Neurocomputing 467 (2022) 418–426

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Discriminative adversarial domain generalization with meta-learning based cross-domain validation

Keyu Chen [a,*], Di Zhuang [a], J. Morris Chang [a]

[a] Department of Electrical Engineering, University of South Florida, Tampa, FL 33620, United States

ARTICLE INFO

Article history:
Received 18 October 2020
Revised 20 August 2021
Accepted 25 September 2021
Available online 30 September 2021
Communicated by Zidong Wang

Keywords:
Convolutional neural network
Domain generalization
Discriminative adversarial learning
Meta-learning

ABSTRACT

The generalization capability of machine learning models, which refers to generalizing the knowledge for an "unseen" domain via learning from one or multiple seen domain(s), is of great importance to develop and deploy machine learning applications in the real-world conditions. Domain Generalization (DG) techniques aim to enhance such generalization capability of machine learning models, where the learnt feature representation and the classifier are two crucial factors to improve generalization and make decisions. In this paper, we propose Discriminative Adversarial Domain Generalization (DADG) with meta-learning-based cross-domain validation. Our proposed framework tries to learn a domain-invariant feature representation from source domains and generalize it to the unseen domains. It contains two main components that work synergistically to build a domain-generalized Deep Neural Network (DNN) model: (i) discriminative adversarial learning, which proactively learns a generalized feature representation on multiple "seen" domains, and (ii) meta-learning based cross domain validation, which simulates train/test domain shift via applying meta-learning techniques in the training process. In the experimental evaluation, a comprehensive comparison has been made among our proposed approach and other existing approaches on three benchmark datasets. The results shown that DADG consistently outperforms a strong baseline DeepAll, and outperforms the other existing DG algorithms in most of the evaluation cases.

1. Introduction

Machine Learning (ML) and Deep Learning (DL) have achieved great success in numerous applications, such as skin lesion analysis [1,2], human activity recognition [3,4], active authentication [5], facial recognition [6–8], botnet detection [9–11] and community detection [12,13]. Most of the ML/DL applications are underlying the assumption that the training and testing data are drawn from the same distribution (domain). However, in practice, it is more common that the data are from various domains. For instance, the image data for the medical diagnosis application might be collected from different hospitals, by different types of devices, or using different data preprocessing protocols. The domain shift issue results in a rapid performance degradation, where the machine learning applications is trained on "seen" domains and tested on other "unseen" domains. Even well-known strong learners such as deep neural networks are sensitive to domain shifts [14]. It is crucial to enhance the generalization capability of machine learning models in the real-world applications. Because, on one hand, it is costly to re-collect/label the data and re-train the model for such "unseen" domains. On the other hand, we can never enumerate all the "unseen" domains in advances.

Domain Generalization (DG), as illustrated in Fig. 1, which aims to learn a domain-invariant feature representation from multiple given domains and expecting good performance on the "unseen" domains. It is one of the techniques that aiming to enhance the generalization capability of machine learning models. However, designing an effective domain generalization approach is challenging. First, a well-designed DG approach should be model-agnostic. Domain shift is a general problem in the designing of ML/DL models, such that the approach should not be designed for a specific network architecture. Second, an effective DG approach should not be data-dependent. There exists different types of domain shift, such as different art forms or different centric-images. A data-dependent approach can lead promising results on some datasets. However, the approach can be overfitting to the particular domain

[*] Corresponding author.
E-mail address: keyu@usf.edu (K. Chen).

https://doi.org/10.1016/j.neucom.2021.09.046
0925-2312/© 2021 Elsevier B.V. All rights reserved.

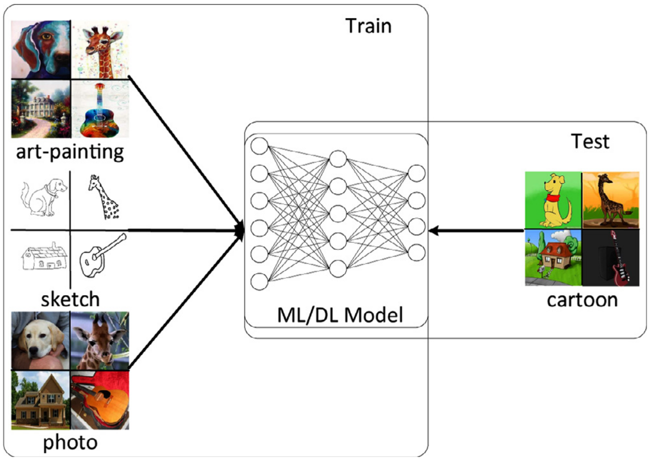


Fig. 1. Multi-source Domain Generalization: training a model on one or multiple seen source domains and test on certain "unseen" target domain.

shift and might not have comparable performance on the other datasets. Hence, it is a challenging task to design an effective DG approach.

To date, a few algorithms have been proposed to enhance the generalization capability of ML/DL models. For instance, D-SAM [15] designs a domain-specific aggregation module for each "seen" domain, and plugs it on a particular network architecture to eliminate the domain specific information. However, it is a model-based approach, because the aggregation module is designed for a particular model, and additional implementation of aggregation module is required when the model changed. Hex [16] is proposed to learn robust representations cross various domains via reducing the model dependence on high-frequency textural information. The original supervised model is trained with an explicit objective to ignore the superficial statistics, which only presents in certain datasets. Its representation learning is fully unsupervised, and performs good on certain image datasets. However, due to the assumption of domain shift and the unsupervised natural, Hex might not have the promising performance on the other image datasets. Approaches that leveraging the idea of meta-learning for domain generalization have been also proposed [17–19]. For instance, MLDG [17] was inspired by MAML [20] to simulate the domain-shift and optimize meta-train and meta-test together during the training phase. However, it only focuses on the classifier optimization, and lacks of effective guidance on the feature representation learning, where the better feature representation can benefit the classifier to make decisions.

In this paper, we present a novel DG approach, Discriminative Adversarial Domain Generalization (DADG). Our DADG contains two main components, discriminative adversarial learning (DAL) and meta-learning based cross domain validation (Meta-CDV). We adopt the DAL to learn the set of features, which provides domain-invariant representation for the following classification task, and apply the Meta-CDV to further enhance the robustness of the classifier. Specifically, on one hand, we consider the DAL component as a discriminator that trains a domain-invariant feature extractor by distinguishing the source domain of corresponding training data. On the other hand, we employ meta-learning optimization strategy to "boost" the objective task classifier by validating it on previously "unseen" domain data in each iteration. The two components guide each other from both feature representation and object recognition level via a model-agnostic process over iterations to build a domain-generalization model. Note that our DADG makes no assumption on the datasets, and it is a model-agnostic approach, which can be applied to any network architectures.

