Domain Generalization via Optimal Transport with Metric Similarity Learning

Fan Zhou[a], Zhuqing Jiang[b], Changjian Shui[c], Boyu Wang[d,e], Brahim Chaib-draa[a,*]

[a]*Department of Computer Science and Software Engineering, Laval University, QC, Canada*
[b]*School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China*
[c]*Department of Electrical and Computer Engineering, Laval University, QC, Canada*
[d]*Department of Computer Science, University of Western Ontario, ON, Canada*
[e]*Vector Insitute, ON, Canada*

Abstract

Generalizing knowledge to unseen domains, where data and labels are unavailable, is crucial for machine learning models. We tackle the domain generalization problem to learn from multiple source domains and generalize to a target domain with unknown statistics. The crucial idea is to extract the underlying invariant features across all the domains. Previous domain generalization approaches mainly focused on learning invariant features and stacking the learned features from each source domain to generalize to a new target domain while ignoring the label information, which will lead to indistinguishable features with an ambiguous classification boundary. For this, one possible solution is to constrain the label-similarity when extracting the invariant features and to take advantage of the label similarities for class-specific cohesion and separation of features across domains. Therefore we adopt optimal transport with Wasserstein distance, which could constrain the class label similarity, for adversarial training and also further deploy a metric learning objective to leverage the label information for achieving distinguishable classification boundary. Empirical results show that our proposed method could outperform most of the baselines. Furthermore, ablation studies also demonstrate the effectiveness of each component of our method.

Keywords: Domain Generalization, Adversarial Learning, Metric Learning

*Corresponding author
Email address: brahim.chaib-draa@ift.ulaval.ca (Brahim Chaib-draa)

1. Introduction

Recent years witness a rapid development of machine learning and its succeeded applications such as computer vision (Ma et al., 2018a; Zhu et al., 2019; Ma et al., 2019b), natural language processing (Ma et al., 2013, 2018b) and cross-modalities learning (Zhu et al., 2019; Xu et al., 2018) with many real-world applications (Xie et al., 2018; Ma et al., 2019a). Traditional machine learning methods are typically based on the assumption that training and testing datasets are from the same distribution. However, in many real-world applications, this assumption may not hold, and the performance could degrade rapidly if the trained models are deployed to domains different from the training dataset (Ganin et al., 2016). More severely, to train a high-performance vision system requires a large amount of labelled data, and getting such labels may be expensive. Taking a pre-trained robotic vision system as an example, during each deployment task, the robot itself (*e.g.* position and angle), the environment (*e.g.* weather and illumination) and the camera (*e.g.* resolution) may result in different image styles. The cost to annotate enough data for each deployment task could be very expensive.

This kind of problem has been widely addressed by transfer learning (TL) (Zhuang et al., 2019) and domain adaptation (DA) (Ganin et al., 2016). In DA, a learner usually has access to the labelled source data and unlabelled target data, and it is typically trained to align the feature distribution between the source and target domain. However, sometimes, we could not expect the target data is accessible for the learner. In the robot example, the distribution divergences (different image styles) from training to testing domain can only be identified after the model is trained and deployed. In this scenario, it's unrealistic to collect samples before deployment. This would require a robot to have abilities to handle domain divergences even though the target data is absent.

We tackle this kind of problem under domain generalization (DG) paradigm, under which the learner has access to many source domains (data and corresponding labels), and aims at generalizing to the new (target) domain, where both data and labels are unknown. The goal of DG is to learn a prediction model on training data from the seen source domains so that it can generalize well on the unseen target domain. An

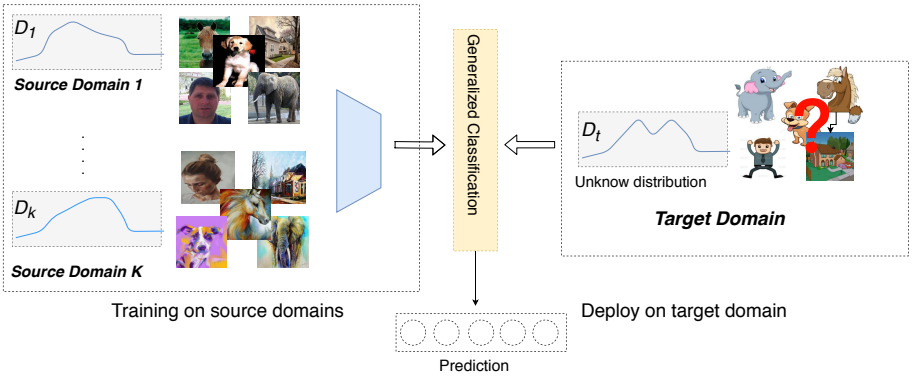


Figure 1: Domain Generalization: A learner faces a set labelled data from several source domains, and it aims at extracting invariant features across the seen source domains and learn to generalize to an unseen domain. Based on the manifold assumption (Goldberg et al., 2009), each domain i is supported by distribution $\mathcal{D}_i$. The learner can measure the source domain distribution via the source datasets but has no information on the unseen target distribution. After training on the source domains, the model is then deployed to a new domain $\mathcal{D}_t$ for prediction.

underlying assumption behind domain generalization is that there exists a common feature space underlying the multiple known source domains and unseen target domain. Specifically, we want to learn domain invariant features across these source domains, and then generalize to a new domain. An example of how domain generalization is processed is illustrated in Fig.1.

A critical problem in DG and DA involves aligning the domain distributions, which typically are achieved by extracting such representations. Previous DA works usually tried to minimize the domain discrepancies, such as KL-divergence and Maximum Mean Discrepancy (MMD) etc. via adversarial training, to achieve domain distribution alignments. Due to the similar problem setting between DA and DG, many previous approaches directly adopt the same adversarial training technique for DG. For example, a MMD metric is adopted by Li et al. (2018b) as a cross-domain regularizer and KL divergence is adopted to measure the domain shift by Li et al. (2017a) for domain generalization problem. The MMD metric is usually implemented in kernel space, which is not sufficient for large-scaled applications, and KL divergence is unbounded, which is also insufficient for a successful measuring domain shift (Zhao et al., 2019).

Besides, previous domain generalization approaches (Ilse et al., 2019; Ghifary et al., 2015; Li et al., 2018c; D'Innocente and Caputo, 2018; Volpi et al., 2018) mainly fo-

cused on applying similar DA technique to extract the invariant features and how to stack the learned features from each domain for generalizing to a new domain. These methods usually ignore the label information and will sometimes make the features became indistinguishable with ambiguous classification boundaries, $a.k.a$ semantic misalignment problem (Deng et al., 2020). A successful generalization should guide the learner not only to align the feature distributions between each domain but also to discriminate the samples in the same class could lie close to each other while samples from different classes could stay apart from each other, $a.k.a.$ feature compactness (Kamnitsas et al., 2018).

Aiming to solve this, we adopt Optimal Transport (OT) with Wasserstein distance to align the feature distribution for domain generalization since it could constrain labelled source samples of the same class to remain close during the transportation process (Courty et al., 2016). Moreover, some information theoretical metrics such as KL divergence is not capable to measure the inherent geometric relations among the different domains (Arjovsky et al., 2017). In contrast, OT can exactly measure their corresponding geometry properties. Besides, compared with (Ben-David et al., 2010), OT benefits from the advantages of Wasserstein distance by its gradient property (Arjovsky et al., 2017) and the promising generalization bound (Redko et al., 2017). The empirical studies (Gulrajani et al., 2017; Shen et al., 2018) also demonstrated the effectiveness of OT for extracting the invariant features to align the marginal distributions of different domains.

Furthermore, although the optimal transport process could constrain the labelled samples of the same class to stay close to each other, our preliminary results showed that just implementing optimal transport for domain generalization is not sufficient for a cohesion and separable classification boundary. The model could still suffer from indistinguishable features (see Fig. 4c). In order to train the model to predict well on all the domains, this separable classification boundary should also be achieved under a domain-agnostic manner. That is, for a pair of instances, no matter which domain they come from, they should stay close to each other if they are in the same class and vice-versa. To this end, we further promote metric learning as an auxiliary objective for leveraging the source domain label information for a domain-independent distin-

guishable classification boundary.

To summarize, we deployed the optimal transport technique with Wasserstein distance for domain generalization for extracting the domain invariant features. To avoid ambiguous classification boundary, we proposed to implement metric learning strategies to achieve a distinguishable feature space. Therefore, we proposed the Wasserstein Adversarial Domain Generalization (*WADG*) algorithm.

