Generalizing to Unseen Domains with Wasserstein Distributional Robustness under Limited Source Knowledge

Jingge Wang, Liyan Xie, Yao Xie, Shao-Lun Huang and Yang Li

Abstract—**Domain generalization aims at learning a universal model that performs well on unseen target domains, incorporating knowledge from multiple source domains. In this research, we consider the scenario where different domain shifts occur among conditional distributions of different classes across domains. When labeled samples in the source domains are limited, existing approaches are not sufficiently robust. To address this problem, we propose a novel domain generalization framework called Wasserstein Distributionally Robust Domain Generalization (WDRDG), inspired by the concept of distributionally robust optimization. We encourage robustness over conditional distributions within class-specific Wasserstein uncertainty sets and optimize the worst-case performance of a classifier over these uncertainty sets. We further develop a test-time adaptation module leveraging optimal transport to quantify the relationship between the unseen target domain and source domains to make adaptive inference for target data. Experiments on the Rotated MNIST, PACS and the VLCS datasets demonstrate that our method could effectively balance the robustness and discriminability in challenging generalization scenarios.**

Index Terms—**Domain generalization, distributionally robust optimization, optimal transport, Wasserstein uncertainty set.**

I. Introduction

In many practical learning applications, labeled training data are only available from fragmented source domains. It is thus a challenge to learn a robust model for future data that could come from a new domain, with unknown domain shift. One commonly acknowledged solution to this challenge is domain generalization [1], which aims at learning a model that generalizes well to target domains based on available training data from multiple source domains and in a total absence of prior knowledge about the target domain. A surge of popularity has been seen recently in the application of domain generalization in various fields, such as computer vision [2]–[8], natural processing [9]–[12], and reinforcement learning [13], etc.

Numerous methods have been developed for learning a generalizable model by exploiting the available data from the source domains, where the shifts across these source domains are implicitly assumed to be representative of the target shift that we will meet at test time. The well-known approaches include learning domain-invariant feature representations through kernel functions [1], [14]–[19], or by distribution alignment [20]–[22], or in an adversarial manner [8], [23]–[26]. The learned invariance across source domains, however, may not be typical if the unseen target shift is of extreme magnitude. In this case, forcing distributions to align in a common representation space may result in a biased model that overfits the source domains, and only performs well for target domains that are similar to certain source domains.

Instead, to explicitly model unseen target domain shifts, meta-learning-based domain generalization methods like MLDG [13] divides the source domains into non-overlapping meta-train and meta-test domains, which fails to hedge against the possible target shift beyond the distribution shifts observed in source domains. Also, these approaches require sufficient source training data to make good meta-optimization within each mini-batch. Possible domain shift could also been modeled by enhancing the diversity of data based on some data augmentations [27], generating data in an adversarial manner [7], [28], [29] or constructing sample interpolation [30], [31]. Learning with limited labeled original samples in this way will weaken their performance, since the new generated data will dominate and the domain shift caused by the artificial data manipulations will largely determine the generalization performance.

In this work, we propose a domain generalization framework to explicitly model the unknown target domain shift under limited source knowledge, by extrapolating beyond the domain shifts among multiple source domains in a probabilistic setting via distributionally robust optimization (DRO) [32]. To model the shifts between training and test distributions, DRO usually assumes the testing data is generated by a perturbed distribution of the underlying data distribution, and the perturbation is bounded explicitly by an uncertainty set. It then optimizes the worst-case performance of a model over the uncertainty set to hedge against the perturbations [33]–[36]. The uncertainty set contains distributions that belong to a non-parametric distribution family, which is typically distributions centered around the empirical training distributions defined via some divergence metrics, e.g., Kullback–Leibler divergence [32], or other f-divergences [37]–[40], or Wasserstein distance [33], [41]–[44], etc. These pre-defined distance constraints

Jingge Wang, Shao-Lun Huang and Yang Li are with Tsinghua Shenzhen International Graduate School, Tsinghua University, China.