In the experimental evaluation, a comprehensive comparison has been made among our DADG and other 8 existing DG algorithms, including DeepAll (i.e., the baseline that simply used pre-trained network, without applying any DG techniques), TF [21], Hex [16], D-SAM [15], MMD-AAE [22], MLDG [17], Feature-Critic (FC) [18], and JiGen [23]. We conduct the comparison and the evaluation of our approach on three well-known DG benchmark datasets: PACS [21], VLCS [24] and Office-Home [25], utilizing two deep neural network architectures, AlexNet and ResNet-18. Our experimental result shows that our approach performs well at cross domain recognition tasks. Specifically, we achieve the best performance on 2 datasets (VLCS and Office-Home) and performs 2^{nd} best on PACS. For instance, on VLCS dataset, we improve on the strong baseline DeepAll by 2.6% (AlexNet) and 3.11% (ResNet-18). Moreover, an ablation study also conducted to evaluate the influence of each component in DADG.

To summarize, our work has the following contributions:

- We present a novel, effective and model-agnostic framework, Discriminative Adversarial Domain Generalization (DADG) to tackle the DG problem. Our approach adopts discriminative adversarial learning to learn the domain-invariant feature extractor and utilizes meta-learning optimization strategy to enhance the robustness of the classifier.
- To the best of our knowledge, DADG is the first work that uses meta-learning optimization to regularize the feature learning of discriminative adversarial learning in domain generalization.
- A comprehensive comparison among our algorithm and the state-of-the-art algorithms has been conducted (Section 4). For the sake of reproducibility and convenience of future studies about domain generalization, we have released our prototype implementation of DADG. [1]

The rest of this paper is organized as follows: Section 2 presents the related literature review. Section 3 presents the notations in common domain generalization problem, and describes our proposed algorithm. Section 4 presents the experimental evaluation. Section 5 presents the conclusion.

2. Related work

2.1. Generative Adversarial Nets (GAN)

Generative Adversarial Nets (GAN) [26] aims to approximate the distribution P_d of a dataset via a generative model. GAN simultaneously trains two components generator G and discriminator D. The two components, generator and discriminator can be built from neural networks (e.g., convolutional layers and fully connected layers). The input of G is sampled from a prior distribution $P_z(z)$ through which G generates fake samples similar to the real samples. Meanwhile, D is trained to differentiate between fake samples and real samples, and sends feedback to G for improvement. GAN can be formed as a two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(G,D) = E_{x \sim P_d}[log(D(x))] + E_{z \sim P_z}[log(1 - D(G(z)))] \quad (1)$$

GAN-based discriminative adversarial learning is able to learn a latent space from multiple different domains, where the latent space is similar to the given domains. It has been used in some domain adaptation works, which we will discuss below.

[1] https://github.com/keyu07/DADG.

2.2. Domain adaptation

Domain adaptation (DA) is one of the closely related work to domain generalization. The main difference between DA and DG is that DA assumes unlabeled target data is available during the training phase, but DG has no access to the target data. Many domain adaptation algorithms [27,14,28,29] are designed via mapping the source and the target domain into a domain-invariant feature space. GAN or GAN-based discriminative adversarial techniques have been utilized in many such domain adaptation works. For instance, ADDA [27] maps the data from the target domain to the source domain through training a domain discriminator. DANN [14] is proposed to train a ‘‘domain classifier” to learn the latent representations of the source and the target domains. Tzeng et al. [29] proposes to use multiple adversarial discriminators to apply on the data of different available source domains. Discriminative adaversarial learning successfully learns the domain-invariant feature representation, which considered as a latent space that similar to all source domains. This success motivates us to optimize the feature learning of domain generalization.

2.3. Domain generalization

In contrast to domain adaptation, domain generalization is a more challenging problem, because it requires no prior knowledge about the target domain. Given a ML/DL application that has multiple ‘‘seen” or/and ‘‘unseen” domains, we observe that each domain has two elements: the private element and the global element. The private element contains the specific representation/information of each domain, while the global element holds the invariant features across different domains. Most of the recent domain generalization works aim to improve the learnt feature by using one of the two strategies: (i) Eliminating the influence of the private elements or (ii) Extracting the global elements. Other than the two main strategies, there are other alternative studies, such as a data augmentation based method [30] and a recent self-supervised learning method JiGen [23]. JiGen [23] uses a jigsaw-puzzle classifier to guide the feature extractor to capture the most informative part of the images, and it achieves current state-of-the-art results on three domain generalization benchmark datasets. We include JiGen [23] in all our evaluations.

Many model-enhancement based studies are proposed under the first strategy. For instance, Li et al. [21] develops a low-rank parameterized network to decrease the size of parameters. D’Innocente et al. [15] proposes to build domain-specific aggregation modules and stack on the backbone network to merge specific and generic information. However, it is a model based approach. Because one set of aggregation modules can only apply on one particular backbone network. Additional implementation is required when we change the network architecture. Hex [16] is proposed to learn robust representations cross various domains via reducing the model dependence on high-frequency textural information. The original supervised model is trained with an explicit objective to ignore the so called superficial statistics, which is presented in the training set but may not be present in future testing sets. Its representation learning is fully unsupervised, and performs good on certain image datasets. However, because the assumption of domain shift and the unsupervised natural of Hex, it might not have the comparable good performance on the other image datasets. However, designing an approach to weaken certain types of domain-specific elements may suffer from overfitting on such domain elements. Though some outstanding results have been shown by this kind of approaches on certain datasets, while may not be able to be generalized to many more ‘‘unseen” domains. For instance, the different domain types are considered as different art forms or different centric-images.

For the second strategy, most of the previous works are focusing on learning domain-invariant representation, which is able to capture the important similar information among multiple different domains and have the capability of generalizing to more ‘‘unseen” domains. As such, these works are more similar to the work of domain adaptation. For instance, Ghifary et al. [31] proposes to learn domain-invariant features via a multi-domain reconstruction auto-encoder. However, the effectiveness for reconstruction the auto-encoder is limited while applying to more complex datasets [21]. Motiian et al. [32] employs maximum mean discrepancy (MMD) and proposes to learn a latent space that minimizes the distance among images that have the same class label but different domains. Li et al. [22] proposes to align source domains to learn a domain-agnostic representation using adversarial autoencoders with MMD constraints, and uses adversarial learning to match the distribution of generated data with a prior distribution.