In order to check the effectiveness of the proposed approach, we tested the algorithm on two benchmarks comparing with some recent domain generalization baselines. The experiment results showed that our proposed algorithm could outperform most of the baselines, which confirms the effectiveness of our proposed algorithm. Furthermore, the ablation studies also demonstrated the contributions of our algorithm.

2. Related Works

2.1. Domain Generalization

The goal of DG is to learn a model that can extract common knowledge that is shared across source domains and generalize well on the target domain. Compare with DA, the main challenge of DG is that the target domain data is not available during the learning process.

A common framework for DG is to extract the most informative and transferable underlying common features from source instances generated from different distributions and to generalize to unseen one. This kind of approach holds with the assumption that there exists an underlying invariant feature distribution among all domains, and that consequently such invariant features can generalize well to a target domain. Muandet et al. (2013) implemented MMD as a distribution regularizer and proposed the kernel-based *Domain Invariant Component Analysis* (DICA) algorithm. An autoencoder-based model was proposed by (Ghifary et al., 2015) under a multi-task learning setting to learn domain-invariant features via adversarial training. (Li et al., 2018c) proposed an end-to-end deep domain generalization approach by leveraging deep neural networks for domain-invariant representation learning. (Motiian et al., 2017) proposed to minimize the semantic alignment loss as well as the separation loss based on deep

learning models. (Li et al., 2018b) proposed a low-rank Convolutional Neural Network model based on domain shift-robust deep learning methods.

There are also some approaches to tackle the domain generalization problems in a meta-learning manner. To the best of our knowledge, (Li et al., 2018a) first proposed to adopt the Meta Agnostic Meta-Learning (MAML) (Finn et al., 2017) which back-propagates the gradients of ordinary loss function of meta-test tasks. As pointed by (Dou et al., 2019), such an approach might lead to a sub-optimal solution, as it is highly abstracted from the feature representations. (Balaji et al., 2018) proposed *MetaReg* algorithm in which a regularization function ($e.g.$ weighted L_1 loss) is implemented for the classification layer of the model but not for the feature extractor layers. Then, (Li et al., 2019) proposes an auxiliary meta loss which is gained based on the feature extractor. Furthermore, the network architecture of (Li et al., 2019) is the widely used feature-critic style model based on a similar model from domain adversarial training technique (Ganin et al., 2016). Dou et al. (2019) and Matsuura and Harada (2020) also started to implement clustering techniques on the invariant feature space for better classification and showed better performance on the target domain.

2.2. Metric Learning

Metric learning aims to learn a discriminative feature embedding where similar samples are closer while different samples are further apart (Deng et al., 2020). Hadsell et al. (2006) proposed the *siamese network* together with *contrastive loss* to guide the instances stay close with each other in the feature space if they have the same labels and push them apart vice-versa. Schroff et al. (2015) proposed the *triplet loss* aiming to learn a feature space where a positive pair has higher similarity than the negative pair when comparing by the same anchor with a given margin. Oh Song et al. (2016) showed that neither the *contrastive loss* nor *triplet loss* could efficiently explore the full pair-wise relations between instances under the mini-batch training setting. They further propose the *lifted structure loss* to fully utilize pair-wise relations across batches. However, it only choose equal number of positive pairs as negative ones randomly, and many informative pairs are discarded (Wang et al., 2019), which restricts the ability of finding the informative pairs. Yi et al. (2014) proposed the binomial deviance

loss which could measure the hard pairs. One remarkable work by Wang et al. (2019) combines the advantages both from *lifted structure loss* and *binomial loss* to leverage the pair-similarity. They proposed to leverage not only pair-similarities (positive or negative pairs with each other) but also self-similarity which enables the learner to collect and weight informative pairs (positive or negative pairs) under an iterative (mining and weighting) manner. For a pair of instances, the self-similarity is gained from itself. Such a multi-similarity has been shown could measure the similarity and could cluster the samplers more efficiently and accurately. In the context of domain generalization, Dou et al. (2019) proposed to guide the learner to leverage from the local similarity in the semantic feature space, in which the authors argued may contain essential domain-independent *general knowledge* for domain generalization and adopt the constrative loss and triplet loss to encourage the clustering for solving this issue. Leveraging from the across-domain class similarity information can encourage the learner to extract robust semantic features that regardless of domains, which is an useful auxiliary information for the learner. If the learner could not separate the samples (from different source domains) with domain-independent class-specific cohesion and separation on the domain invariant feature space, it would still suffer from ambiguous decision boundaries. This ambiguous decision boundaries might still be sensitive to the unseen target domain. Matsuura and Harada (2020) implement unsupervised clustering on source domains and showed better classification performance. Our work is orthogonal to previous works, proposing to enforce more distinguishable invariant features space via Wasserstein adversarial training and encouraging to leverage from label similarity information for better classification boundary.

3. Preliminaries and Problem Setup

We start by introducing some preliminaries. In order to better summarize the notations symbols in this work, we provide the list of notations and symbols in Table 1.

3.1. Notations and Definitions

Following Redko et al. (2017) and Li et al. (2017a), suppose we have m known source domains distributions $\{\mathcal{D}_i\}_{i=1}^{m}$, and i^{th} domain contains N_i labeled instances in total,

Table 1: List of notations

Symbol	Meaning	Symbol	Meaning
F	The feature extraction function	θ_f	Parameter of feature extraction network
D	The critic function	θ_d	Parameter of critic network
C	The classification function	θ_c	Parameter for classification network
m	The number of source domains	$\mathbf{x}_j^{(i)}$	The i-th instance from the j-th domain
N_i	The number of instances in the i-th domain	$\mathbf{X}^{(i)}$	The set of instances in the i-th domain $\mathbf{X}^{(i)} = \{\mathbf{x}_j^{(i)}\}_{j=1}^{N}$
$\mathcal{D}$	The data distribution. $\mathcal{D}_i$ are the source domain distributions	$Z^{(i)}$	The extracted feature from domain i
$W_1(\mathcal{D}_i, \mathcal{D}_j)$	Wasserstein-1 distance over two distributions $\mathcal{D}_i$ and $\mathcal{D}_j$	y_j	The label for corresponding instance x_j
$\mathbf{S}$	The similarity matrix	$S_{i,j}$	The value of i-th row and j-th column of the similarity matrix $\mathbf{S}$
$w_{i,j}$	The weight for similarity $S_{i,j}$	ϵ	Small margin for roughly select the positive and negative pairs
α	Fixed parameter for positive mining	β	Fixed parameter for negative mining
λ	Parameter for self-similarity mining	λ_d	Coefficient for regularizing the adversarial objective
λ_s	Coefficient for regularizing the metric learning objective	$\mathcal{L}$	The objective functions, $\mathcal{L}_C$ is the classification loss, $\mathcal{L}_D$ is the adversarial loss, $\mathcal{L}_{MS}$ is the metric similarity loss

denoted by $\{(\mathbf{x}_j^{(i)}, y_j^{(i)})\}_{j=1}^{N_i}$, where $\mathbf{x}_j^{(i)} \in \mathbb{R}^n$ is the j^{th} instance feature from the i^{th} domain and $y_j^{(i)} \in \{1, \ldots, K\}$ are the corresponding labels. For a hypothesis class $\mathcal{H}$, the expected source and target risk of a hypothesis $h \in \mathcal{H}$ over domain distribution $\mathcal{D}_i$ is the probabilities that h wrongly predicts on the entire distribution $\mathcal{D}_i$: $\epsilon_i(h) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}\ell(h(\mathbf{x}, y))$, where $\ell(\cdot)$ is the loss function. The empirical loss is also defined by: $\hat{\epsilon}_i(h) = \frac{1}{N_i}\sum_{j=1}^{N_i}\ell(h(\mathbf{x}_j, y_j))$.

In the setting of domain generalization, we only have the access to the seen source domains $\mathcal{D}_i$ but have no information about the target domain. The learner is expected to extract the underlying invariant feature space across the source domains and generalize to a new target domain.

3.2. Optimal Transport and Wasserstein Distance

We follow Redko et al. (2017) and define $c : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ as the cost function for transporting one unit of mass $\mathbf{x}$ to $\mathbf{x}'$, then the primal form of the Wasserstein distance

between $\mathcal{D}_i$ and $\mathcal{D}_j$ could be computed by,

$$W_p^p(\mathcal{D}_i, \mathcal{D}_j) = \inf_{\gamma \in \Pi(\mathcal{D}_i, \mathcal{D}_j)} \int_{\mathbb{R}^n \times \mathbb{R}^n} c(\mathbf{x}, \mathbf{x}')^p d\gamma(\mathbf{x}, \mathbf{x}') \tag{1}$$

where $\Pi(\mathcal{D}_i, \mathcal{D}_j)$ is the probability coupling on $\mathbb{R}^n \times \mathbb{R}^n$ with marginals $\mathcal{D}_i$ and $\mathcal{D}_j$ referring to all the possible coupling functions. Throughout this paper, we adopt Wasserstein-1 distance only ($p = 1$).

