Towards Robust Model Reuse in the Presence of Latent Domains

Jie-Jing Shao[1] , **Zhanzhan Cheng**[2] , **Yu-Feng Li**[1*] and **Shiliang Pu**[2]

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2]Hikvision Research Institute, Hangzhou, China
{shaojj, liyf}@lamda.nju.edu.cn, {chengzhanzhan, pushiliang.hri}@hikvision.com,

Abstract

Model reuse tries to adapt well pre-trained models to a new target task, without access of raw data. It attracts much attention since it reduces the learning resources. Previous model reuse studies typically operate in a single-domain scenario, i.e., the target samples arise from one single domain. However, in practice the target samples often arise from multiple latent or unknown domains, e.g., the images for cars may arise from latent domains such as *photo*, *line drawing*, *cartoon*, etc. The methods based on single-domain may no longer be feasible for multiple latent domains and may sometimes even lead to performance degeneration. To address the above issue, in this paper we propose the MRL (Model Reuse for multiple Latent domains) method. Both domain characteristics and pre-trained models are considered for the exploration of instances in the target task. Theoretically, the overall considerations are packed in a bi-level optimization framework with a reliable generalization. Moreover, through an ensemble of multiple models, the model robustness is improved with a theoretical guarantee. Empirical results on diverse real-world data sets clearly validate the effectiveness of proposed algorithms.

1 Introduction

In traditional machine learning, great efforts have been devoted to collect massive labeled data [LeCun *et al.*, 2015], explore smart optimization techniques [Kingma and Ba, 2015] and large-scale computation power [Dean *et al.*, 2012] to train accurate models. Nowadays, a great deal of well-trained machine learning models have been readily available to use. However, once given a new data set, the user still has to retrain a model, which obviously causes huge waste of public model resources. Meanwhile, due to privacy and security concerns, a large number of open source models usually do not allow to access raw data. Therefore, it is highly desirable to study *model reuse* [Zhou, 2016] which is able to reduce the learning resources for a new target task using pre-trained models without the access of raw data. Model reuse has attracted much attention and many algorithms have been developed recently, e.g., [Yang *et al.*, 2007; Ye *et al.*, 2018; Shi and Li, 2019; Li *et al.*, 2021].

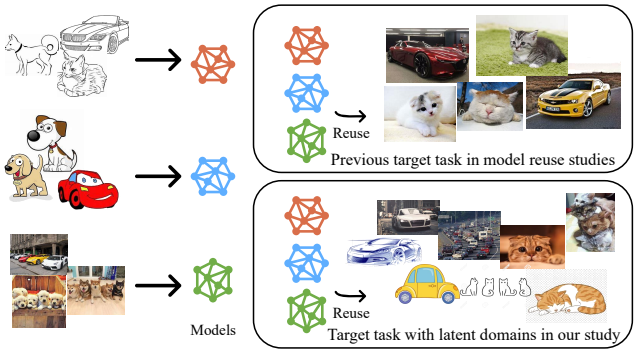


Figure 1: Comparison to previous model reuse studies. They assume the data in the target task belong to a single domain. In practice, the target task may contain several latent domains, such as the photos, line drawings, and cartoons returned by web image search for 'car' and 'cat'. What is more difficult is one usually only gets the category label (car or cat), rather than the domain labels (photo or cartoon).

Previous model reuse studies typically assume that the instances in the target task are from one single domain [Yang *et al.*, 2007; Ye *et al.*, 2018; Shi and Li, 2019; Li *et al.*, 2021]. In many real situations, however, such an assumption is hard to hold since the target task often consists of multiple latent (unknown) domains [Mancini *et al.*, 2019]. For example, images found on the web are often a collection of many hidden domains. As shown in Figure 1, images searched from web for 'car' or 'cat' consist of multiple latent domains, such as *photos*, *group shots*, *line drawings*, *cartoons*, etc, where the domain labels are unknown beforehand. Similar situations also arise in speech recognition where the samples of the target application are often from a mixture of multiple groups of speakers (domains) [Liao, 2013]. In face recognition, the target application needs to tackle samples from multiple latent domains such as front, left and right pose [Sim *et al.*, 2002]. In addition, it is often available to have the category label (such as 'car'), while difficult to collect the domain labels (such as 'photo', 'cartoon') [Mancini *et al.*, 2019].

*This research was supported by the National Key R&D Program of China (2017YFB1001903), the NSFC (61772262) and Hikvision Cooperation Fund. Y.-F. Li is the corresponding author.

It is obvious that such setting is different to the single-domain scenario in previous model reuse studies. The model reuse methods based on single-domain scenario may no longer be feasible for multiple latent domains and may sometimes even lead to performance degeneration.

To address the above issue, in this paper, a novel method MRL (Model Reuse for Latent domains) is proposed. Our basic assumption is that the instance diversity caused by domain characteristics is helpful for the exploitation of pre-trained models. Moreover, a flexible ensemble usually performs more robust than a single model. Specifically, we first construct attention transfer based on smoothness assumption [Zhou and Belkin, 2014], i.e., similar instances should have similar concept compositions within their latent domains, which motivates a similar exploitation of pre-trained models. We put it in a knowledge distillation manner and then implement knowledge transfer via a black-box prediction from pre-trained models. To achieve a reliable generalization, motivated by safe weakly supervised learning [Guo *et al.*, 2020], we enforce the learned model to be also well-performed on the given labeled examples in the target task, which is cast as a bi-level optimization framework with effective solutions. Empirical studies on a number of real-world data sets show that MRL achieves a clear performance gain over multiple algorithms.