Our approach also belongs to the second strategy. We use the discriminative adversarial learning to learn a latent distribution among the source domains. By doing so, we achieve a domain-invariant feature representation that different domains are indistinguishable. Beyond the domain-invariant feature representation, in order to improve the relevant classification task, we also propose a more robust classifier, by using meta-learning based optimization, which leads more competitive classification results. To the best of our knowledge, this is the first work that uses meta-learning optimization to regularize the discriminative adversarial learning in domain generalization.

2.4. Meta-learning

Meta-learning introduces a concept ‘‘learning-to-learn” and recently receives great interests with applications including few-shot learning [20,33,34] and learning optimizations [35,36]. It learns from various tasks during training and such that the model can be quickly generalized to new tasks. MAML [20] is typical in those works. It utilizes sampled episodes during training, where each episode is designed to simulate the few-shot tasks in a train-test split manner. Recently, a few works have applied this episodic meta-learning optimization method in domain generalization [17,19,18]. For instance, MLDG [17] borrows the idea of [20] to optimize the classifier, by simulating the train-test domain shift during training phase. MetaReg [19] proposes to learn a regularization function for the network classifier. Li et al. [18] proposes to simultaneously learn an auxiliary loss and measure whether the performance of validation set has been improved. However, MLDG [17] and MetaReg [19] only focus on classifier optimization, and are lacking of details addressing the learning of a domain-invariant feature space. The success of meta-learning method on the enhancement of classifier robustness motivates us to optimize the network classifier for domain generalization. To summarize, in order to address the challenging domain generalization problem, we apply discriminative adversarial learning and meta-learning, where the discriminative adversarial learning extracts domain-invariant feature representation, and meta-learning enhances the classifier robustness.

3. Methodology

The design of DADG is based on our assumption that there exists a domain-invariant feature representation, which contains the common information for both the ‘‘seen” and ‘‘unseen” domains. It should satisfy the following properties: (i) The feature representation should be invariant in terms of data distributions (domains). Since ML/DL models are designed to transfer the knowledge from seen domains to unseen domains, they could fail if the distributions differ a lot. (ii) It should keep the variance between different objects (classes). This helps the model to capture the unique information of different objects and to make precise decisions. We use two key components in DADG to address the above two properties: discriminative adversarial learning (DAL) and meta-learning based cross domain validation (Meta-CDV). DAL aims to learn a domain-invariant feature representation where different data distributions are indistinguishable. Therefore, the domain variance will be minimized. Meta-CDV brings the learnt features to supervised learning by training a classifier in a meta-learning manner. It evaluates the validation performance of previous unseen domains within each training iteration.

We introduce Fig. 2 to better illustrate our DADG in high level. The goal of DADG is to find the optimized feature representation point, which satisfies the two properties. A, B and C present the different domains. DAL and Meta-CDV address DG in two aspects: (i) As shown by the orange lines, the dash lines are the gradient directions when tackling feature learning on different domains ∇D_A and ∇D_B, respectively. While the solid line is the actual gradient direction guided by DAL and finally reaches a representation point indistinguishable from given domains. (ii) As shown by the blue lines, the dash lines indicate the gradient directions when solving certain tasks ∇T_A and ∇T_B, respectively. While the solid line denotes the actual gradient direction led by classification task on two domains and further optimized by cross domain validation (∇T_C). The model finally learns a domain-invariant feature representation point that satisfies the two properties.

In the rest of this section, we denote the input data space as $x \in X$, the class label space as $y \in Y$ and the domain label (i.e., belonging to which distribution) space as $y^d \in Y^d$. The source domains are described as $D_i \in S$, and the target domains as T. Also, please note that in the rest of this section, the superscript of each parameter indicates different updating stages within one iteration, denoted as m, while the subscript indicates different iterations, denoted as n. We introduce our two main components in the remaining sections: DAL in 3.1 and Meta-CDV in 3.2. Finally we summarize the two components together in 3.3.

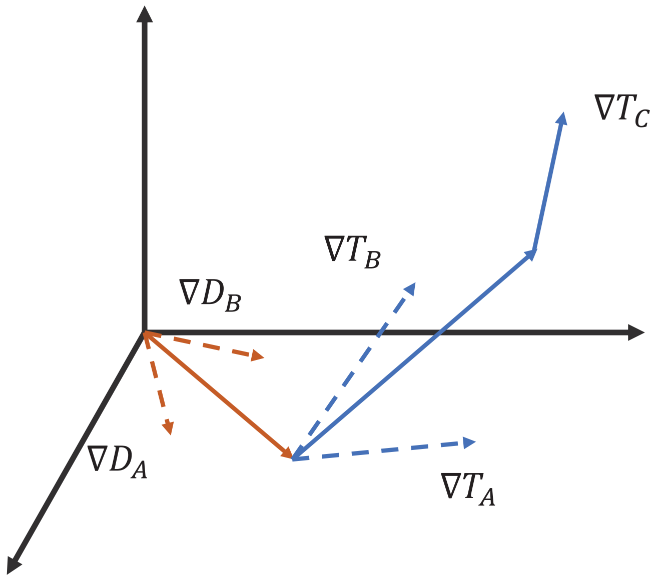


Fig. 2. Diagram of our DADG. Better view in colors.

3.1. Discriminative adversarial learning

As described above, the goal of this component is to learn a domain classification model, which aims to classify data from different domains. We consider our DAL containing two parts: (i) a feature extractor f_θ with parameter θ, and (ii) a discriminator d_ψ with parameter ψ. Both θ and ψ are learnable parameters during training phase.

In our approach, we first randomly divide the source domains S into two mutually exclusive sets: S_d for DAL and S_c for Meta-CDV. The discriminator acts as a domain classifier, which takes the learnt sample features $f_\theta(x_j)$ of each arbitrary input x_j and tries to discriminate its domain label y^d. Thus, we need to learn the parameters (ψ) that minimize the classification loss, which as follows:

$$\mathscr{L}_{disc}(d_{\psi_n^m}(f_{\theta_n^m}(x_j)), y_j^d) \tag{2}$$

The loss function of DAL is presented as follows:

$$F(\cdot) = \sum_{D_i \in S_d} \sum_{x_j \in D_i} \mathscr{L}_{disc}(d_{\psi_n^m}(f_{\theta_n^m}(x_j)), y_j^d) \tag{3}$$

The objective of the feature extractor is to maximize the discriminative loss, to achieve indistinguishable of the learnt feature representation. Following the design of GAN [26], the objective function of our discriminative adversarial learning can be written as the following minimax optimization:

$$\underset{\psi_n^m}{argmin}\, \underset{\theta_n^m}{max} F(\cdot) \tag{4}$$

Such minimax parameter updating can be achieved by gradient reversal layer (GRL) [28], which placed between the feature extractor and discriminator. During forward propagation, GRL keeps the learnable parameters same. During back propagation, it multiply the gradient by $-\lambda$ and pass it to the preceding layer.