Computing the primal form of Wasserstein distance (Eq. 1) is computational inefficiently. Assuming $|\mathcal{D}_i| = n, |\mathcal{D}_j| = m$, the time complexity for directly computing Eq. 1 is $\mathcal{O}(n^3 + m^3)$. On the contrary, leveraging the *Kantorovich-Rubinstein duality* (Wainwright, 2019) of Wasserstein distances could help to get a more efficient approximation. Assume f a 1-Lipschitz-continuous $w.r.t.$ the cost function: $\|f(x) - f(x')\| \leq c(x, x')$, we can prove that for any function f,

$$W_1(\mathcal{D}_i, \mathcal{D}_j) \geq \mathbb{E}_{x \sim \mathcal{D}_i} d(x) - \mathbb{E}_{x' \sim \mathcal{D}_j} d(x')$$

The equality arrives when f reaches the maximum of the right side,

$$W_1(\mathcal{D}_i, \mathcal{D}_j) = \sup_{\|f\|_L < 1} \mathbb{E}_{x \in \mathcal{D}_i} f(x) - \mathbb{E}_{x' \in \mathcal{D}_j} f(x') \tag{2}$$

In practice, such a function f could be approximated by a neural-network, which allows us to compute this Kantorovich-Rubinstein duality efficiently by computing the expectation and the complexity $w.r.t.$ $f(x)$ is only $\mathcal{O}(n+m)$. Empirically, to compute the $\sup$ is equivalent to find out the maximum of W_1 (by an $\arg\max$ operation). General neural network optimizer ($e.g.$ SGD or Adam) can efficiently solve the maximum problem to evaluate the dual value of W_1 distance.

Optimal transport theory and Wasserstein distance were recently investigated in the context of machine learning (Arjovsky et al., 2017) especially in the domain adaptation area (Courty et al., 2016; Zhou et al., 2020). The general idea of implementing the optimal transport technique for domain generalization across domains is illustrated in Fig. 2. To learn domain invariant features, OT technique is implemented to achieve domain alignments for extracting invariant features. After the OT transition, the invariant

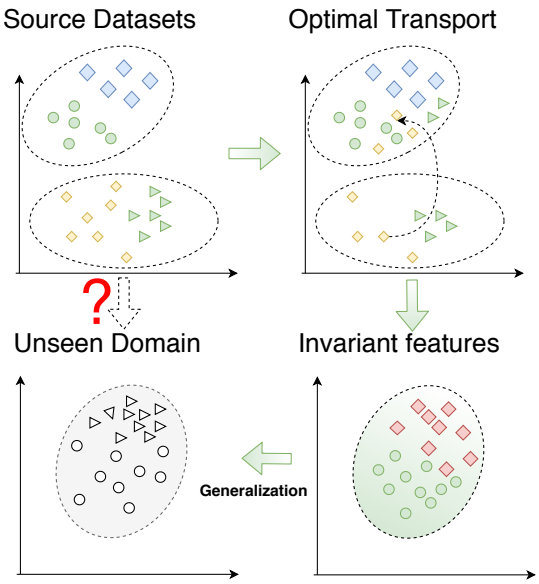


Figure 2: Use optimal transport (OT) for domain generalization: Typically to directly predict on the unseen domain (the white dashed arrow) is difficult. In order to learn domain invariant features, as showed in the direction of the green arrow we adopted the OT technique to achieve domain alignments for extracting invariant features. After the OT transition, the invariant features can be generalized to unseen domain.

features can be generalized to unseen domain.

3.3. Metric Learning

For a pair of instances $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$, the notion of *positive pairs* usually refers to the condition where pair i, j have same labels ($y_i = y_j$), while the negative pairs usually refers to the condition $y_i \neq y_j$. The central idea of metric learning is to encourage a pair of instances who have the same labels to be closer, and push negative pairs to be apart from each other (Wu et al., 2017).

Follow the framework of Wang et al. (2019), we show the general pair-weighting process of metric learning. Assuming the feature extractor f parameterized by $\boldsymbol{\theta}_f$ projects the instance $\mathbf{x} \in \mathbb{R}^n$ to a d-dimensional normalized space: $f(\mathbf{x}; \boldsymbol{\theta}_f) : \mathbb{R}^n \rightarrow [0, 1]^d$. Then, for two samples $\mathbf{x}_i$ and $\mathbf{x}_j$, the similarity between them could be defined as the inner product of the corresponding feature vector:

$$S_{i,j} := \langle f(\mathbf{x}_i; \boldsymbol{\theta}_f), f(\mathbf{x}_j; \boldsymbol{\theta}_f) \rangle \tag{3}$$

To leverage the across-domain class similarity information can encourage the learner to extract the classification boundary that regardless of domains, which is an useful auxiliary information for the learner. We further elaborate it in section 4.2.

4. Proposed Method

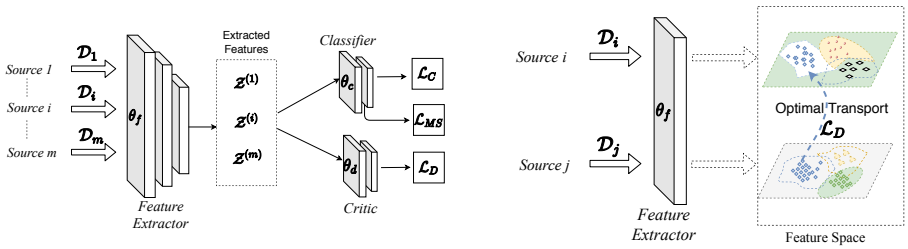
The high-level idea of WADG algorithm is to learn a domain-invariant feature space and domain-agnostic classification boundary. Firstly, we align the marginal distribution of different source domains via optimal transport by minimizing the Wasserstein distance to achieve the domain-invariant feature space. And then, we adopt metric learning objective to guide the learner to leverage the class similarity information for a better classification boundary. A general workflow of our method is illustrated in Fig. 3a. The model contains three major parts: a feature extractor, a classifier and a critic function.

The feature extractor function F, parameterized by $\boldsymbol{\theta}_f$, extracts the features from different source domain. For set of instances $\mathbf{X}^{(i)} = \{\mathbf{x}_j^{(i)}\}_{j=1}^{N_i}$ from domain $\mathcal{D}_i$, we can then denote the extracted feature from domain i as $\mathbf{Z}^{(i)} = F(\mathbf{X}^{(i)})$. The classification function C, parameterized by $\boldsymbol{\theta}_c$, is expected to learn to predict labels of instances from all the domains correctly. The critic function D, parameterized by $\boldsymbol{\theta}_d$, aims to measure the empirical Wasserstein distance between features from a pair of source domains. For the target domain, all the instances and labels are absent during the training time.

WADG aims to learn the domain-agnostic features with distinguishable classification boundary. During each train round, the network receives the labelled data from all domains and train the classifier under a supervised mode with the classification loss $\mathcal{L}_C$. For the classification process, we use the typical cross-entropy loss for all m source domains:

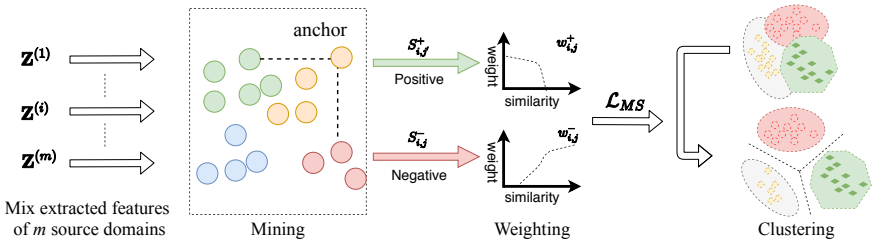
$$\mathcal{L}_C = -\sum_{i=1}^{m}\sum_{j=1}^{N_i} y_j \log(\mathbb{P}(C(F(\mathbf{x}_j^{(i)})))) \tag{4}$$

Through this, the model could learn to train the category information on over all the domains. The feature extractor F is then trained to minimize the estimated Wasserstein Distance in an adversarial manner with the critic D with an objective $\mathcal{L}_D$. We then



(a) The whole workflow the proposed WADG model.

(b) Optimal Transport for Feature Alignment.