2 Related Work

Model reuse is different to Federated Learning [Yang *et al.*, 2019], a recently emerging area, which also provides data privacy protection. Federated learning trains a joint machine learning model on decentralized clients via iterative model aggregation between clients and control severs. Unlike federated learning, the goal of model reuse is not to build a joint model from multiple separated data sources, but to help new task with the use of pre-trained models.

Model reuse is also different to various multi-task learning studies [Pentina and Lampert, 2017], where multi-task learning aims to facilitate all the tasks via joint learning multiple tasks, and model reuse aims to facilitate the target task only via the use of pre-trained models.

Our work is related to various ensemble-based weakly supervised frameworks such as *semi-supervised multi-task learning* [Pentina and Lampert, 2017], *semi-supervised ensemble learning* [Bennett *et al.*, 2002], and *multi-source domain adaptation* [Hoffman *et al.*, 2018]. It is worth noting that these frameworks assume that target data are collected form a single domain or extract data from related tasks in a white-box way, which is not the case in our study.

There are a couple of studies proposed on mining from latent domains. Hoffman *et al.* [2012] proposed a constrained clustering method to discover latent domains from labeled data and build a mixture transform to implement domain adaptation. Xiong *et al.* [2014] proposed a squared-loss mutual information clustering model based on domain-specific local subspace estimation. Mancini *et al.* [2019] presented a novel deep architecture to assign a data sample to a latent domain through a side branch. There are also some studies [Mansour *et al.*, 2008; Hoffman *et al.*, 2018] proposing the distribution-weighted ensemble for target task composed with latent domains. However, they all learn in a scratch manner, without the exploration of pre-trained models.

There are some efforts on transferring a single pre-trained network to a new target task [Romero *et al.*, 2015; Jang *et al.*, 2019]. In their studies, the target samples still raise from one single domain and the pre-trained models are required to be deep models, which is clearly different to our work.

3 Proposed MRL Method

3.1 Notation and Problem Formulation

Let $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ be the feature-label space where $\mathbf{x} \in \mathbb{R}^d$ is the d-dimensional feature space and $\mathcal{Y}$ is the label space. Formally, training data set from the task consists of n labeled instances $D_l = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ and m unlabeled instances $D_u = \{\mathbf{x}_{n+1}, \cdots, \mathbf{x}_{n+m}\}$. Usually, n is too few to derive a competitive model from scratch. m is relatively large because unlabeled data is much cheaper to obtain. Note that the target samples does not have any domain label.

Suppose there are K pre-trained models $\{h_1, \cdots, h_K\}$ each corresponds to a latent domain, built on the same feature-label space $\mathcal{X} \times \mathcal{Y}$ with different sources. Formally, there is a small constant error rate $\epsilon > 0$ with a certain loss function L on their task distribution $\mathcal{D}_i$: $\forall i \in [K], \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_i}[L(h_i(\mathbf{x}), y)] \leq \epsilon$. The goal of model reuse is to learn a model $h(\mathbf{x}; \theta) : \{\mathcal{X}; \Theta\} \to \mathcal{Y}$ parameterized by $\theta \in \Theta$ from training data to minimize the generalization risk $R_T(h) = \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_T}[L(h(\mathbf{x}; \theta), y)]$ where $\mathcal{D}_T$ denotes the distribution of the target task. Generally, L is convex and bounded by a certain constant M. Without loss of generality, we assume $\mathcal{D}_T = \sum_{k=1}^{K} \lambda_k \mathcal{D}_k$ with unknown $\lambda = [\lambda_1, \ldots, \lambda_K]$ [Mansour *et al.*, 2008]According to structural risk minimization, it is formalized as following:

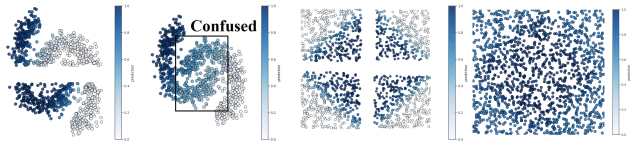
$$\min_{\theta\in\Theta} \quad \sum_{i=1}^{n} L\left(h(\mathbf{x}_i; \theta), y_i\right) + \sum_{\mathbf{x}} \Omega\left(\mathbf{x}; \theta, \{h_k\}_{k=1}^{K}\right) \quad (1)$$

where $\mathbf{x} \in \{\mathbf{x}_i\}_{i=1}^{n+m}$ and $\Omega(\mathbf{x}; \theta, \{h_k\}_{k=1}^{K})$ refers to the regularization term with the help of pre-trained models, which is the key to model reuse. Ω could be realized in different ways, like output consistency [Hinton *et al.*, 2015], representation consistency [Romero *et al.*, 2015], etc. In this paper, we choose output consistency to instantiate Ω, as it is good at coping with the probability output on label space, no matter for neural networks or trees. Specifically, $\Omega(\mathbf{x}; \theta, \{h_k\}_{k=1}^{K})$ could be rewritten as $\mathbb{D}[h(\mathbf{x}; \theta), \mathcal{R}(\mathbf{x})]$, where $\mathcal{R}$ represents a specific reuse strategy and $\mathbb{D}$ represents distance measure, like KL-Divergence, L2 distance, etc.