To summarize, we update the parameters of feature extractor and discriminator as follows:

$$\theta_n^{m+1} \leftarrow \theta_n^m - \alpha \cdot \nabla(-\lambda \cdot F(\cdot)) \tag{5}$$

$$\psi_{n+1}^m \leftarrow \psi_n^m - \alpha \cdot \nabla F(\cdot) \tag{6}$$

where the α is the DAL learning rate. Thereafter, the θ_n^{m+1} will be shared in further training within the same iteration (as we illustrated in Fig. 3 step ①), and φ_{n+1}^m will be used in the next iteration.

3.2. Meta-learning based cross domain validation

After the feature extractor has been trained to minimize the domain variance, we adopt meta-learning based cross domain validation (Meta-CDV) to address the enhancement of the classifier robustness. Robust classifier is able to help the feature extractor to keep the discriminant power between various classes. This is accomplished by training the classification model on 2 seen domains S_d in DAL and validating the performance on cross domains S_c.

To train the model on seen domains S_d, the classification model is composed of the feature extractor f_θ from DAL and a classifier c_φ with parameters φ. The training loss is defined as follows:

$$\mathscr{L}_{train}(c_{\varphi_n^m}(f_{\theta_n^{m+1}}(x_j), y_j) \tag{7}$$

where x_j is an arbitrary input and y_j is the corresponding output label.

The loss function of classification training on seen domains is presented as follows (as illustrated in Fig. 3 step ②):

$$G(\cdot) = \sum_{D_i \in S_d} \sum_{x_j \in D_i} \mathscr{L}_{train}(c_{\varphi_n^m}(f_{\theta_n^{m+1}}(x_j)), y_j) \tag{8}$$

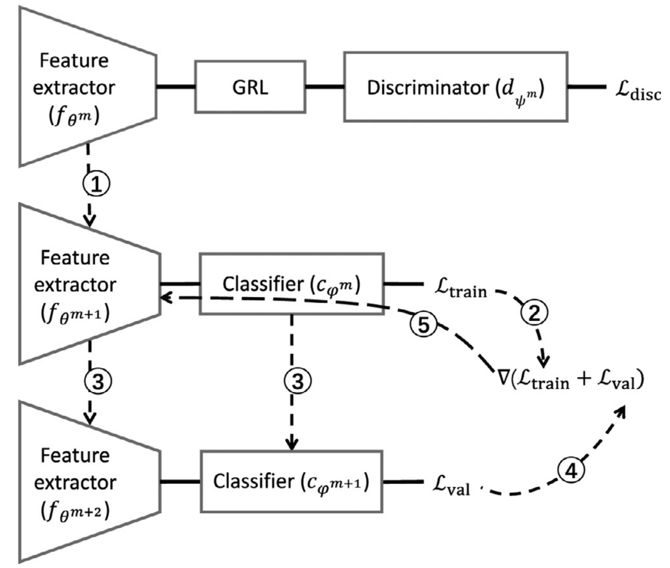

Feature extractor (f_{θ^m})
GRL
Discriminator (d_{ψ^m})
$\mathcal{L}_{\text{disc}}$
①
Feature extractor ($f_{\theta^{m+1}}$)
Classifier (c_{φ^m})
$\mathcal{L}_{\text{train}}$
②
⑤
$\nabla(\mathcal{L}_{\text{train}} + \mathcal{L}_{\text{val}})$
③
③
Feature extractor ($f_{\theta^{m+2}}$)
Classifier ($c_{\varphi^{m+1}}$)
$\mathcal{L}_{\text{val}}$
④

Fig. 3. The training flow of DADG.

Note that the training is performed over the updated feature extractor parameters θ^{m+1} in DAL. As such, the parameters are updated as follows:

$$\theta_n^{m+2} \leftarrow \theta_n^{m+1} - \beta \cdot \nabla G(\cdot) \tag{9}$$

$$\varphi_n^{m+1} \leftarrow \varphi_n^{m} - \beta \cdot \nabla G(\cdot) \tag{10}$$

where the β is the classification learning rate. Here the updated parameter θ_n^{m+1} is involved in the calculation of training loss. It also means that we need the second derivative with respect to θ, while minimizing the loss function 8.

After finishing the classification task on seen domains, we evaluate the performance on cross domains S_c to boost the classification model. This process simulates the virtual train/test settings. The evaluation is performed on the updated parameters θ_n^{m+2} and φ_n^{m+1} (as illustrated in Fig. 3 step ③). More concretely, this evaluation come up with the cross domain validation loss:

$$\mathscr{L}_{val}(c_{\varphi_n^{m+1}}(f_{\theta_n^{m+2}}(x_j)), y_j) \tag{11}$$

The loss function of cross domain validation is as follows:

$$H(\cdot) = \sum_{D_i \in S_c} \sum_{x_j \in D_i} \mathscr{L}_{val}(c_{\varphi_n^{m+1}}(f_{\theta_n^{m+2}}(x_j)), y_j) \tag{12}$$

Finally, as illustrated in Fig. 3 step ②, ④ and ⑤, we update our classification model by adding the training loss $\mathscr{L}_{train}$ and cross domain validation loss $\mathscr{L}_{val}$ at the end of each iteration:

$$\theta_{n+1}^{m} \leftarrow \theta_n^{m+1} - \gamma \cdot \nabla H(\cdot) \tag{13}$$

$$\varphi_{n+1}^{m} \leftarrow \varphi_n^{m} - \gamma \cdot \nabla H(\cdot) \tag{14}$$

where γ presents the learning rate of cross domain validation. Note that the parameter updating on seen domains classification is performed over the parameter θ_n^{m+1} and φ_n^{m}, whereas the cross domain validation is evaluated over parameter θ_n^{m+2} and φ_n^{m+1}. In other words, the optimization of our classification model is involved in third derivative with respect to θ and second derivative with respect to φ.