Liyan Xie is with the School of Data Science, Chinese University of Hong Kong, Shenzhen, China.

Yao Xie is with the School of Industrial and Systems and Engineering, Georgia Institute of Technology, USA.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the authors.

of uncertainty sets will confer robustness against a set of perturbations of distributions.

As a promising tool that connects distribution uncertainty and model robustness, DRO has been incorporated into domain generalization in some works. Volpi et al. [7] augmented the data distribution in an adversarial manner, which appends some new perturbed samples from the fictitious worst-case target distributions at each iteration, and the model is updated on these samples. Duchi et al. [40] solves the DRO to learn a model within a f-divergence uncertainty set and learns the best radius of the set in a heuristic way by validating on part of the training data. Let X denote the input feature and Y denote the label. While the studies by [7] and [40] discuss the distributional shifts directly in the joint distribution $P(X, Y)$, our work takes a distinct approach by decomposing the joint distribution and establishing class-specific distributional uncertainty sets, which enables us to manage possible varying degrees of distributional perturbations for each class in a more explicit manner.

When labeled training source samples are limited in source domains, the distributional perturbations for each class could vary widely. In such a scenario, unifying these varying degrees of domain perturbations within a single shared uncertainty set as have been done for the joint distribution is potentially overlooking the inherent differences among classes. As such, to explicitly examine the distributional shift among classes, we decompose the joint distribution $P(X, Y) = P(X|Y)P(Y)$ and address each part independently. Our primary focus lies in managing the class-conditional shift [45], under the assumption that there is no shift in the class prior distribution, i.e., the distribution $P(Y)$ stays consistent across all source domains. Furthermore, we also illustrate how our research can be readily expanded to situations that involve a shift in the class prior distribution. To be more specific, we encode the domain perturbations of each class within a class-specific Wasserstein uncertainty set. Compared with Kullback–Leibler divergence, Wasserstein distance is well-known for its ability to measure divergence between distributions defined on different probability space, which may happen when the limited samples have no overlap. While the classic DRO with one Wasserstein uncertainty set can be formulated into a tractable convex problem [46], tractability results for DRO with multiple Wasserstein uncertainty sets for each class are also available [34].

It is crucial to set appropriate uncertainty sets based on training data from multiple source domains for the success of DRO, since they control the conservatism of the optimization problem [43]. A richer uncertainty set may contain more true target distributions with higher confidence, but comes with more conservative and less practical solution. More precise uncertainty set incentivizes higher complexity and more difficult solution. Therefore, uncertainty sets should be large enough to guarantee robustness, but not so large as to overlap with each other. We manage to control the discriminability among class-specific uncertainty sets with additional constraints while ensuring the largest possible uncertainty.

When performing classification on data from target domains, we conduct a test-time adaptation strategy to further reduce the domain shift and make inference for testing data adaptively. We employ optimal transport weights to apply the optimal classifier learned from the source distributions on the test sample, which we prove to be equivalent to transporting the target samples to source domains before making the prediction.

In summary, our main contributions include:

- We propose a domain generalization framework that solves the Wasserstein distributionally robust optimization problem to learn a robust model over multiple source domains, where class-conditional domain shifts are formulated in a probabilistic setting within class-specific Wasserstein uncertainty sets.
- To improve upon the original Wasserstein distributionally robust optimization method with heuristic magnitude of uncertainty, we design a constraint that balances robustness and discriminability of uncertainty sets.
- We develop a test-time optimal transport-based adaptation module to make adaptive and robust inferences for samples in the target domain. A generalization bound on the target classifier is presented. Experiments on several multi-domain vision datasets show the effectiveness of our proposed framework comparing with the state-of-the-arts.