3.2 Consistent Reuse and Its Deficiencies

A straightforward and effective method to utilize pre-trained models is to consider a linear ensemble of existing models with weights $\mathbf{w}$ for the target task, i.e.,

$$\begin{aligned} &\mathcal{R}_{\mathbf{w}}(\{h_k\}_{k=1}^{K}, D_T) = \sum_{i=1}^{K} w_i h_i(\mathbf{x}) \\ &\text{s.t.} \quad \mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathcal{W}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_T} L\left(\sum_{i=1}^{K} w_i h_i(\mathbf{x}), y\right). \end{aligned} \quad (2)$$



(a) 2 sources (b) $\frac{1}{2}\sum_{k=1}^{2} h_k(\boldsymbol{x})$ (c) 4 sources (d) $\frac{1}{4}\sum_{k=1}^{4} h_k(\boldsymbol{x})$

Figure 2: Consistent reuse cases on two-moon data and diamond data. Each individual model from multiple sources has achieved an accurate performance on its raw task as illustrated in (a) and (c), while the joint model performs poorly on the mixed distribution.

where $\mathcal{W}$ is typically a convex set as $\{\mathbf{w} \mid w_i \geq 0, \sum_i w_i = 1\}$. The weights could be solved in various ways, such as performance error [Murugesan *et al.*, 2016; Shi and Li, 2019; Zhao *et al.*, 2020] , min-max game [Li *et al.*, 2021], empirical discrepancy [Pentina and Lampert, 2017] and maximum mean discrepancy [Duan *et al.*, 2012], etc. These methods rely on a consistent scheme that utilizes the target data in a same manner. We call them consistent reuse in this paper. Now, we present the robustness analysis of consistent reuse on $\mathcal{D}_T$.

Theorem 1. *Note that $\mathcal{D}_T = \sum_{i=1}^{K} \lambda_i \mathcal{D}_i$, the upper bound of consistent reuse $\mathcal{R}_w$ satisfies, for $\alpha > 1$,*

$$\min_{\mathbf{w}} \ \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_T} L(\mathcal{R}_w(\mathbf{x}), y) \leq \sum_{i=1}^{K} w_i [d_\alpha(\mathcal{D}_T \,||\, \mathcal{D}_i)\epsilon]^{\frac{\alpha-1}{\alpha}} M^{\frac{1}{\alpha}}$$

where $d_\alpha(\mathcal{D}\|\mathcal{D}') = e^{D_\alpha(\mathcal{D}\|\mathcal{D}')}$ denotes the exponential of the Rényi Divergence of two distributions $\mathcal{D}$ and $\mathcal{D}'$. In the worst case $\forall i \in [K], d_\alpha(\mathcal{D}_i \,||\, \mathcal{D}_T) \to \infty$, $\alpha \to 1$, the upper bound could be tailored as:

$$\min_{\mathbf{w}} \max_{\mathcal{D}_T} \ \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_T} L(\mathcal{R}_w(\mathbf{x}), y) \leq M. \tag{3}$$

Theorem 1 indicates that consistent reuse is not effective to distribution $\mathcal{D}_T$, particularly there is no guarantee for consistent reuse in the worst case. Figure 2 illustrates our observation on synthetic data. When we adapt previous accurate models from multiple sources to the target distribution $\mathcal{D}_T = \frac{1}{K}\sum_{k=1}^{K} \mathcal{D}_k$, consistent reuse $\frac{1}{K}\sum_{k=1}^{K} h_k$ is poor because of the large divergence between any individual source distribution $\mathcal{D}_i$ and the target distribution $\mathcal{D}_T$. As a result, each individual h_k performs well on one part of data distribution while performs poorly on the whole distribution.

To address such critical limitation, we propose an approach to explore the instance diversity in the target distribution associated with latent domains. In the following subsections, attention reuse, which is designed to explore instance diversity and exploit pre-trained models, is first introduced and then the optimization scheme.

3.3 Attention Reuse

As human being, one often asks different questions to different teachers/experts, e.g., ask the maths teacher for theoretical questions and the engineering teacher for practical questions. Although one single teacher may not be able to handle all the questions, one can ask the corresponding teacher for diverse queries. In this section, we simulate such a human pattern to build an attention mechanism to implement a flexible knowledge transfer.

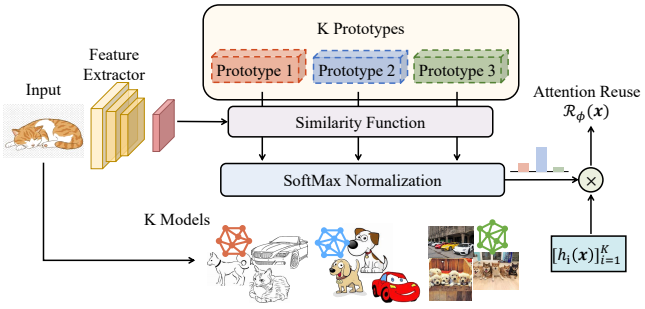


Figure 3: Attention Reuse. Through attention mechanism, we feed various instances (queries) to corresponding models (experts).