3.3. Summary of DADG

As illustrated in Fig. 3, the DAL and Meta-CDV optimize the model by addressing different aspects of domain generalization, and work synergistically within one iteration. In each iteration, we randomly split the train/validation (S_d/S_c) domains. DAL learns a domain-invariant feature extractor (f_θ) by maximizing the discriminative loss. Then, our approach learns a robust classification model by adopting a simple classification training and cross domain validation, which optimized in meta-learning based manner. For the whole process, the objective function can be introduced as:

$$\underset{\psi_n^m}{argmin}\, \underset{\theta_n^m}{max} F(\cdot) + \underset{\theta_n^{m+1}, \varphi_n^m}{argmin}(G(\cdot) + H(\cdot)) \tag{15}$$

Once Eq. 15 is optimized to converge on the source domains, we evaluate the classification model using unseen domains.

4. Experimental evaluation

We conduct our experiments on 3 benchmark datasets (PACS [21], VLCS [24] and Office-Home [25]) and 2 deep neural network architectures with pretrained parameters (AlexNet [37] and ResNet-18 [38]) to evaluate the generalization capability of our proposed approach. A comprehensive comparison has been made among our approach and other baseline approaches. The presented results are shown that our DADG performs consistently comparable in all the evaluations, and achieves the state-of-the-art results in two datasets. The effectiveness of each component in our approach also discussed. All the details are described in following.

4.1. Baseline approaches

We compare our proposed approach performance with following baseline DG approaches.

- **DeepAll** is the baseline that simply use deep learning network to train the aggregation of all source domains and test the unseen domain. It is a strong baseline that surpasses many previous DG works [21].
- **TF** [21] introduces a low-rank parameter network to decrease the size of parameters. This work also shows that the DeepAll can surpass many previous studies and first provides PACS dataset.
- **Hex** [16] attempts to reduce the sensitivity of a model on high frequency texture information, and thus to increase model domain-robustness.
- **MMD-AAE** [22] is based on adversarial autoencoder. It aligns different domain distributions to an arbitrary prior via MMD regularization, to learn an invariant feature representation.
- **Feature-Critic(FC)** [18] aims to train a robust feature extractor. It uses meta-learning approach, along with an auxiliary loss to measure whether the updated parameter has improved the performance on the validation set.
- **MLDG** [17] is the first work that addresses domain generalization using meta-learning. It is inspired by MAML [20] and proposed visual cross domain classification task by splitting source domains into meta-train and meta-test.
- **D-SAM** [15] plugs parallel domain-specific aggregation modules on a given network architecture to neglect domain specific information.
- **JiGen** [23] is the first work that addresses DG by self-supervised learning. It divides each image into small patches and shuffle the order. Then, trains an object classifier and a jigsaw order

classifier simultaneously. It achieves the state-of-the-art results on the three datasets VLCS [24], PACS [21] and Office-Home [25].

4.2. Experimental datasets

We utilize three well-known domain generalization benchmark datasets.

- **VLCS** [24] is composed of 10,729 images with resolution 227 × 227, taken from 4 different datasets (i.e., domains): PASCAL **V**OC2007 [39], **L**abelMe [40], **C**altech101 [41] and **S**un09 [42]. It depicts 5 categories (i.e., classes): bird, car, chair, dog and person.
- **PACS** [21] contains more severe domain shifts than VLCS. PACS aggregates 9,991 images in 7 different classes: dog, elephant, giraffe, guitar, house, horse and person. It shared by 4 different domains: **P**hoto, **A**rt, **C**artoon and **S**ketch.
- **Office-Home** [25] was created to evaluate DA and DG algorithms for object recognition in deep learning. There are 15,592 images from 4 different domains: Art, Clipart, Product and real-world images, each domain includes 65 classes.

4.3. Experimental setting

All three benchmark datasets contain the data of four different domains. We first hold one domain (i.e., the target domain) for testing and the rest three for training. Then, in the training phase, we randomly select two domains to apply discriminative adversarial learning (DAL), and select one domain to boost our classifier by meta-learning based cross domain validation (Meta-CDV). Our discriminator consists of two fully connected layer with 1024 neurons each and one output layer with 1 neuron.

The neural network is updated by stochastic gradient descent (SGD) in 2000 iterations during training. We use cross-entropy loss for both DAL (domain classification task) and Meta-CDV (classification task). Negative log-likelihood loss also tested for classification task, but it hardly effects the performance.

For most the hyperparameters, we followed MLDG: base classification learning rate $\beta = 5 \times 10^{-4}$, cross domain validation learning rate $\gamma = 5 \times 10^{-4}$, momentum = 0.9 and weight decay = 5×10^{-5}. The DAL learning rate is $\alpha = 5 \times 10^{-5}$. While a big α value will lead an unstable training process and 5 $\times 10^{-5}$ is appropriate for PACS, VLCS and Office-Home. The value of α should be picked carefully on the other datasets, 1/10 of the β and γ is suggested. The model-agnostic can be achieved by simply changing the backbone network architectures without additional implementation. All of our experiments are implemented using PyTorch, on a server with GTX 1080Ti 11 GB GPU.

4.4. Effectiveness analysis

In this section, we discuss the performance of our proposed approach and the baseline approaches in terms of classification accuracy. Tables 1–3 show the results of datasets VLCS, PACS and Office-Home. To make a more comprehensive comparison, we implement MLDG our own, because only demo code is provided by the author. Besides, we implement Hex, Feature-Critic, D-SAM and JiGen by using the code that are provided by the authors. All the implementations are evaluated on the datasets or network architectures they did not report. Our results of these approaches are highlighted in the three tables with *. The details of each dataset are presented below:

VLCS: We follow the standard protocol of MTAE [31] to randomly divide the data of each source domain into training (70%) and testing (30%) sets. Finally we test on all the images in target domain. The upper and bottom part of Table 1 show the results when using different network architectures AlexNet and ResNet-18, respectively. From Table 1, we can observe that (i) The baseline DeepAll performs competitively and surpasses many previous DG works on overall performance, such as HEX, Feature-Critic and D-SAM. But our approach outperforms DeepAll in all target domain cases and on different network architectures. (ii) On AlexNet, our DADG performs better than DeepAll by 2.6% and better than Jigen by 1.27%, such that we achieve the new state-of-the-art result on VLCS dataset. More specifically, DADG provides the best results in two (i.e., VOC and SUN respectively) out of four target cases. (iii) On ResNet-18, DADG surpasses the previous SOTA result Jigen in average performance and performs the best in three out of four target domain cases.

PACS: We follow the training protocol of TF, considering three domains as source domains and the remaining one as target. The evaluation results are shown in Table 2, we can see that: (i) On AlexNet, although we do not achieve the best performance on any target domain cases, our DADG provides consistently comparable results, and performs the 2^{nd} best in average results. (ii) On ResNet-18, we have two best results on Art-paint (79.89%) and Cartoon (76.25%), and only slight worse (0.34%) than the best JiGen in average performance.