(c) Metric Learning for Clustering Proces

Figure 3: The proposed WADG method. (a): the general workflow of WADG method. The model mainly consists of three parts, the feature extractor, classifier and critic function. During training, the model receives all the source domains. The feature extractor is trained to learn invariant features together with the critic function in an adversarial manner. (b): For each pair of source domains $\mathcal{D}_i$ and $\mathcal{D}_j$, optimal transport process for aligning the features from different domains. (c): The metric learning process. For a batch of all source domain instances, we first roughly mining the positive and negative pairs via Eq. 7. Then, compute the corresponding weights via Eq. 11 and Eq. 12 to compute $\mathcal{L}_{MS}$ to guide the clustering process.

adopt a metric learning objective (namely, $\mathcal{L}_{MS}$) for leveraging the similarities for a better classification boundary. Our full method then solve the joint loss function,

$$\mathcal{L} = \arg\min_{\theta_f, \theta_c} \max_{\theta_d} \mathcal{L}_C + \mathcal{L}_D + \mathcal{L}_{MS},$$

where $\mathcal{L}_D$ is the adversarial objective function, and $\mathcal{L}_{MS}$ is the metric learning objective function. In the sequel, we will elaborate these two objectives in section 4.1 and section 4.2, respectively.

4.1. Adversarial Domain Generalization via Optimal Transport

As optimal transport could constrain labelled source samples of the same class to remain close during the transportation process (Courty et al., 2016). We deploy optimal transport with Wasserstein distance (Redko et al., 2017; Shen et al., 2018) for aligning the marginal feature distribution over all the source domains.

A brief workflow of the optimal transport for a pair of sourcce domains is illustrated in Fig. 3b. The critic function D estimates the empirical Wasserstein Distance between the each source domain through a pair of instances from the empirical sets $\mathbf{x}^{(i)} \in \mathbf{X}^{(i)}$ and $\mathbf{x}^{(j)} \in \mathbf{X}^{(j)}$. In practice (Shen et al., 2018), the dual term Eq. 2 of Wasserstein distance could be computed by,

$$W_1(\mathbf{X}^{(i)}, \mathbf{X}^{(j)}) = \frac{1}{N_i} \sum_{\mathbf{x}^{(i)} \in \mathbf{X}^{(i)}} D(F(\mathbf{x}^{(i)})) - \frac{1}{N_j} \sum_{\mathbf{x}^{(j)} \in \mathbf{X}^{(j)}} D(F(\mathbf{x}^{(j)})) \tag{5}$$

As in domain generalization setting, there usually exists more that two source domains, we can sum all the empirical Wasserstein distance between each pair of source domains,

$$\mathcal{L}_D = \sum_{i=1}^{m} \sum_{j=i+1}^{m} \big[\frac{1}{N_i} \sum_{\mathbf{x}^{(i)} \in \mathbf{X}^{(i)}} D(F(\mathbf{x}^{(i)})) - \frac{1}{N_j} \sum_{\mathbf{x}^{(j)} \in \mathbf{X}^{(j)}} D(F(\mathbf{x}^{(j)})) \big] \tag{6}$$

Throughout this pair-wise optimal transport process, the learner could extract a domain-invariant feature space, we then propose to apply metric learning approaches to leverage the class label similarity for domain independent clustering feature extraction. We then introduce the metric learning for domain agnostic clustering in the next section.

4.2. Metric Learning for Domain Agnostic Classification Boundary

As aforementioned, only aligning the marginal features via adversarial training is not sufficient for DG since there may exist a ambiguous decision boundary (Dou et al., 2019). When predicting on the target domain, the learner may still suffer from this ambiguous decision boundary. To solve this, we propose to implement the metric learning techniques to help cluster the instances and promote a better prediction boundary for better generalization.

To solve this, except to the supervised source classification and alignment of the marginal distribution across domains with the Wasserstein adversarial training defined above, we then further encourage robust domain-independent local clustering via leverage from label information using the metric learning objective. The brief workflow is illustrated in Fig. 3c. Specifically, we adopt the metric learning objective to require the images

regardless of their domains to follow the two aspects: 1) images from the same class are semantically similar, thereby should be mapped nearby in the embedding space (semantic clustering), while 2) instances from different classes should be mapped apart from each other in embedding space. Since goal of domain generalization aims to learn to hypothesis could predict well on all the domains, the clustering should also be achieved under a domain-agnostic manner.

To this end, we mix the instances from all the source domains together and encourage the clustering for domain agnostic features via the metric learning techniques to achieve a domain-independent clustering decision boundary. For this, during each training iteration, for a batch $\{\mathbf{x}_1^{(i)}, y_1^{(i)}, \ldots, \mathbf{x}_b^{(i)}, y_b^{(i)}\}_{i=1}^m$ from m source domains with batch size b, we mix all the instances from each domain and denoted by $\{(\mathbf{x}_i^B, y_i^B)\}_{i=1}^{m'}$ with total size m'. We first measure the relative similarity between the negative and positive pairs, which is introduced in the next sub-section.

4.2.1. Pair Similarity Mining

Assume $\mathbf{x}_i^B$ is an anchor, a negative pair $\{\mathbf{x}_i^B, \mathbf{x}_j^B\}$ and a positive pair $\{\mathbf{x}_i^B, \mathbf{x}_{j'}^B\}$ are selected if S_{ij} and $S_{i,j'}$ satisfy the negative condition $S_{i,j}^-$ and the positive condition $S_{i,j}^+$, respectively :

$$S_{i,j}^- \geq \min_{y_i = y_k} S_{i,k} - \epsilon, \quad S_{i,j'}^+ \leq \min_{y_i \neq y_k} S_{i,k} + \epsilon \tag{7}$$

where ϵ is a given margin. Through Eq. 7 and specific margin ϵ, we will have a set of negative pairs $\mathcal{N}$ and a set of positive pairs $\mathcal{P}$. This process (Eq. 7) could roughly cluster the instances with each anchor by selecting informative pairs (inside of the margin), and discard the less informative ones (outside of the margin).

With such roughly selected informative pairs $\mathcal{N}$ and $\mathcal{P}$, we then assign the instance with different weights. Intuitively, if a instance has higher similarity with an anchor, then it should stay closer with the anchor and vice-versa. We introduce the weighting process in the next section.

4.2.2. Pair Weighting

For instances of positive pairs, if they are more similar with the anchor, then it should have higher weights while give the negative pairs with lower weights if they are more dissimilar, no matter which domain they come from. Through this process, we can push the instances into several groups via measure their similarities.

For N instances, computing the similarity between each pair could result in a similarity matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$. For a loss function based on pair similarity, it can usually be defined by $\mathcal{F}(\mathbf{S}, y)$. Let $S_{i,j}$ be the i^{th} row, j^{th} column element of matrix $\mathbf{S}$. The gradient $w.r.t$ the network could be computed by,

$$\frac{\partial \mathcal{F}(\mathbf{S}, y)}{\partial \boldsymbol{\theta}_f} = \frac{\partial \mathcal{F}(\mathbf{S}, y)}{\partial \mathbf{S}} \frac{\partial \mathbf{S}}{\partial \boldsymbol{\theta}_f} = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial \mathcal{F}(\mathbf{S}, y)}{\partial S_{i,j}} \frac{\partial S_{i,j}}{\partial \boldsymbol{\theta}_f} \tag{8}$$

Eq. 8 could be reformulated into a new loss function $\mathcal{L}_{MS}$ as,

$$\mathcal{L}_{MS} = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial \mathcal{F}(\mathbf{S}, y)}{\partial S_{i,j}} S_{i,j} \tag{9}$$

usually the metric loss defined $w.r.t$ similarity matrix $\mathbf{S}$ and label y could be reformulated by Eq. 9. The term $\frac{\partial \mathcal{F}(\mathbf{S}, y)}{\partial S_{i,j}}$ in Eq. 9 could be treated as an constant scalar since it doesn't contain the gradient of $\mathcal{L}_{MS}$ $w.r.t$ $\boldsymbol{\theta}_f$. Then, we just need to compute the gradient term $\frac{\partial \mathcal{F}_{i,j}}{\partial \boldsymbol{\theta}_f}$ for the positive and negative pairs. Since the goal is to encourage the positive pairs to be closer, then we can assume the gradient ≤ 0, $i.e.$, $\frac{\partial \mathcal{F}_{i,j}}{\partial \boldsymbol{\theta}_f} \leq 0$. Conversely, for a negative pair, we could assume $\frac{\partial \mathcal{F}_{i,j}}{\partial \boldsymbol{\theta}_f} \geq 0$. Thus, Eq. 9 is transformed by the summation over all the positive pair ($y_i = y_j$) and negative pairs ($y_i \neq y_j$),