To achieve it, one first needs to properly specify how attention is defined w.r.t. given pre-trained models $\{h_k\}_{k=1}^{K}$. To that end, here we consider attention as the model-wise confidence $\mathbf{w}$ for vary queries. Based on smoothness assumption [Zhou and Belkin, 2014], that is, similar instances should have similar concept compositions within their latent domains, we further argue if model h_k could correctly predict $\mathbf{x}$, it is likely to correctly predict instances similar to $\mathbf{x}$. Thus, we construct an attention reuse $\mathcal{R}_\phi$ based on an instance-aware function $g : \mathcal{X} \to \mathcal{W}$ parameterized by ϕ.

$$\mathcal{R}_\phi(\{h_k\}_{k=1}^{K}, \mathcal{D}_T) = \sum_{i=1}^{K} g(\mathbf{x}; \phi^*)_i h_i(\mathbf{x}) \tag{4}$$

$$\text{s.t.} \quad \phi^* = \arg\min_{\phi} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_T} L\left(\sum_{i=1}^{K} g(\mathbf{x}; \phi)_i h_i(\mathbf{x}), y\right).$$

We construct a description of the marginal distribution for each model, which is named as *Prototype*. The final prediction is then computed as a weighted sum of the predictions $\{h_k(\mathbf{x})\}_{k=1}^{K}$, where the weight assigned to each prediction $h_k(\mathbf{x})$ is computed by a compatibility function of the query with its corresponding *Prototype*, as illustrated in Figure 3. In practice, we use a feature extractor $\psi : \mathcal{X} \to \mathbb{R}^p$ to obtain a low-dimensional embedding of input, and build a dot-product attention [Vaswani *et al.*, 2017] on $\psi(x)$. The K prototypes $[P_k]_{k=1}^{K} \in \mathbb{R}^{p\times k}$ and predictions $[h_k(\mathbf{x}_i)]_{k=1}^{K}$ are packed together into matrices P and H. We compute the attention output for query $\mathbf{x}$ as:

$$\mathcal{R}_\phi(\mathbf{x}) = \text{Attention}(\mathbf{x}, P, H) = \text{SoftMax}(\psi(\mathbf{x})^T P)H$$

where ψ could be instantiated as convolution networks, like a resnet-backbone pre-trained on Imagenet. Notice that the attention mechanism could be optimized via Eq. 4 when we only have category labels.

Compared to consistent reuse, attention reuse favors a better exploration of the target data with latent domains through feeding various instances to corresponding models. It owns better robustness guarantees based on [Mansour *et al.*, 2008].

Algorithm 1 The proposed MRL method

Input: Training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ and $\{\mathbf{x}_i\}_{i=n+1}^{n+m}$, predictions H, learning rate η_θ and η_ϕ, maximal iteration T.
Output: Target model $h(\theta_T)$

```
Initialize θ_0 through supervised learning.
Initialize φ_0 on labeled data via Eq. 4.
for t = 0 to T − 1 do
    Sample data from {(x_i, y_i)}_{i=1}^n and {x_i}_{i=n+1}^{n+m}
    Compute inner loss L_inner(θ_t, φ_t)
    Update θ_{t+1} = θ_t − η_θ ∇_θ L_inner(θ_t, φ_t)
    Update φ_{t+1} = φ_t − η_φ ∇_φ L_outer(θ_{t+1})
end for
return h(θ_T)
```

Theorem 2 (Robustness)**.** *Note that $\mathcal{D}_T = \sum_{i=1}^{K} \lambda_i \mathcal{D}_i$, the upper bound of attention reuse in the worst case satisfies:*

$$\min_{\phi} \max_{\mathcal{D}_T} \ \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_T} L\left(\mathcal{R}_\phi(\mathbf{x}), y\right) \leq \epsilon.$$

Compared to Theorem 1, Theorem 2 indicates that the proposed attention reuse is provably more robust than previous consistent reuse. The experimental results also confirmed our theoretical findings.

3.4 Optimization and Analysis

We denote the target model derived under Eq. 1 and ϕ as $\hat{\theta}(\phi)$. Remind that the goal of model reuse is to maximize the generalization $\mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_T}[L(h(\mathbf{x}; \hat{\theta}(\phi)), y)]$ of the target model $\hat{\theta}(\phi)$. In practice, we derive ϕ through the empirical risk on labeled data as $\sum_{i=1}^{n} L(h(\mathbf{x}_i; \hat{\theta}(\phi)), y_i)$. To simplify the notation, we denote $\hat{\theta}(\phi)$ as $\hat{\theta}$. The overall consideration is cast under a bi-level optimization scheme [Colson *et al.*, 2007].

$$\min_{\phi} \sum_{i=1}^{n} L(h(\mathbf{x}_i; \hat{\theta}), y_i) \tag{5}$$

$$\text{s.t. } \hat{\theta} = \arg\min_{\theta\in\Theta} \sum_{i=1}^{n} L\left(h(\mathbf{x}_i; \theta), y_i\right) + \sum_{i=n+1}^{n+m} \Omega_\phi\left(\mathbf{x}_i; \theta, H\right)$$

where $\Omega_\phi\left(\mathbf{x}_i; \theta, H\right) = \mathbb{D}[h(\mathbf{x}; \theta), \mathcal{R}_\phi(\mathbf{x})]$.