Office-Home: We follow the protocol of D-SAM, also considering three as source domains and the rest one as target. The results are shown in Table 3, and we can observe that: (i) The advantage of D-SAM in average results originates from its results on Art and Clipart, but the rest two were lower than DeepAll. (ii) Our DADG achieves the best in two target cases and the best in average results, and improves the previous SOTA result Jigen by 1.02%.

Summary of the Experimental Evaluation: From the experimental evaluation analyzed above, we conclude that: (i) DeepAll exceeds many previous approaches in different datasets. In general, only MLDG, JiGen and our DADG can outperform DeepAll in all three datasets. (ii) As we mentioned in Section 2.3. The approaches that aim to neglect particular domain-specific information, may assist the model in some datasets but fail in others. For instance, HEX and D-SAM are better than DeepAll on PACS, but worse than DeepAll on VLCS. (iii) our DADG has consistently comparable results in all the datasets and achieves the SOTA results on VLCS and Office-Home, also the second best on PACS. On VLCS and Office-Home, DADG outperforms the previous SOTA JiGen all over 1%.

4.5. Impact of different DADG components

In this section, we conduct an extended study using PACS dataset with network architecture AlexNet to investigate the impact of the two key components (i.e., DAL and Meta-CDV) in our proposed approach DADG. Specifically, we test the performance in terms of classification accuracy by excluding each component in our approach respectively. DADG-DAL only contained the discriminative adversarial learning (DAL) component and trained the classification model conventionally instead of in meta-learning manner. While DADG-CDV meant that we removed the DAL component and only updated the classification model parameters in meta-learning manner.

From the results in Table 4, we can see that DADG-DAL and DADG-CDV consistently perform better than DeepAll, and our full version DADG surpasses both baseline models in average performance and in every target domain cases. In the comparison between DADG-DAL and DADG-CDV, the DADG-DAL consistently better than the DADG-CDV. The results in Table 4 show that: (i) Employing discriminative adversarial learning is able to effectively

Table 1
Cross domain classification accuracy (in %) on VLCS dataset when using network architecture AlexNet and ResNet-18. The results of our implementation were the average over 20 repetitions. Each column name indicates the target domain. Best performance in bold.

VLCS	VOC	LabelMe	Caltech	Sun	Avg.
		AlexNet			
TF [21]	69.99	63.49	93.63	61.32	72.11
HEX* [16]	68.51	**63.67**	89.63	62.12	70.98
MMD-AAE [22]	67.70	62.60	94.40	64.40	72.28
FC* [18]	66.79	61.48	95.68	63.13	71.77
MLDG* [17]	70.01	61.06	95.68	65.08	72.96
D-SAM [15]	63.75	54.81	94.96	64.56	69.52
JiGen [23]	70.62	60.90	**96.93**	64.30	73.19
DeepAll	68.11	61.30	94.44	63.58	71.86
DADG	**70.77**	63.44	96.80	**66.81**	**74.46**
		ResNet-18			
MLDG* [17]	74.41	63.45	96.75	69.35	75.99
D-SAM* [15]	70.42	58.70	88.90	**71.36**	72.35
JiGen* [23]	74.91	63.00	98.39	69.37	76.42
DeepAll	73.84	62.17	97.10	67.28	75.10
DADG	**76.17**	**67.22**	**98.50**	70.95	**78.21**

Table 2
Cross domain classification accuracy (in %) on PACS dataset when using network architecture AlexNet and ResNet-18. The results of our implementation were the average over 20 repetitions. Each column name indicates the target domain. Best performance in bold.

PACS	Photo	Art-paint	Cartoon	Sketch	Avg.
		AlexNet			
TF [21]	89.50	62.86	66.97	57.51	69.21
HEX [16]	87.90	66.80	69.70	56.30	70.18
FC [18]	**90.10**	64.40	68.60	58.40	70.38
MLDG[17]	88.00	66.23	66.88	58.96	70.02
D-SAM [15]	85.55	63.87	70.70	64.66	71.20
JiGen [23]	89.00	**67.63**	**71.71**	**65.18**	**73.38**
DeepAll	88.65	63.12	66.16	60.27	69.55
DADG	89.76	66.21	70.28	62.18	72.11
		ResNet-18			
MLDG* [17]	94.03	76.42	73.03	68.15	77.91
D-SAM [15]	95.30	77.33	72.43	**77.83**	**80.72**
JiGen [23]	**96.03**	79.42	75.25	71.35	80.51
DeepAll	93.06	75.60	72.30	68.10	77.27
DADG	94.86	**79.89**	**76.25**	70.51	80.38

Table 3
Cross domain classification accuracy (in %) on Office-Home dataset when using ResNet-18. The results of our implementation were the average over 20 repetitions. Each column name indicates the target domain. Best performance in bold.

Office-Home	Art	Clipart	Product	Real-World	Avg.
		ResNet-18			
MLDG* [17]	52.88	45.72	69.90	72.68	60.30
D-SAM [15]	**58.03**	44.37	69.22	71.45	60.77
JiGen [23]	53.04	47.51	**71.47**	72.79	61.20
DeepAll	54.31	41.41	70.31	73.03	59.77
DADG	55.57	**48.71**	70.90	**73.70**	**62.22**

Table 4
Cross domain classification accuracy (in %) on PACS dataset using AlexNet. The results of our implementation were the average over 20 repetitions. Each column name indicates the target domain. Best performance in bold.

PACS	Photo	Art-paint	Cartoon	Sketch	Avg.
		AlexNet			
DeepAll	88.65	63.12	66.16	60.27	69.55
DADG-DAL	89.51	65.43	69.19	61.70	71.46
DADG-CDV	89.10	64.22	68.24	60.60	70.54
DADG	**89.76**	**66.21**	**70.28**	**62.18**	**72.11**

guide the feature extractor to learn the invariant features among multiple source domains. (ii) Since the only difference between DeepAll and DADG-CDV is the updating manner. Thus applying meta-learning based cross domain validation can make the classification model more robust. (iii) The full version DADG consistently performs the best in every single case, which has shown that combining domain invariant representation and robust classifier together helped the model to enhance generalization. (iv) The domain-invariant representation plays a more crucial role rather than the robust classifier. Because the invariant representation provides a easier task for the classifier to make decision.