$$\begin{aligned}
\mathcal{L}_{MS} &= \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial \mathcal{F}(\mathbf{S}, y)}{\partial S_{i,j}} S_{i,j} \\
&= \sum_{i=1}^{N} \left(\sum_{j=1, y_j \neq y_i}^{N} \frac{\partial \mathcal{F}(\mathbf{S}, y)}{\partial S_{i,j}} S_{i,j} + \sum_{j=1, y_j = y_i}^{N} \frac{\partial \mathcal{F}(\mathbf{S}, y)}{\partial S_{i,j}} S_{i,j} \right) \\
&= \sum_{i=1}^{N} \left(\sum_{j=1, y_j \neq y_i}^{N} w_{i,j} S_{i,j} - \sum_{j=1, y_j = y_i}^{N} w_{i,j} S_{i,j} \right)
\end{aligned} \tag{10}$$

where $w_{i,j} = \left|\frac{\partial S_{i,j}}{\partial \boldsymbol{\theta}_f}\right|$ is regarded as the weight for similarity $S_{i,j}$. For each pair of instances i, j, we could assign different weights according to their similarities $S_{i,j}$. Then $w^+_{i,j}$ or $w^-_{i,j}$ could be defined as the weight of a positive or negative pairs' similarity, respectively. Previously, Yi et al. (2014) and Wang et al. (2019) applied a soft function for measuring the similarity. We then consider the similarity of the pair itself ($i.e.$ self-similarity), the negative similarity and the positive similarity. The weight of self-similarity could be measured by $\exp(S_{i,j} - \lambda)$ with a small threshold λ. For a selected negative pair $\{\mathbf{x}^B_i, \mathbf{x}^B_j\} \in \mathcal{N}$ the corresponding weight (see Eq. 10) could be defined by the soft function of self-similarity together with the negative similarity:

$$\begin{aligned} w^-_{i,j} &= \frac{1}{\exp(\beta(\lambda - S_{ij})) + \sum_{k\in\mathcal{N}} \exp(\beta(S_{i,k} - \lambda))} \\ &= \frac{\exp(\beta(S_{ij} - \lambda))}{1 + \sum_{k\in\mathcal{N}} \exp(\beta(S_{ik} - \lambda))} \end{aligned} \tag{11}$$

Similarly, the weight of a positive pair $\{\mathbf{x}^B_i, \mathbf{x}^B_j\} \in \mathcal{P}$ is defined by,

$$w^+_{i,j} = \frac{1}{\exp(-\alpha(\lambda - S_{i,j})) + \sum_{k\in\mathcal{P}} \exp(-\alpha(S_{i,k} - S_{i,j}))} \tag{12}$$

Then, take Eq. 11 and Eq. 12 into Eq. 10, and integrate Eq. 10 with the similarity mining $S_{i,j}$, we have the objective function for clustering,

$$\mathcal{L}_{MS} = \frac{1}{m}\sum_{i=1}^{m}\Big\{\frac{1}{\alpha}\log[1 + \sum_{k\in\mathcal{P}_i}\exp(-\alpha(S_{ik} - \lambda))] + \frac{1}{\beta}\log[1 + \sum_{k\in\mathcal{N}_i}\exp(\beta(S_{ik} - \lambda))]\Big\} \tag{13}$$

where λ, α and β are fixed hyper-parameters, we elaborate them in the empirical setting section 5.2. Then, the whole objective of our proposed method is,

$$\mathcal{L} = \arg\min_{\theta_f,\theta_c}\max_{\theta_d} \mathcal{L}_C + \lambda_d\mathcal{L}_D + \lambda_s\mathcal{L}_{MS} \tag{14}$$

where λ_d and λ_s are coefficients to regularize $\mathcal{L}_d$ and $\mathcal{L}_{MS}$ respectively.

Based on these above, we propose the WADG algorithm in Algorithm 1. And we show the empirical results in the next section.

Algorithm 1 The proposed WADG algorithm (one round)

Require: Samples from different source domains $\{\mathcal{D}_i\}_{i=1}^{M}$
Ensure: Neural network parameters $\boldsymbol{\theta}_f$, $\boldsymbol{\theta}_c$, $\boldsymbol{\theta}_d$

1: **for** mini-batch of samples $\{(\mathbf{x}_s^{(i)}, y_s^{(i)})\}$ from source domains **do**
2: Compute the classification loss $\mathcal{L}_C$ over all the domains according to Eq. 4
3: Compute the Wasserstein distance $\mathcal{L}_D$ between each pair of source domains according to Eq. 6
4: Mix the pairs from different domains and compute the similarity by Eq. 3
5: Roughly select the positive and negative pairs by solving Eq. 7
6: Compute similarity loss $\mathcal{L}_{MS}$ on all the source instances by Eq. 13
7: Update θ_f, θ_c and 1_d by solving Eq. 14 with learning rate η:

$$\boldsymbol{\theta}_f \leftarrow \boldsymbol{\theta}_f - \eta \frac{\partial(\mathcal{L}_C + \lambda_d \mathcal{L}_D + \lambda_s \mathcal{L}_{MS})}{\partial \boldsymbol{\theta}_f},$$
$$\boldsymbol{\theta}_c \leftarrow \boldsymbol{\theta}_c - \eta \frac{\partial(\mathcal{L}_C + \lambda_d \mathcal{L}_D + \lambda_s \mathcal{L}_{MS})}{\partial \boldsymbol{\theta}_c},$$
$$\boldsymbol{\theta}_d \leftarrow \boldsymbol{\theta}_d + \eta \frac{\partial \mathcal{L}_D}{\partial \boldsymbol{\theta}_d}$$

8: **end for**
9: Return the optimal parameters $\boldsymbol{\theta}_f^\star$, $\boldsymbol{\theta}_c^\star$ and $\boldsymbol{\theta}_d^\star$

5. Experiments and Results

5.1. Datasets

In order to evaluate our proposed approach, we implement experiments on three common used datasest: **VLCS** (Torralba and Efros, 2011), **PACS** (Li et al., 2017a) and **Office-home** (Venkateswara et al., 2017) dataset. The VLCS dataset contains images from 4 different domains: PASCAL VOC2007 (V), LabelMe (L), Caltech (C), and SUN09 (S). Each domain includes five classes: *bird, car, chair, dog* and *person*. PACS dataset is a recent benchmark dataset for domain generalization. It consists of four domains: Photo (P), Art painting (A), Cartoon (C), Sketch (S), with objects from seven classes: dog, elephant, giraffe, guitar, house, horse, person. Office-Home is a more challenging dataset, which contains four different domains: *Art* (Ar), *Clipart* (Cl), *Product* (Pr) and *Real World* (Rw), with 65 categories in each domain. Previous work showed that matter the adversarial model is trained under supervised Long et al. (2017), semi-supervised Zhou et al. (2020) or unsupervised Long et al. (2018) way, the model

will suffer from learning the diverse feature. To test our domain generalization model on this dataset could also help to affirm the effectiveness of our approach.

5.2. Baselines and Implementation details

To show the effectiveness of our proposed approach, we compare our algorithm on the benchmark datasets with the following recent domain generalization methods.

- ***Deep All***: We follow the standard evaluation protocol of Domain Generalization to set up the pre-trained Alexnet or ResNet-18 fine-tuned on the aggregation of all source domains with only the classification loss.
- TF (Li et al., 2017b): A low-rank parameterized Convolution Neural Network model which aims to reduce the total number of model parameters for an end-to-end Domain Generalization training.
- CIDDG (Li et al., 2018c): Matches the conditional distribution by change the class prior.
- MLDG (Li et al., 2018a): The meta-learning approach for domain generalization. It runs the meta-optimization on simulated meta-train/ meta-test sets with domain shift
- CCSA (Motiian et al., 2017): The contrastive semantic alignment loss was adopted together with the source classification loss function for both the domain adaptation and domain generalization problem.
- MMD-AAE (Li et al., 2018b): The Adversarial Autoencoder model was adopted together with the Mean-Max Discrepancy to extract a domain invariant feature for generalization.
- D-SAM (D'Innocente and Caputo, 2018): It aggregates domain-specific modules and merges general and specific information together for generalization.
- JiGen (Carlucci et al., 2019): It achieves domain generalization by solving the Jigsaw puzzle via the unsupervised task.

Table 2: The hyper-parameter values for experiments

Hyper-parameters	**Value**	**Hyper-parameters**	**Value**
learning rate	PACS: 5×10^{-4}	λ	1.0
	Office-home: 2×10^{-4}	α	2.0
λ_d	$\lambda_d = \frac{2}{1+\exp(-10p)} - 1$	β	40.0
λ_s	$[1e-4, 1e-5]$	ϵ	0.1

- MASF (Dou et al., 2019): A meta-learning style method which based on MLDG and combined with Consitrastive Loss/ Triplet Loss to encourage domain-independent semantic feature space.