There are various algorithms solving the bi-level optimization in Eq. (5), such as single-level reduction, gradient descent and evolutionary algorithms [Sinha *et al.*, 2018]. We choose an efficient alternating optimization method [Jang *et al.*, 2019] to solve it. The overall optimization procedure is summarized in Algorithm 1. We first optimize $h(\theta)$ and ϕ on labeled data and employ them as the initialization, for bi-level optimization. In inner optimization, we update θ with reuse strategy $\mathcal{R}_\phi$ and predictions H from previous models. In outer optimization, we can obtain the outer objective, and update ϕ through bi-level gradient $\nabla_\phi \mathcal{L}_{outer}(\theta_{t+1})$, which can be calculated by implicit function theorem and chain rule [Shu *et al.*, 2019].

Convergence

The convergence analysis of bi-level optimization have been well-studied in previous studies [Ren *et al.*, 2018; Shu *et al.*, 2019]. Suppose the loss function is ζ-Lipschitz smooth and the gradient in inner/outer optimization is bounded by δ. Our method could achieve $E[||\nabla L_{outer}(\theta_t)||^2] \leq \epsilon_T$ in $O(1/\epsilon_T^2)$.

Generalization

We further analyze the generalization risk of MRL based on [Zhao *et al.*, 2019; Guo *et al.*, 2020] to better understand the comparison w.r.t. learning from scratch.

Theorem 3 (Generalization)**.** *Assume L is ζ-Lipschitz continuous w.r.t. ϕ. Let $\phi \in \mathbb{R}^{d'}(d' = K * p)$ be the parameters in a unit ball, and n be the labeled data size. Let $\phi^* = \arg\max_{\phi\in\mathbb{R}^{d'}} R_T(\hat{\theta}(\phi))$ be the optimal parameter in the unit ball, and $\hat{\phi}$ be the empirical optima among a candidate set $\mathcal{A}$. With probability at least $1-\delta$ we have,*

$$R_T(\hat{\theta}(\phi^*)) \leq R_T(\hat{\theta}(\hat{\phi})) + \frac{3\zeta + \sqrt{4d' \ln(n) + 8\ln(2/\delta)}}{\sqrt{n}}.$$

It is noteworthy that supervised learning which optimizes high-dimensional (d) parameters θ, achieves the optimal weight in the order $O(\sqrt{d\ln(d)\ln(n)/n})$ [Shalev-Shwartz and Ben-David, 2014]. By contrast, we learn a low-dimensional ($d' \ll d$) attention module ϕ via bi-level optimization, sharing a order $O(\sqrt{d'\ln(n)/n})$, as established in Theorem 3, favors a better order than learning from scratch.

In this work, we forge a connection between *knowledge distillation* and *teacher-student semi-supervised learning* [Qi and Luo, 2019] to build a target model, decoupled with pre-trained models, i.e., does not need to recall pre-trained models for new queries.

4 Empirical Study

4.1 Experimental Setup

To validate our method, we perform experiments on diverse tasks, including Digital Recognition, Attribute Classification and Face Recognition. All competing methods are implemented on PyTorch[1]. The output consistency $\mathbb{D}$ is instantiated with L2 distance. The hyper-parameters are adjusted by the validation set for all methods.

Learn from Scratch Besides supervised learning, we have compared three popular teacher-student SSL methods in deep learning community, i.e., **Pseudo Label** (PL) [Lee, 2013], **Temporal Ensembling** (TE) [Laine and Aila, 2017] and **Virtual Adversarial Training** (VAT) [Miyato *et al.*, 2018]. PL teaches unlabeled data via entropy regularization, TE tracks a model ensemble over time to have a better teacher model, and VAT generates adversarial noise on input to construct a robust teacher. They utilize the consistency of unlabeled data without much prior knowledge, showing promising performance.

Learn with Reuse We have also compared the effect of model reuse with the best single teacher selected from labeled data, as well as the consistent reuse. Without loss of generality, we implement black-box knowledge transfer based on distillation [Hinton *et al.*, 2015]. We denote them as **MR-BS** (Model Reuse with Best Single source) and **MR-CR** (Model Reuse with Consistent Reuse), respectively.

[1] https://pytorch.org/

	Digital Recognition							
	MNIST	SVHN	USPS	MS	MU	SU	MSU	Ave. Gain
Supervised	.837 ± .018	.660 ± .013	.794 ± .023	.726 ± .021	.857 ± .014	.691 ± .009	.745 ± .006	-
PL	.870 ± .018	.706 ± .018	.818 ± .024	.762 ± .010	.890 ± .008	.728 ± .014	.774 ± .011	.034
TE	.853 ± .017	.710 ± .017	.822 ± .032	.774 ± .006	.879 ± .015	.736 ± .010	.752 ± .011	.031
VAT	.834 ± .020	.668 ± .023	.827 ± .024	.716 ± .024	.871 ± .026	.706 ± .015	.760 ± .018	.010
MR-BS	.899 ± .008	.664 ± .017	.851 ± .017	.786 ± .003	.936 ± .001	.820 ± .003	.756 ± .007	.057
MR-CR	.877 ± .006	.681 ± .013	.865 ± .008	.746 ± .010	.877 ± .014	.719 ± .012	.758 ± .011	.030
MRL	**.971 ± .002**	**.764 ± .010**	**.898 ± .005**	**.867 ± .007**	**.956 ± .006**	**.837 ± .003**	**.867 ± .004**	**.131**

