-shot. The fine-tuning baseline updates for 300 steps in total, *i.e.*, one step amounts to 30 actual steps.

baseline which has full access to samples in the target domain. The results in Table 1 show that domain shift brings a significant performance drop to MAML, demonstrating the necessity of addressing domain-shift in few-shot learning. TML can reduce the performance gap moderately. TML is effective at minimizing the domain divergence without harming performance in source domain. To show this, we also evaluate its few-shot classification performance on the source domain. The results in Table 2 demonstrate that TML performs as well as MAML in the source domain, proving TML can learn domain-invariant representations benefiting applications in both source and target domains.

Moreover, fast adaptation is important in practical applications. Therefore, we evaluate adaptation speed (in terms of adaptation steps) of different methods. Given a new task from the target domain, each model is updated for several steps (*e.g.*, 10) with gradient descent using the task-provided few-shot training data. We plot the few-shot classification performance against model updating steps for the naive fine-tuning model, MAML,

Table 3: Few-shot classification results for office images. The left-most column shows the cross-domain direction, where **A**, **D**, **W** denote Amazon, DSLR, Webcam respectively. The results are reported in terms of classification accuracy (in %), averaged over 500 tasks. "M + A" denotes the MAML+ADDA baseline.

Direction	5-way 1-shot				5-way 5-shot			
	Fine-tune	MAML	M + A	TML	Fine-tune	MAML	M + A	TML
A ⇒ D	42.44	45.90	49.43	**54.50**	74.12	71.74	71.63	**80.72**
A ⇒ W	41.67	46.43	48.50	**53.83**	69.71	70.70	69.70	**78.59**
D ⇒ W	46.48	72.77	72.20	**76.70**	74.02	77.51	78.78	**88.97**
W ⇒ D	49.28	80.20	80.93	**82.83**	78.91	90.48	89.55	**91.25**
Direction	10-way 1-shot				10-way 5-shot			
	Fine-tune	MAML	M + A	TML	Fine-tune	MAML	M + A	TML
A ⇒ D	32.62	32.90	35.05	**41.92**	64.37	60.20	60.04	**71.47**
A ⇒ W	31.47	32.72	35.02	**38.80**	60.39	60.13	61.36	**66.22**
D ⇒ W	34.78	51.43	50.33	**58.67**	65.01	80.08	80.22	**82.08**
W ⇒ D	38.66	58.58	57.07	**60.92**	69.09	81.02	81.17	**83.88**

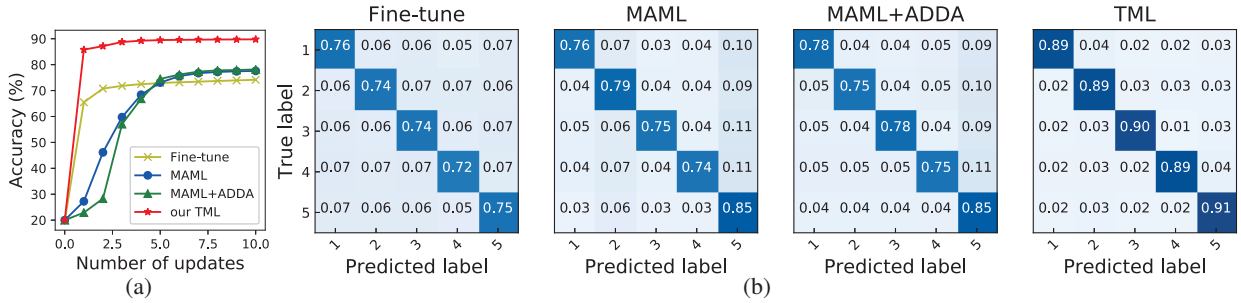

Figure 4: (a) Adaptation speed comparison on office datasets (D ⇒ W, 5-way 5-shot). (b) Corresponding few-shot classification confusion matrices on task-specific testing set.

MAML+ADDA and TML in Fig. 3. TML presents much faster adaptation than others and achieves the best performance. Notably, TML only needs one step adaptation to achieve better performance than all the baselines.

4.2 Results on Office Datasets

We then evaluate TML on the more challenging office dataset for four few-shot settings and three cross-domain directions: Amazon ⇒ DSLR, Amazon ⇒ Webcam and Webcam ⇒ DSLR. Since Amazon provides sufficient training examples, we always take it as source domain. The experimental results are shown in Table 1. Similar to the digit images, TML brings improvement up to 10.72% over MAML and MAML+ADDA, and performs significantly better than the fine-tuning baseline.

We also analyze the adaptation speed of different approaches to multiple 5-way 5-shot learning tasks from Dslr to Webcam domains, which is visualized in Fig. 4a. TML adapts significantly faster than all the baselines, demonstrating the meta-adaptation is effective for augmenting model adaptation ability. For understanding few-shot classification performance more transparently, we also plot classification confusion matrix for all the approaches in Fig. 4b. TML provides more accurate classification for all the 5 categories than baselines, showing its effectiveness in overcoming challenges from both domain-shift and limited training examples. In contrast, MAML+ADDA degrades the performance of MAML for the second category, demonstrating blind domain adaptation may confuse some categories and harm the few-shot learning performance.