4.6. Impact of linear related domains

We assume that there exists a domain-invariant feature representation for both source and target domains. However, it is also possible that some target domains are less relevant or even irrelevant to the source domains.

The domain types were considered as different art forms (art, cartoon in PACS) or different centric images (LabelMe and SUN in VLCS) in previous sections. It is very hard to define whether the target domain is less relevant to the source domains. To explore this situation, we conduct an experiment using digit images in six different angles as six different domains. To be more specific, we adopt the MNIST [43] dataset and randomly chose 1,000 images in each class and trained with AlexNet [44]. We denote the digit images rotated with 0°by R_0 and then rotate the digit images in a counter-clock wise direction by 15°, 30°, 45°, 60°and 75°. Since the rotation angles are continues related, which means sometimes the target domains are out of the scope of source domains (irrelevant). For example, when the R_0 and R_{15} are as the target domains, we consider that the target domains are out of the scope of source domains. During the training phase, 4 domains are selected as source domains and the rest 2 are target domains. For each iteration, our DADG randomly adopts 2 source domains for DAL and another 2 for Meta-CDV. The model performance will evaluated on the rest 2 target domains. A comparison is made among DeepAll, MLDG [17] and DADG.

Table 5
Cross domain classification accuracy (in %) on MNIST rotation dataset using AlexNet. The results of our implementation were the average over 10 repetitions. Best performance in bold.

Source	Target	DeepAll	MLDG* [17]	DADG
		AlexNet		
$R_{30}, R_{45}, R_{60}, R_{75}$	R_0, R_{15}	69.35	69.51	**69.57**
$R_{15}, R_{45}, R_{60}, R_{75}$	R_0, R_{30}	89.30	88.89	**89.53**
$R_{15}, R_{30}, R_{60}, R_{75}$	R_0, R_{45}	89.16	89.18	**89.29**
$R_{15}, R_{30}, R_{45}, R_{75}$	R_0, R_{60}	88.72	89.10	**89.18**
$R_{15}, R_{30}, R_{45}, R_{60}$	R_0, R_{75}	84.55	**84.64**	84.59
$R_0, R_{45}, R_{60}, R_{75}$	R_{15}, R_{30}	92.62	92.56	**92.72**
$R_0, R_{30}, R_{60}, R_{75}$	R_{15}, R_{45}	94.17	**94.43**	94.33
$R_0, R_{30}, R_{45}, R_{75}$	R_{15}, R_{60}	94.58	**94.65**	94.61
$R_0, R_{30}, R_{45}, R_{60}$	R_{15}, R_{75}	90.16	90.16	**90.19**
$R_0, R_{15}, R_{60}, R_{75}$	R_{30}, R_{45}	92.09	92.41	**92.62**
$R_0, R_{15}, R_{45}, R_{75}$	R_{30}, R_{60}	94.35	**94.44**	94.40
$R_0, R_{15}, R_{45}, R_{60}$	R_{30}, R_{75}	89.93	90.13	**90.24**
$R_0, R_{15}, R_{30}, R_{75}$	R_{45}, R_{60}	92.00	**92.28**	92.22
$R_0, R_{15}, R_{30}, R_{60}$	R_{45}, R_{75}	89.47	89.54	**89.65**
$R_0, R_{15}, R_{30}, R_{45}$	R_{60}, R_{75}	74.14	74.80	**74.83**
Average		88.31	88.45	**88.53**

From the results in Table 5, we can see that: (i) For all 3 approaches, the performance are better when the target domains close to 30°and 45°, and much worse when the target domains close to 0°and 75°. (ii) Our DADG outperforms the other two approaches in 10 out of total 15 different cases and achieves the best overall average accuracy among the 3 approaches. The results show the performance drop when the target domains are irrelevant to the source domains. It happens to all the approaches and can be considered as a common situation in domain generalization. Although our DADG outperforms other 2 in average, only 10/15 better than the MLDG. Compare to the performance on VLCS, PACS and Office-Home (Tables 1–3), our DADG does not show significant advantage on this experiment. Because we select 2 source domains to do discriminative adversarial learning (DAL), and the rest source domains will train with Meta-CDV. When the number of source domains increased, DAL only contributes small portion in each iteration. As we mentioned in the Section 4.5 (iv), DAL plays a more critical role than Meta-CDV. Finally, if we have a great number of source domains, the contribution of DAL can be even negligible. Thus, our DADG is sensitive to the number of source domains.

5. Conclusion

In this paper, we proposed DADG, a novel domain generalization approach, that contains two main components, discriminative adversarial learning and meta-learning based cross domain validation. The discriminative adversarial learning component learns a domain-invariant feature extractor, while the meta-learning based cross domain validation component trains a robust classifier for the objective task (i.e., classification task). Extensive experiments have been conducted to show that our feature extractor and classifier could achieve good generalization performance on three domain generalization benchmark datasets. Experimental results indicate that the feature extractor and classifier achieve good generalization on three benchmark domain generalization datasets. The experimental results also show that our approach consistently beat the strong baseline DeepAll. For instance, while using PACS dataset, our approach performs better than DeepAll by 1.56% (AlexNet) and 3.11% (ResNet-18). Notably, we also reach the state-of-the-art performance on VLCS and Office-Home datasets, and improve the average accuracy by over 1% in each case. As we mentioned in the 4.6, in current stage, our DADG is sensitive to the number of source domains because the DAL tends to be unimportant with the increasing number. In the future work, we plan to address this limitation and design a approach that can handle various number of source domains.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Effort sponsored in part by United States Special Operations Command (USSOCOM), under Partnership Intermediary Agreement No. H92222-15-3-0001-01. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. (The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the United States Special Operations Command.)