	Attribute Classification							
	Smart	Slow	Bulbous	Solitary	Nestspot	Lean	Spots	Ave. Gain
Supervised	.820 ± .010	.853 ± .021	.846 ± .011	.812 ± .012	.862 ± .009	.853 ± .009	.820 ± .021	-
PL	.855 ± .008	.886 ± .012	.865 ± .008	.858 ± .007	.879 ± .009	.859 ± .013	.847 ± .011	.026
TE	.855 ± .014	.878 ± .008	.864 ± .006	.849 ± .009	.873 ± .007	.855 ± .006	.862 ± .008	.024
VAT	.855 ± .011	.884 ± .015	.876 ± .007	.857 ± .021	.887 ± .008	.861 ± .006	.871 ± .010	.032
MR-BS	.811 ± .028	.870 ± .011	.844 ± .009	.818 ± .017	.852 ± .014	.851 ± .010	.862 ± .007	.006
MR-CR	.827 ± .013	.874 ± .011	.852 ± .017	.826 ± .016	.852 ± .013	.847 ± .007	.866 ± .006	.011
MRL	**.876 ± .016**	**.901 ± .006**	**.890 ± .011**	**.874 ± .011**	**.896 ± .009**	**.889 ± .005**	**.896 ± .008**	**.051**

Table 1: Accuracy (mean ± std) for digital recognition and attribute classification. For reuse methods, if the performance is worse than SSL, the corresponding entries are boxed. The entries with the best performance in each subtask are bolded.

4.2 Tasks on Digital Recognition

We begin our study with a generalization of digital recognition task, which consists of three digital recognition datasets: SVHN, MNIST, and USPS. Generally, we use the full recommended training sets per domain to learn the source models, and select test images from 3 datasets to construct the target tasks. For consistency, we resize these images to 28x28, and convert the images from SVHN to gray scale.

During reuse, we take 10% samples for validation and 40% samples for testing. Besides, we extract 500, 100, 100 labeled samples from SVHN, MNIST and USPS respectively, and the remaining samples as unlabeled data. Specifically, we train a CNN with 3 convolution layers and take the output of convolution as the representation (512-dim) to derive attention.

Table 1 shows the average accuracy of 5 random splits on 7 different target tasks composed with different domains. For examples, "MSU" means the target task composed with three domains, "MS" means the target task composed with domains MNIST and SVHN. It can be found that the consistent reuse method is often inferior to the direct semi-supervised learning methods, and its performance is not robust. In contrast, our method consistently outperforms semi-supervised learning methods and baselines, and shows stronger robustness. Particularly in the average, our method has a 13.1% performance gain over supervised baseline, which is clearly better than the consistency method (3.0%) and SSL (3.4%).

4.3 Tasks on Attribute Classification

Our second set of experiments is based on the Animals with Attributes 2 dataset[2], which contains 37,322 images of 50 animal classes. These animals are aligned to 85 binary attributes,

[2]https://cvml.ist.ac.at/AwA2/

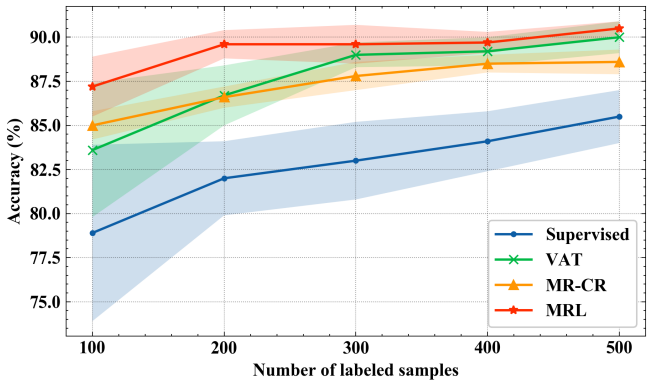


Figure 4: Results of knowledge transfer with varying numbers of labeled samples on attribute classification task.

e.g. color, habitat, via a class-attribute binary matrix, indicating whether an animal possesses each feature.

We select 4 attributes about habitat {plains, mountains, water, tree}, to simulate the different sources and construct multiple tasks of identifying whether the animal on a given image possesses a certain attribute or not. Firstly, we split 27,322 samples and divided them into these 4 sources according to the habitats. For example, *horses* belong to *plains*, *dolphins* belong to *water*, and *pandas* belong to *tree*. Then, four models were built on these sources respectively. Such a setting is consistent with data sources for different geographic locations in real-world. The target task is selected from some advanced attributes, for example, to determine whether the animal has spots, whether it is smart, etc. For the remaining 10,000 samples, we use 50% as training set, 10% examples