- MMLD (Matsuura and Harada, 2020): An approach that mixes all the source domains by assigning a pseudo domain label for extract domain-independent cluster feature space.

Following the general evaluation protocol of domain generalization ($e.g.$ Dou et al. (2019); Matsuura and Harada (2020)), on PACS and VLCS dataset. We first test our algorithm on by using AlexNet (Krizhevsky et al., 2012) backbones by removing the last layer as feature extractor. For preparing the dataset, we follow the train/val./test split and the data pre-processing protocol of Matsuura and Harada (2020). As for the classifier, we initialize a three-layers MLP whose input has the same number of inputs as the feature extractor's output and to have the same number of outputs as the number of object categories (2048-256-256-K), where K is the number of classes. For the critic network, we also adopt a three-layers MLP (2048-1024-1024-1). For the metric learning objective, we use the output of the second layer of classifier network (with size 256) for computing the similarity.

In order to better demonstrating the hyper-parameters used in this work, we firstly summarized the value of hyper-parameters in Table. 2. The corresponding descriptions are provided in the following parts. We adopt the ADAM (Kingma and Ba, 2014) optimizer for training with learning rate ranging from 5×10^{-4} to 5×10^{-5} for the whole model together with mini-batch size 64.

Table 3: Empirical Results (accuracy %) on PACS dataset with pre-trained AlexNet as Feature Extractor. For each column, we refer the generalization taks as the target domain name. For example, the third column 'Cartoon' refers to the generalization tasks where domain *Cartoon* is the target domain while the model is trained on the rest three domains.

Method	Art	Cartoon	Sketch	Photo	Avg.
Deep All	63.30	63.13	54.07	87.70	67.05
TF(Li et al., 2017b)	62.86	66.97	57.51	89.50	59.21
CIDDG(Li et al., 2018c)	62.70	69.73	64.45	78.65	68.88
MLDG (Li et al., 2018a)	66.23	66.88	58.96	88.00	70.01
D-SAM(D'Innocente and Caputo, 2018)	63.87	70.70	64.66	85.55	71.20
JiGen(Carlucci et al., 2019)	67.63	71.71	65.18	89.00	73.38
MASF(Dou et al., 2019)	**70.35**	72.46	67.33	90.68	75.21
MMLD(Matsuura and Harada, 2020)	69.27	**72.83**	66.44	88.98	74.38
Ours	70.21	72.51	**70.32**	**89.81**	**75.71**

For stable training, we set coefficient $\lambda_d = \frac{2}{1+\exp(-10p)} - 1$ to regularize the adversarial loss, where p is the training progress, to regularize the adversarial loss. This regularization scheme λ_d has been widely used in adversarial training based domain adaptation and generalization setting ($e.g.$ (Long et al., 2017; Wen et al., 2019; Matsuura and Harada, 2020)) and have been proved could help to stabilize the training process. For the setting of λ_s, we follow the setting of (Dou et al., 2019) and set the value to 10^{-4}. In our preliminary validation results, the performance is not sensitive with $\lambda_d \in [0, 1]$. We also tried to range λ_s from 10^{-3} to 10^{-6} via reverse validation and didn't observe obvious differences.

For the hyper-parameters in $\mathcal{L}_{MS}$ (see Eq. 7 and Eq. 13), we empirically set $\lambda = 1.0$, $\epsilon = 0.1$, $\alpha = 2.0$, $\beta = 40.0$. Here ϵ is for roughly selecting the positive and negative pairs and λ is a small margin parameters. Previous work (Weinberger and Saul, 2009) has shown that choosing $\epsilon = 0.1$ and $\lambda = 1.0$ could be optimal performance for general metric learning problems, our preliminary validation results also showed that when $\lambda \in [0.5, 2.0]$, the performance didn't have too much difference in our domain generalization tasks. Besides, α and β are two parameters used for positive and negative mining referred by Ustinova and Lempitsky (2016) in which α was set to 2.0 and β was set to 50.0. In our paramilitary validation results, we found the setting of $\beta \in [30.0, 45.0]$ could guarantee stable performance in our domain generalization problems but rather 50.0 in the original (Ustinova and Lempitsky, 2016) and $\alpha \in [1.0, 5.0]$ could also

Table 4: Empirical Results (accuracy %) on VLCS dataset with pre-trained AlexNet as Feature Extractor.

Method	Caltech	LabelMe	Pascal	Sun	Avg.
Deep All	92.86	63.10	68.67	64.11	72.19
D-MATE (Ghifary et al., 2015)	89.05	60.13	63.90	61.33	68.60
CIDDG (Li et al., 2018c)	88.83	63.06	64.38	62.10	69.59
CCSA (Motiian et al., 2017)	92.30	62.10	67.10	59.10	70.15
SLRC (Ding and Fu, 2017)	92.76	62.34	65.25	63.54	70.97
TF (Li et al., 2017b)	93.63	63.49	69.99	61.32	72.11
MMD-AAE (Li et al., 2018b)	94.40	62.60	67.70	64.40	72.28
D-SAM (D'Innocente and Caputo, 2018)	91.75	56.95	58.95	60.84	67.03
MLDG (Li et al., 2018a)	94.4	61.3	67.7	65.9	73.30
JiGen (Carlucci et al., 2019)	96.93	60.90	70.62	64.30	73.19
MASF (Dou et al., 2019)	94.78	64.90	69.14	67.64	74.11
MMLD (Matsuura and Harada, 2020)	96.66	58.77	71.96	**68.13**	73.88
Ours	**97.85**	**65.26**	**71.47**	66.62	**75.31**

have good performance. Based on those findings we report the empirical results with $\alpha = 2.0$ and $\beta = 40.0$.

Follow the setting of (Carlucci et al., 2019), we then examined our algorithm on office-home dataset. For this Office-home dataset, we also use reverse validation to set the learning rate as $2e-4$ for the whole model together with mini-batch size 128. For the rest hyper-parameters we keep the same with PACS and VLCS experiments. To avoid over-training, we also adopt the early stopping technique. All the experiments are implemented by *Pytorch* (Paszke et al., 2019).

5.3. Experiments Results

We firstly report the empirical results on PACS and VLCS dataset using AlexNet as feature extractor. For each generalization task, we train the model on all the source domains and test on the target domain and report the average of top 5 accuracy. The results on PACS and VLCS dataset using AlexNet are reported in Table 3 and Table 4, respectively. For each table, the empirical results refers to the average accuracy about training on source domains while testing on the target domain. From the empirical results, we could see our method could outperform the baselines both on the PACS and VLCS dataset, indicating an improvement on benchmark performances. This showed the effectiveness of our proposed method. Except to these two common evaluation benchmark, to show the effectiveness of our method on more large-scaled dataset, we

Table 5: Empirical Results on Office-home dataset

	Art	Clipart	Product	Real-World	Avg.
Deep All	55.59	42.42	70.34	70.86	59.81
D-SAM(D'Innocente and Caputo, 2018)	**58.03**	44.37	69.22	71.45	60.77
JiGen(Carlucci et al., 2019)	53.04	**47.51**	71.47	72.79	61.20
Ours	55.34	44.82	**72.03**	**73.55**	**61.44**

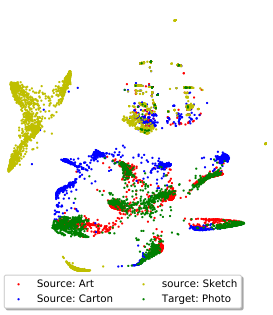
then report the empirical results on Office-home dataset in Table 5. As stated before, Office-home is a more larger and challenging dataset contains more diverse features from 65 different classes. To evaluate the performance on this dataset requires large amount of computational resources. Due to the limits, we follow the evaluation protocol of (Carlucci et al., 2019) to report the empirical results. From those results, we could observe that our algorithm could outperform the previous Domain Generalization method, this also confirm the effectiveness of our proposed method.