5 Conclusion

This work introduced the new cross-domain meta learning problems challenged by insufficiency of training examples and varying characteristics of tasks. We developed the first transferable meta learning (TML) algorithm which substantially extends existing meta learning algorithms to solve new tasks in different domains and relieve the issues brought by insufficient training tasks.

Acknowlegement

This work was partially supported by NUS startup R-263-000-C08-133,MOE Tier-I R-263-000-C21-112, NUS IDS R-263-000-C67-646, ECRA R-263-000-C87-133 and MOE Tier-II R-263-000-D17-112.

References

Andrychowicz, Marcin, Denil, Misha, Gomez, Sergio, Hoffman, Matthew W, Pfau, David, Schaul, Tom, and de Freitas, Nando. Learning to learn by gradient descent by gradient descent. In *NIPS*, 2016.

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

Finn, Chelsea, Abbeel, Pieter, and Levine, Sergey. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

Ganin, Yaroslav, Ustinova, Evgeniya, Ajakan, Hana, Germain, Pascal, Larochelle, Hugo, Laviolette, François, Marchand, Mario, and Lempitsky, Victor. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *NIPS*, 2014.

Graves, Alex, Wayne, Greg, and Danihelka, Ivo. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Hariharan, Bharath and Girshick, Ross. Low-shot visual recognition by shrinking and hallucinating features. *arXiv preprint arXiv:1606.02819*, 2016.

Hoffman, Judy, Tzeng, Eric, Donahue, Jeff, Jia, Yangqing, Saenko, Kate, and Darrell, Trevor. One-shot adaptation of supervised deep convolutional models. *arXiv preprint arXiv:1312.6204*, 2013.

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Koch, Gregory, Zemel, Richard, and Salakhutdinov, Ruslan. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

Lake, Brenden M, Salakhutdinov, Ruslan R, and Tenenbaum, Josh. One-shot learning by inverting a compositional causal process. In *NIPS*, 2013.

Le Cun, Yann, Jackel, LD, Boser, B, Denker, JS, Graf, HP, Guyon, I, Henderson, D, Howard, RE, and Hubbard, W. Handwritten digit recognition: Applications of neural network chips and automatic learning. *IEEE Communications Magazine*, 27(11):41–46, 1989.

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

Liu, Ming-Yu and Tuzel, Oncel. Coupled generative adversarial networks. In *NIPS*, 2016.

Long, Mingsheng, Cao, Yue, Wang, Jianmin, and Jordan, Michael I. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

Luo, Zelun, Zou, Yuliang, Hoffman, Judy, and Fei-Fei, Li F. Label efficient learning of transferable representations acrosss domains and tasks. In *NIPS*, 2017.

Mishra, Nikhil, Rohaninejad, Mostafa, Chen, Xi, and Abbeel, Pieter. Meta-learning with temporal convolutions. *arXiv preprint arXiv:1707.03141*, 2017.

Motiian, Saeid, Jones, Quinn, Iranmanesh, Seyed, and Doretto, Gianfranco. Few-shot adversarial domain adaptation. In *NIPS*, 2017a.

Motiian, Saeid, Piccirilli, Marco, Adjeroh, Donald A, and Doretto, Gianfranco. Unified deep supervised domain adaptation and generalization. In *ICCV*, 2017b.

Munkhdalai, Tsendsuren and Yu, Hong. Meta networks. *arXiv preprint arXiv:1703.00837*, 2017.

Netzer, Yuval, Wang, Tao, Coates, Adam, Bissacco, Alessandro, Wu, Bo, and Ng, Andrew Y. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 5, 2011.

Ravi, Sachin and Larochelle, Hugo. Optimization as a model for few-shot learning. 2016.

Saenko, Kate, Kulis, Brian, Fritz, Mario, and Darrell, Trevor. Adapting visual category models to new domains. *ECCV*, 2010.

Santoro, Adam, Bartunov, Sergey, Botvinick, Matthew, Wierstra, Daan, and Lillicrap, Timothy. Meta-learning with memory-augmented neural networks. In *ICML*, 2016a.

Santoro, Adam, Bartunov, Sergey, Botvinick, Matthew, Wierstra, Daan, and Lillicrap, Timothy. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065*, 2016b.

Snell, Jake, Swersky, Kevin, and Zemel, Richard. Prototypical networks for few-shot learning. In *NIPS*, 2017.

Sun, Baochen and Saenko, Kate. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pp. 443–450. Springer, 2016.

Tzeng, Eric, Hoffman, Judy, Zhang, Ning, Saenko, Kate, and Darrell, Trevor. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

Tzeng, Eric, Hoffman, Judy, Darrell, Trevor, and Saenko, Kate. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4068–4076, 2015.

Tzeng, Eric, Hoffman, Judy, Saenko, Kate, and Darrell, Trevor. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464*, 2017.

Vinyals, Oriol, Blundell, Charles, Lillicrap, Tim, Wierstra, Daan, et al. Matching networks for one shot learning. In *NIPS*, 2016.

Wang, Yu-Xiong and Hebert, Martial. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, 2016.