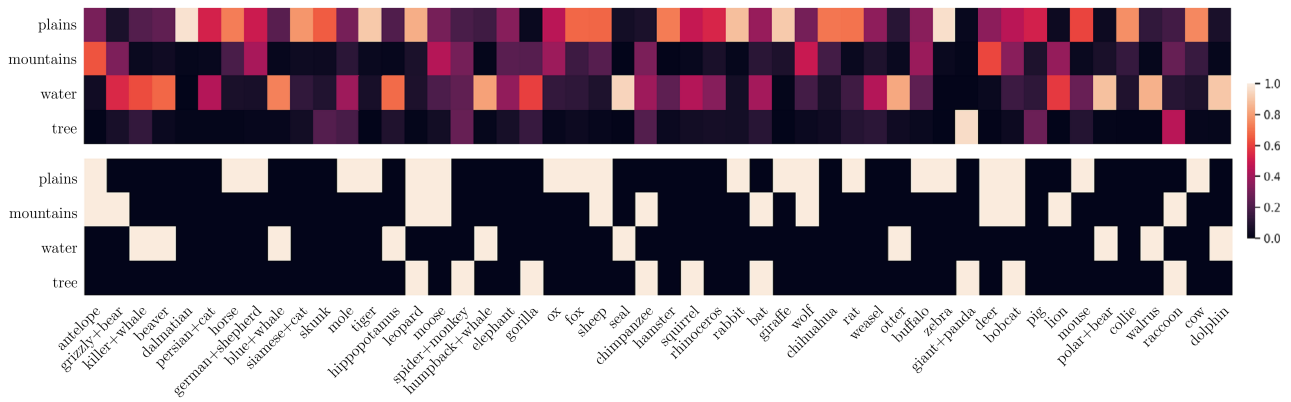


Figure 5: Visualization (best in color) of attention values on pre-trained models and corresponding habitat attributes. The lower part contains habitat information for various animals. The upper part contains the weight learned by our method on 500 labeled examples. We could find it has reasonably mined the relationship between different instances and habitats (latent domains). For example, the weights of f_{plains} and $f_{mountains}$ on antelope are larger than others, which feed such instances into corresponding models.

as validation set and take the others for testing. In addition, we construct an $\{2048, 200, 2\}$ MLP as the target model, and use the 2048-dimensional features, taken by convolution of ILSVRC pre-trained ResNet101 [Xian *et al.*, 2018], as the input of our MLP and attention module. The average results (with 200 labeled samples) under 10 times random splits are reported. From Table 1, it can be found that the consistent reuse method does not work well, i.e., it is usually inferior to the direct semi-supervised learning methods. In contrast, our method behaves much more robust, i.e., it consistently outperforms semi-supervised learning methods and baselines over all the subtasks.

A key factor of model reuse is to study the effectiveness when the labeled data is limited. In these cases, model reuse would be more desriable. We sample various labeled samples and compare the performance of Supervised, VAT, MR-CR and MRL on "spots" task. As shown in Figure 4, MRL consistent achieves better performance gain over compared methods, especially when labeled samples are few.

To further evaluate the effectiveness of attention module, we visualize the average value of similarity learned on different prototypes in the upper part of Figure 5. The lower part has shown the ground truth of habitat. We could find that our attention module could reasonablely learn the relationship between instances and habitats (latent domains).

4.4 Tasks on Face Recognition

Finally, we evaluate our method on the CMU Multi-PIE dataset [Sim *et al.*, 2002], which is a facial expression dataset. In this experiment, five domains generated from Multi-PIE (each corresponding to a distinct pose) from 68 individuals. Specifically, five subsets, i.e., PIE05 (left), PIE07 (upward), PIE09 (downward), PIE27 (front), PIE29 (right) are constructed, and the face images in each subset are taken under different illumination and expression conditions. These subsets[3] are based on SURF features and the dimension of features is 1024.

[3]https://github.com/jindongwang/transferlearning/blob/master/data/dataset.md

	1-shot	2-shot
Supervised	.632 ± .016	.815 ± .011
PL	.637 ± .013	.800 ± .010
TE	.692 ± .011	.823 ± .015
VAT	.704 ± .011	.825 ± .008
MR-BS	.657 ± .014	.812 ± .014
MR-CR	.650 ± .023	.802 ± .012
MRL	**.851 ± .009**	**.885 ± .008**

Table 2: Accuracy (mean ± std) for face recognition.

We extract 800 samples per domain to construct the target task, the remaining samples are used as the source data to build pre-trained models separately. Due to the limited number of samples, we use 680 samples (2 samples per category per domain) as testing set, and 340 samples (1 sample per category per domain) as validation set. The rest of the target data is regarded as unlabeled data pool. In addition, we use Logistic Regression as the base model, and use SURF feature as the input of attention module. The results on one-shot (1 sample per category per domain) and two-shot cases in Table 2 further demonstrate the robustness of our proposal.

5 Conclusion

In this paper, we study the *model reuse for latent domains* problem where the target data are composed with latent or unknown domains. This is a new kind of model reuse which to the best of our knowledge, has not been thoroughly studied. In this paper we propose a novel MRL method. Both domain characteristics and pre-trained models are considered for the exploration of instances in the target task. The learned model is enforce to be well-performed on two different objectives, which is cast as a bi-level optimization with effective solutions in a reliable generalization. Empirical results verify our effectiveness and robustness.

References

[Bennett *et al.*, 2002] K. P. Bennett, A. Demiriz, and R. Maclin. Exploiting unlabeled data in ensemble methods. In *ACM SIGKDD*, pages 289–296, Edmonton, Canada, 2002.

[Colson *et al.*, 2007] B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *ANOR*, 153(1):235–256, 2007.

[Dean *et al.*, 2012] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. W. Senior, P. A. Tucker, K. Yang, and A. Y. Ng. Large scale distributed deep networks. In *NIPS*, pages 1232–1240, Lake Tahoe, NV, 2012.

[Duan *et al.*, 2012] L. X. Duan, D. Xu, and I. W. Tsang. Domain adaptation from multiple sources: A domain-dependent regularization approach. *IEEE TNNLS*, 23(3):504–518, 2012.