5.4. Further Analysis

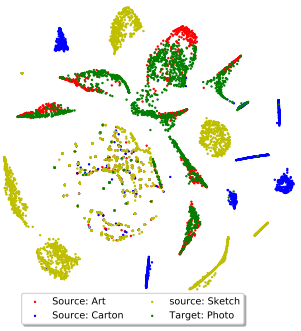
To further show the effectiveness of our algorithm especially on more deep models, follow Dou et al. (2019), we also report the results of our algorithm by using ResNet-18 backbone on PACS dataset in Table 6. The ResNet-18 backbone, the output feature dim will be 512. From the results, we could observe that our method could outperform the baselines on most generalization tasks and on average $+1.6\%$ accuracy improvement. Then, we implement ablation studies on each component of our algorithm. We report the empirical results of ablation studies in Table 7, where we test the ablation studies on both the AlexNet backbone and ResNet-18 backbone. We compare the ablations by, (1) *Deep All*: Train the model using feature extractor on source domain datasets with classification loss only, that is, neither optimal transport nor metric learning techniques is adopted. (2) *No* $\mathcal{L}_D$: Train the model with classification loss and metric learning loss but without adversarial training component; (3) $\mathcal{L}_{MS}$ *w.o.* w^+: omit the positive weighting scheme in $\mathcal{L}_{MS}$ (4) $\mathcal{L}_{MS}$ *w.o.* w^- : omit the positive weighting scheme in $\mathcal{L}_{MS}$. (5) *No* $\mathcal{L}_{MS}$: Train the model with classification loss and adversarial loss but without metric learning component; (6) *WADG-All*: Train the model with full objective Eq. 14.

Table 6: Empirical Results (accuracy %) on PACS dataset with pre-trained ResNet-18 as feature extractor .

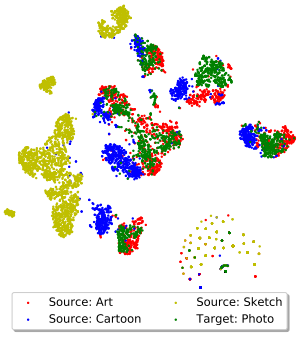
Method	Art	Cartoon	Sketch	Photo	Avg.
Deep All	77.87	75.89	69.27	95.19	79.55
D-SAM(D'Innocente and Caputo, 2018)	77.33	72.43	77.83	95.30	80.72
JiGen(Carlucci et al., 2019)	79.42	75.25	71.35	96.03	80.51
MASF(Dou et al., 2019)	80.29	77.17	71.69	94.99	81.04
MMLD(Matsuura and Harada, 2020)	81.28	77.16	72.29	**96.09**	81.83
Ours	**81.56**	**78.02**	**78.43**	95.82	**83.45**



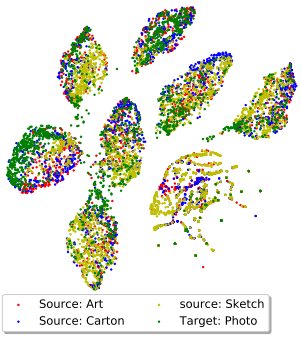
(a) Deep All



(b) No $\mathcal{L}_D$

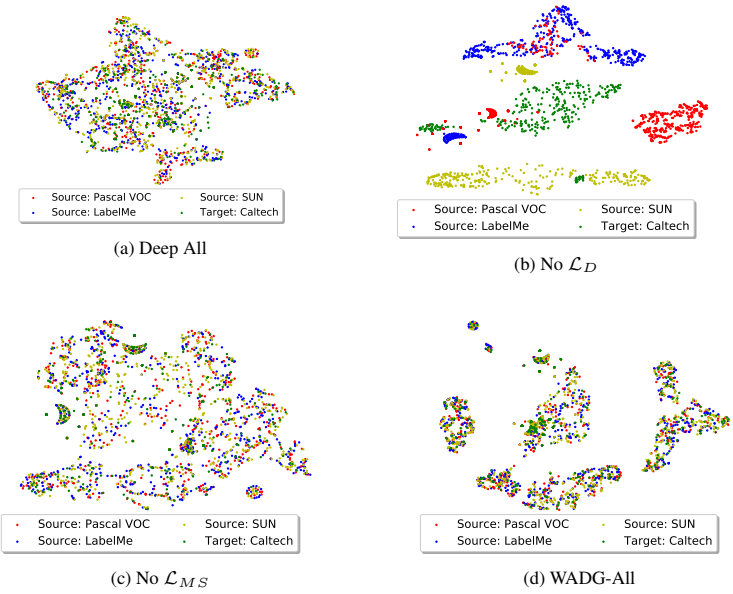


(c) No $\mathcal{L}_{MS}$



(d) WADG-All

Figure 4: T-SNE visualization of ablation studies on PACS dataset for Target domain as *Photo*. Detailed analysis is presented in section 5.4.



(a) Deep All (b) No $\mathcal{L}_D$

(c) No $\mathcal{L}_{MS}$ (d) WADG-All

Figure 5: T-SNE visualization of ablation studies on VLCS dataset for Target domain as *Caltech*. Detailed analysis is presented in section 5.4.

From the results, we could observe that one we omit the adversarial training, the accuracy would drop off rapidly ($\sim$ 3.5% with AlexNet backbone and $\sim$ 5.8% with ResNet-18 backbone). The contribution of the metric learning loss is relatively small compared with adversarial loss. Comparing the ablations $\mathcal{L}_{MS}$ *w.o.* w^+ and $\mathcal{L}_{MS}$ *w.o.* w^-, we could observe almost similar accuracy. This indicates that the positive and negative weighting scheme of the metric learning objective may have equivalent contribution. . Once we omit the metric learning loss, the performance will drop $\sim$ 2.1% and $\sim$ 2.5% with AlexNet and ResNet-18 backbone, respectively.

Then, to better understand the contribution of each component of our algorithm, the T-SNE visualization of the ablation studies of each components on PACS and VLCS dataset are represented in Fig. 4 for the generalization task of target domain *Photo*. and Fig. 5, respectively. Since our goal is to not only align the feature distribution but also encourage a cohesion and separable boundary, in order to show the alignment and clustering performance, we report the T-SNE features of all the source domains and target domain to show the feature alignment and clustering across domains.

Table 7: Ablation Studies on PACS dataset on all components of our proposed method using AlexNet and ResNet-18 backbone

Ablation	AlexNet					ResNet-18				
	Art	Carton	Sketch	Photo	Avg.	Art	Carton	Sketch	Photo	Avg.
Deep All	63.30	63.13	54.07	87.70	67.05	77.87	75.89	69.27	95.19	79.55
No $\mathcal{L}_D$	65.80	69.64	63.91	89.53	72.22	74.62	73.02	68.67	94.86	77.79
No $\mathcal{L}_{MS}$	66.78	71.47	68.12	88.87	73.65	78.25	76.27	73.42	95.68	80.91
$\mathcal{L}_{MS}$ w.o. w^+	66.31	70.86	67.11	88.97	73.31	80.58	77.95	75.13	95.63	82.32
$\mathcal{L}_{MS}$ w.o. w^-	66.41	70.95	68.73	87.38	73.37	79.98	77.65	77.89	95.21	82.68
WADG-All	**70.21**	**72.51**	**70.32**	**89.81**	**75.71**	**81.56**	**78.02**	**78.43**	**95.82**	**83.45**

For PASC dataset, as we can see, the T-SNE features by *Deep All* could neither project the instances from different domains to align with each other nor cluster the features into groups. The T-SNE features by *No* $\mathcal{L}_D$ showed the metric learning loss could to some extent to cluster the features, but without the adversarial training, the features could not be aligned well. The T-SNE features by *No* $\mathcal{L}_{MS}$ showed that the adversarial training could help to align the features from different domains but could not have a good clustering performance. The T-SNE features by *WADG-All* showed that the full objective could help to not only align the features from different domains but also could cluster the features from different domains into several cluster groups, which confirms the effective of our algorithm.

As for the VLCS dataset, we could observe similar performance on the T-SNE on the VLCS dataset while the features are somehow overlap with each other. This is due to the features in Caltech domain is somehow easy to learn and predict. As also analyzed in Li et al. (2017a), a supervised model on Caltech domain could achieved $\sim 100\%$ accuracy, which also confirms that the features in Caltech domain is easy to learn indicating the features might be more likely overlapping with each other. As we can see from Fig.5d, the WADG method could help to separate the features with each other, which again confirms the effectiveness of our proposed method.

6. Conclusion

In this paper, we proposed the Wasserstein Adversarial Domain Generalization algorithm for not only aligning the source domain features and transferring to an unseen target domain but also leveraging the label information across domains. We first adopt

optimal transport with Wasserstein distance for aligning the marginal distribution and then adopt the metric learning method to encourage a domain-independent distinguishable feature space for a clear classification boundary. The experiments results showed our proposed algorithm could outperform most of the baseline methods on two standard benchmark datasets. Furthermore, the ablation studies and visualization of the T-SNE features also confirmed the effectiveness of our algorithm.

Acknowledgement

This work has been partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC), The Fonds de recherche du Québec - Nature et technologies (FRQNT). Fan Zhou is supported by China Scholarship Council.

References

Arjovsky M, Chintala S, Bottou L. Wasserstein gan. arXiv preprint arXiv:170107875 2017;.