References

[1] F. Perez, S. Avila, E. Valle, Solo or ensemble? Choosing a cnn architecture for melanoma classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019.
[2] N.N. Di Zhuang, K. Chen, J.M. Chang, Saia: split artificial intelligence architecture for mobile healthcare systems, arXiv preprint arXiv:2004.12059..
[3] D. Tao, L. Jin, Y. Yuan, Y. Xue, Ensemble manifold rank preserving for acceleration-based human activity recognition, IEEE Trans. Neural Networks Learn. Syst. 27 (6) (2014) 1392–1404.
[4] D. Zhuang, J.M. Chang, Utility-aware privacy-preserving data releasing, arXiv preprint arXiv:2005.04369..
[5] P.-Y. Wu, C.-C. Fang, J.M. Chang, S.-Y. Kung, Cost-effective kernel ridge regression implementation for keystroke-based active authentication system, IEEE Trans. Cybern. 47 (11) (2016) 3916–3927.
[6] C. Ding, D. Tao, Trunk-branch ensemble convolutional neural networks for video-based face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 1002–1014.
[7] H. Nguyen, D. Zhuang, P.-Y. Wu, M. Chang, Autogan-based dimension reduction for privacy preservation, Neurocomputing..
[8] D. Zhuang, S. Wang, J.M. Chang, Fripal: face recognition in privacy abstraction layer, in: 2017 IEEE Conference on Dependable and Secure Computing, IEEE, 2017, pp. 441–448.
[9] L. Mai, D.K. Noh, Cluster ensemble with link-based approach for botnet detection, J. Netw. Syst. Manage. 26 (3) (2018) 616–639.
[10] D. Zhuang, J.M. Chang, Peerhunter: detecting peer-to-peer botnets through community behavior analysis, in: 2017 IEEE Conference on Dependable and Secure Computing, IEEE, 2017, pp. 493–500.
[11] D. Zhuang, J.M. Chang, Enhanced peerhunter: detecting peer-to-peer botnets through network-flow level community behavior analysis, IEEE Trans. Inf. Forensics Secur. 14 (6) (2018) 1485–1500.
[12] A. Tagarelli, A. Amelio, F. Gullo, Ensemble-based community detection in multilayer networks, Data Min. Knowl. Disc. 31 (5) (2017) 1506–1543.
[13] D. Zhuang, M.J. Chang, M. Li, Dynamo: Dynamic community detection by incrementally maximizing modularity, IEEE Trans. Knowl. Data Eng..
[14] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, in: Domain Adaptation in Computer Vision Applications, Springer, 2017, pp. 189–209..
[15] A. D'Innocente, B. Caputo, Domain generalization with domain-specific aggregation modules, in, German Conference on Pattern Recognition, Springer (2018) 187–198.
[16] H. Wang, Z. He, Z.L. Lipton, E.P. Xing, Learning robust representations by projecting superficial statistics out, in: International Conference on Learning Representations, 2019..
[17] D. Li, Y. Yang, Y.-Z. Song, T.M. Hospedales, Learning to generalize: Meta-learning for domain generalization, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018..
[18] Y. Li, Y. Yang, W. Zhou, T. Hospedales, Feature-critic networks for heterogeneous domain generalisation, in: The Thirty-sixth International Conference on Machine Learning, 2019..
[19] Y. Balaji, S. Sankaranarayanan, R. Chellappa, Metareg: towards domain generalization using meta-regularization, Advances in Neural Information Processing Systems (2018) 998–1008.
[20] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 1126–1135..

[21] D. Li, Y. Yang, Y.-Z. Song, T.M. Hospedales, Deeper, broader and artier domain generalization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5542–5550.

[22] H. Li, S. Jialin Pan, S. Wang, A.C. Kot, Domain generalization with adversarial feature learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5400–5409.

[23] F.M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, T. Tommasi, Domain generalization by solving jigsaw puzzles, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2229–2238.

[24] A. Torralba, A.A. Efros, et al., Unbiased look at dataset bias., in: CVPR, vol. 1, Citeseer, 2011, p. 7..

[25] H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5018–5027.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680..

[27] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7167–7176.

[28] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: International conference on machine learning, PMLR, 2015, pp. 1180–1189..

[29] E. Tzeng, J. Hoffman, T. Darrell, K. Saenko, Simultaneous deep transfer across domains and tasks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4068–4076.

[30] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel, Domain randomization for transferring deep neural networks from simulation to the real world, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2017, pp. 23–30.

[31] M. Ghifary, W. Bastiaan Kleijn, M. Zhang, D. Balduzzi, Domain generalization for object recognition with multi-task autoencoders, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 2551–2559.

[32] S. Motiian, M. Piccirilli, D.A. Adjeroh, G. Doretto, Unified deep supervised domain adaptation and generalization, in: The IEEE International Conference on Computer Vision (ICCV), 2017.

[33] A. Nichol, J. Schulman, Reptile: a scalable metalearning algorithm, arXiv preprint arXiv:1803.02999 2..

[34] A. Rajeswaran, C. Finn, S.M. Kakade, S. Levine, Meta-learning with implicit gradients, Advances in Neural Information Processing Systems (2019) 113–124.

[35] K. Li, J. Malik, Learning to optimize neural nets, arXiv preprint arXiv:1703.00441..

[36] M. Andrychowicz, M. Denil, S. Gomez, M.W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, N. De Freitas, Learning to learn by gradient descent, in: Advances in neural information processing systems, 2016, pp. 3981–3989..

[37] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105..

[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[39] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.

[40] B.C. Russell, A. Torralba, K.P. Murphy, W.T. Freeman, Labelme: a database and web-based tool for image annotation, Int. J. Comput. Vis. 77 (1–3) (2008) 157–173.

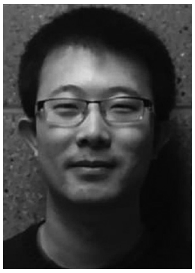
[41] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories, in: 2004 conference on computer vision and pattern recognition workshop, IEEE, 2004, pp. 178–178..

[42] M.J. Choi, J.J. Lim, A. Torralba, A.S. Willsky, Exploiting hierarchical context on a large database of object categories, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 129–136.

[43] Y. LeCun, C. Cortes, MNIST handwritten digit database [cited 2016-01-14 14:24:11]. URL: http://yann.lecun.com/exdb/mnist/..

[44] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105..

Keyu Chen received the MS degree in electrical engineering from the University of South Florida, Tampa. He is currently working toward the PhD degree in electrical engineering at the University of South Florida, Tampa. His research interests include machine learning, deep learning, natural language processing and data analytics.



Di Zhuang received the BE degree in computer science and information security from Nankai University, China. He received the PhD degree in electrical engineering at the University of South Florida, Tampa. His research interests include cyber security, social network science, privacy enhancing technologies, machine learning, and deep learning. He is a student member of the IEEE.

J. Morris Chang received the PhD degree from North Carolina State University. He is a professor with the Department of Electrical Engineering, University of South Florida. His past industrial experiences include positions at Texas Instruments, the Microelectronic Center of North Carolina, and AT&T Bell Labs. He received the University Excellence in Teaching Award at the Illinois Institute of Technology in 1999. His research interests include cyber security, wireless networks, and energy efficient computer systems. In the last six years, his research projects on cyber security have been funded by DARPA. He is a handling editor of the Journal of Microprocessors and Microsystems and an editor of IEEE IT Professional. He is a senior member of the IEEE.