[Guo *et al.*, 2020] L.-Z. Guo, Z.-Y. Zhang, Y. Jiang, Y.-F. Li, and Z.-H. Zhou. Safe deep semi-supervised learning for unseen-class unlabeled data. In *ICML*, pages 3897–3906, Virtual Event, 2020.

[Hinton *et al.*, 2015] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

[Hoffman *et al.*, 2012] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko. Discovering latent domains for multisource domain adaptation. In *ECCV*, pages 702–715, Florence, Italy, 2012.

[Hoffman *et al.*, 2018] J. Hoffman, M. Mohri, and N. S. Zhang. Algorithms and theory for multiple-source adaptation. In *NIPS*, pages 8256–8266, Montréal, Canada, 2018.

[Jang *et al.*, 2019] Y. Jang, H. Lee, S. J. Hwang, and J. Shin. Learning what and where to transfer. In *ICML*, pages 3030–3039, Long Beach, CA, 2019.

[Kingma and Ba, 2015] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, San Diego, CA, 2015.

[Laine and Aila, 2017] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, Toulon, France, 2017.

[LeCun *et al.*, 2015] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[Lee, 2013] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop*, page 2, 2013.

[Li *et al.*, 2021] Y.-F. Li, L.-Z. Guo, and Z.-H. Zhou. Towards safe weakly supervised learning. *IEEE TPAMI*, 43(1):334–346, 2021.

[Liao, 2013] H. Liao. Speaker adaptation of context dependent deep neural networks. In *IEEE ICASSP*, pages 7947–7951, Vancouver, Canada, 2013.

[Mancini *et al.*, 2019] M. Mancini, L. Porzi, S. R. Bulo, B. Caputo, and E. Ricci. Inferring latent domains for unsupervised deep domain adaptation. *IEEE TPAMI*, 43(1):485–498, 2019.

[Mansour *et al.*, 2008] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In *NIPS*, pages 1041–1048, Vancouver, Canada, 2008.

[Miyato *et al.*, 2018] T. Miyato, S. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE TPAMI*, 41(8):1979–1993, 2018.

[Murugesan *et al.*, 2016] K. Murugesan, H. X. Liu, J. G. Carbonell, and Y. M. Yang. Adaptive smoothed online multi-task learning. In *NIPS*, pages 4296–4304, Barcelona, Spain, 2016.

[Pentina and Lampert, 2017] A. Pentina and C. H. Lampert. Multi-task learning with labeled and unlabeled tasks. In *ICML*, pages 2807–2816, Sydney, Australia, 2017.

[Qi and Luo, 2019] G.-J. Qi and J. B. Luo. Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods. *ArXiv:1903.11260*, 2019.

[Ren *et al.*, 2018] M. Ren, W. Y. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, pages 4331–4340, Stockholm, Sweden, 2018.

[Romero *et al.*, 2015] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, San Diego, CA, 2015.

[Shalev-Shwartz and Ben-David, 2014] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

[Shi and Li, 2019] F. Shi and Y.-F. Li. Rapid performance gain through active model reuse. In *IJCAI*, pages 3404–3410, Macao, China, 2019.

[Shu *et al.*, 2019] J. Shu, Q. Xie, L. X. Yi, Q. Zhao, S. P. Zhou, Z. B. Xu, and D. Y. Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NIPS*, pages 1917–1928, Vancouver, Canada, 2019.

[Sim *et al.*, 2002] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. In *IEEE AFGR*, pages 53–58, Washington, D.C., 2002.

[Sinha *et al.*, 2018] A. Sinha, P. Malo, and K. Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE TEC*, 22(2):276–295, 2018.

[Vaswani *et al.*, 2017] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, Long Beach, CA, 2017.

[Xian *et al.*, 2018] Y. Q. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE TPAMI*, 41(9):2251–2265, 2018.

[Xiong *et al.*, 2014] C. Xiong, S. McCloskey, S.-H. Hsieh, and J. J. Corso. Latent domains modeling for visual domain adaptation. In *AAAI*, pages 2860–2866, Québec City, Canada, 2014.

[Yang *et al.*, 2007] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM MM*, pages 188–197, Augsburg, Germany, 2007.

[Yang *et al.*, 2019] Q. Yang, Y. Liu, T. J. Chen, and Y. X. Tong. Federated machine learning: Concept and applications. *ACM TIST*, 10(2):1–19, 2019.

[Ye *et al.*, 2018] H.-J. Ye, D.-C. Zhan, Y. Jiang, and Z.-H. Zhou. Rectify heterogeneous models with semantic mapping. In *ICML*, pages 5630–5639, Stockholm, Sweden, 2018.

[Zhao *et al.*, 2019] S. Zhao, M. M. Fard, H. Narasimhan, and M. Gupta. Metric-optimized example weights. In *ICML*, pages 7533–7542, Long Beach, CA, 2019.

[Zhao *et al.*, 2020] P. Zhao, L.-W. Cai, and Z.-H. Zhou. Handling concept drift via model reuse. *ML*, 109(3):533–568, 2020.

[Zhou and Belkin, 2014] X. Y Zhou and M. Belkin. Semi-supervised learning. In *Academic Press Library in Signal Processing*, volume 1, pages 1239–1269. 2014.

[Zhou, 2016] Z.-H. Zhou. Learnware: on the future of machine learning. *FCS*, 10(4):589–590, 2016.