Balaji Y, Sankaranarayanan S, Chellappa R. Metareg: Towards domain generalization using meta-regularization. In: Advances in Neural Information Processing Systems. 2018. p. 998–1008.

Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan J. A theory of learning from different domains. Machine Learning 2010;79:151–75. URL: http://www.springerlink.com/content/q6qk230685577n52/.

Carlucci FM, D'Innocente A, Bucci S, Caputo B, Tommasi T. Domain generalization by solving jigsaw puzzles. In: CVPR. 2019. .

Courty N, Flamary R, Tuia D, Rakotomamonjy A. Optimal transport for domain adaptation. IEEE transactions on pattern analysis and machine intelligence 2016;39(9):1853–65.

Deng W, Zheng L, Sun Y, Jiao J. Rethinking triplet loss for domain adaptation. IEEE Transactions on Circuits and Systems for Video Technology 2020;:1–.

Ding Z, Fu Y. Deep domain generalization with structured low-rank constraint. IEEE Transactions on Image Processing 2017;27(1):304–13.

Dou Q, de Castro DC, Kamnitsas K, Glocker B. Domain generalization via model-agnostic learning of semantic features. In: Advances in Neural Information Processing Systems. 2019. p. 6447–58.

D'Innocente A, Caputo B. Domain generalization with domain-specific aggregation modules. In: German Conference on Pattern Recognition. Springer; 2018. p. 187–98.

Finn C, Abbeel P, Levine S. 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. p. 1126–35.

Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V. Domain-adversarial training of neural networks. The Journal of Machine Learning Research 2016;17(1):2096–30.

Ghifary M, Bastiaan Kleijn W, Zhang M, Balduzzi D. Domain generalization for object recognition with multi-task autoencoders. In: Proceedings of the IEEE international conference on computer vision. 2015. p. 2551–9.

Goldberg A, Zhu X, Singh A, Xu Z, Nowak R. Multi-manifold semi-supervised learning. In: Artificial Intelligence and Statistics. 2009. p. 169–76.

Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC. Improved training of wasserstein gans. In: Advances in neural information processing systems. 2017. p. 5767–77.

Hadsell R, Chopra S, LeCun Y. Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). IEEE; volume 2; 2006. p. 1735–42.

Ilse M, Tomczak JM, Louizos C, Welling M. Diva: Domain invariant variational autoencoders. arXiv preprint arXiv:190510427 2019;.

Kamnitsas K, Castro DC, Folgoc LL, Walker I, Tanno R, Rueckert D, Glocker B, Criminisi A, Nori A. Semi-supervised learning via compact latent space clustering. arXiv preprint arXiv:180602679 2018;.

Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980 2014;.

Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. 2012. p. 1097–105.

Li D, Yang Y, Song YZ, Hospedales T. Deeper, broader and artier domain generalization. In: International Conference on Computer Vision. 2017a. .

Li D, Yang Y, Song YZ, Hospedales TM. Deeper, broader and artier domain generalization. In: Proceedings of the IEEE international conference on computer vision. 2017b. p. 5542–50.

Li D, Yang Y, Song YZ, Hospedales TM. Learning to generalize: Meta-learning for domain generalization. In: Thirty-Second AAAI Conference on Artificial Intelligence. 2018a. .

Li H, Jialin Pan S, Wang S, Kot AC. Domain generalization with adversarial feature learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018b. p. 5400–9.

Li Y, Tian X, Gong M, Liu Y, Liu T, Zhang K, Tao D. Deep domain generalization via conditional invariant adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV). 2018c. p. 624–39.

Li Y, Yang Y, Zhou W, Hospedales TM. Feature-critic networks for heterogeneous domain generalization. arXiv preprint arXiv:190111448 2019;.

Long M, Cao Z, Wang J, Jordan MI. Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems. 2018. p. 1640–50.

Long M, Cao Z, Wang J, Philip SY. Learning multiple tasks with multilinear relationship networks. In: Advances in neural information processing systems. 2017. p. 1594–603.

Ma Z, Chang D, Xie J, Ding Y, Wen S, Li X, Si Z, Guo J. Fine-grained vehicle classification with channel max pooling modified cnns. IEEE Transactions on Vehicular Technology 2019a;68(4):3224–33.

Ma Z, Ding Y, Wen S, Xie J, Jin Y, Si Z, Wang H. Shoe-print image retrieval with multi-part weighted cnn. IEEE Access 2019b;7:59728–36.

Ma Z, Lai Y, Kleijn WB, Song YZ, Wang L, Guo J. Variational bayesian learning for dirichlet process mixture of inverted dirichlet distributions in non-gaussian image feature modeling. IEEE transactions on neural networks and learning systems 2018a;30(2):449–63.

Ma Z, Leijon A, Kleijn WB. Vector quantization of lsf parameters with a mixture of dirichlet distributions. IEEE Transactions on Audio, Speech, and Language Processing 2013;21(9):1777–90.

Ma Z, Yu H, Chen W, Guo J. Short utterance based speech language identification in intelligent vehicles with time-scale modifications and deep bottleneck features. IEEE transactions on vehicular technology 2018b;68(1):121–8.

Matsuura T, Harada T. Domain generalization using a mixture of multiple latent domains. In: AAAI. 2020. .

Motiian S, Piccirilli M, Adjeroh DA, Doretto G. Unified deep supervised domain adaptation and generalization. In: The IEEE International Conference on Computer Vision (ICCV). 2017. .

Muandet K, Balduzzi D, Schölkopf B. Domain generalization via invariant feature representation. In: International Conference on Machine Learning. 2013. p. 10–8.

Oh Song H, Xiang Y, Jegelka S, Savarese S. Deep metric learning via lifted structured feature embedding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 4004–12.

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al. Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. 2019. p. 8024–35.

Redko I, Habrard A, Sebban M. Theoretical analysis of domain adaptation with optimal transport. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer; 2017. p. 737–53.

Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. p. 815–23.

Shen J, Qu Y, Zhang W, Yu Y. Wasserstein distance guided representation learning for domain adaptation. In: AAAI Conference on Artificial Intelligence. 2018. .

Torralba A, Efros AA. Unbiased look at dataset bias. In: CVPR 2011. IEEE; 2011. p. 1521–8.

Ustinova E, Lempitsky V. Learning deep embeddings with histogram loss. In: Advances in Neural Information Processing Systems. 2016. p. 4170–8.

Venkateswara H, Eusebio J, Chakraborty S, Panchanathan S. Deep hashing network for unsupervised domain adaptation. In: (IEEE) Conference on Computer Vision and Pattern Recognition (CVPR). 2017. .

Volpi R, Namkoong H, Sener O, Duchi JC, Murino V, Savarese S. Generalizing to unseen domains via adversarial data augmentation. In: Advances in Neural Information Processing Systems. 2018. p. 5334–44.

Wainwright MJ. High-dimensional statistics: A non-asymptotic viewpoint. volume 48. Cambridge University Press, 2019.

Wang X, Han X, Huang W, Dong D, Scott MR. Multi-similarity loss with general pair weighting for deep metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019. p. 5022–30.

Weinberger KQ, Saul LK. Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research 2009;10(2).

Wen J, Zheng N, Yuan J, Gong Z, Chen C. Bayesian uncertainty matching for unsupervised domain adaptation. arXiv preprint arXiv:190609693 2019;.

Wu CY, Manmatha R, Smola AJ, Krahenbuhl P. Sampling matters in deep embedding learning. In: Proceedings of the IEEE International Conference on Computer Vision. 2017. p. 2840–8.

Xie J, Song Z, Li Y, Ma Z. Mobile big data analysis with machine learning. arXiv preprint arXiv:180800803 2018;.

Xu P, Yin Q, Huang Y, Song YZ, Ma Z, Wang L, Xiang T, Kleijn WB, Guo J. Cross-modal subspace learning for fine-grained sketch-based image retrieval. Neurocomputing 2018;278:75–86.

Yi D, Lei Z, Liao S, Li SZ. Deep metric learning for person re-identification. In: 2014 22nd International Conference on Pattern Recognition. IEEE; 2014. p. 34–9.

Zhao H, Combes RTd, Zhang K, Gordon GJ. On learning invariant representation for domain adaptation. arXiv preprint arXiv:190109453 2019;.

Zhou F, Shui C, Huang B, Wang B, Chaib-draa B. Discriminative active learning for domain adaptation. arXiv preprint arXiv:200511653 2020;.

Zhu F, Ma Z, Li X, Chen G, Chien JT, Xue JH, Guo J. Image-text dual neural network with decision strategy for small-sample image classification. Neurocomputing 2019;328:182–8.

Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, Xiong H, He Q. A comprehensive survey on transfer learning. arXiv preprint arXiv:191102685 2019;.