II. Preliminaries and Problem Setup

For the common K-class classification problem, denote the feature space as $\mathcal{X} \subset \mathbb{R}^d$ and the label space as $\mathcal{Y} = \{1, \ldots, K\}$. Let $\phi : \mathcal{X} \to \Delta_K$ be the prediction function which assigns each feature vector $\boldsymbol{x}$ as class k with likelihood $\phi_k(\boldsymbol{x})$. Here $\Delta_K := \{\boldsymbol{\xi} \in \mathbb{R}^K : \boldsymbol{\xi}_i \geq 0, \sum_{i=1}^K \boldsymbol{\xi}_i = 1\}$ denotes the probability simplex. Based on the prediction function ϕ, the corresponding classifier Φ maps each feature vector $\boldsymbol{x}$ to the class $\Phi(\boldsymbol{x}) = \arg\max_k \{\phi_k(\boldsymbol{x})\}$ (ties are broken arbitrarily). In the following, we will also use ϕ to represent the classifier.

Given training samples $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ drawn i.i.d from the true data-generating distribution over $\mathcal{X} \times \mathcal{Y}$, we denote the empirical class-conditional distributions for each class as

$$\widehat{Q}_k := \frac{1}{|i : y_i = k|} \sum_{i=1}^{n} \delta_{\boldsymbol{x}_i} 1\{y_i = k\}, \ k = 1, \ldots, K.$$

Here, $\delta_{\boldsymbol{x}}$ indicates a Dirac measure centered at $\boldsymbol{x}$ and $1\{\cdot\}$ is the indicator function. Therefore, $\widehat{Q}_k$ can be viewed as the empirical distribution for training samples within the class k. In light of [34], [35], the test distribution of each class is likely to be distributions centered around the empirical class-conditional distribution $\widehat{Q}_k$ within the uncertainty set defined using, for example, the Wasserstein distance.

The Wasserstein distance [47], [48] of order p between any two distributions P and Q, is defined as:

$$\mathcal{W}_p(P, Q) = \left(\min_{\gamma \in \Gamma(P,Q)} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{x}') \sim \gamma} \left[\|\boldsymbol{x} - \boldsymbol{x}'\|^p \right] \right)^{1/p}, \quad (1)$$

where $\Gamma(P, Q)$ is the collection of all joint distributions with the first and second marginals being the distribution P and Q, respectively. We consider the Wasserstein distance of order $p = 2$, and the corresponding norm $\|\cdot\|$ is set as Euclidean distance. Thus, we have the test distribution of each class k belongs to the following set:

$$\mathcal{P}_k = \left\{ P_k \in \mathscr{P}(\mathcal{X}) : \mathcal{W}_2 \left(P_k, \widehat{Q}_k \right) \leq \theta_k \right\}, \tag{2}$$

where $\theta_k \geq 0$ denotes the radius of the uncertainty set and $\mathscr{P}(\mathcal{X})$ denotes the set of all probability distributions over $\mathcal{X}$. A robust classifier Φ (or equivalently the prediction function ϕ) can be obtained by solving the following minimax optimization problem:

$$\min_{\phi:\mathcal{X}\to\Delta_K} \max_{P_k\in\mathcal{P}_k, 1\leq k\leq K} \Psi\left(\phi; P_1, \ldots, P_K\right), \tag{3}$$

where $\Psi\left(\phi; P_1, \ldots, P_K\right)$ is the total risk of the classifier ϕ on certain distributions $P_1, \ldots, P_K$. The inner maximum problem refers to the worst-case risk over uncertainty sets $\mathcal{P}_1, \ldots, \mathcal{P}_K$. Suppose $(\phi^*; P_1^*, \ldots, P_K^*)$ is an optimal solution pair to the saddle-point problem (3), then $P_1^*, \ldots, P_K^*$ are called the least favorable distributions (LFDs) [49], and ϕ^* induces the optimal classifier that minimizes the worst-case risk.

The likelihood that a sample is misclassified is usually taken as the risk, i.e., $1 - \phi_k(\boldsymbol{x})$ for any sample $\boldsymbol{x}$ with real label k. Specially, when assuming the simple case with equal class prior distributions $\mathbb{P}(y = k) = 1/K, k = 1, \ldots, K$ for all classes, the total risk of misclassifying data from all K classes is

$$\Psi\left(\phi; P_1, \ldots P_K\right) = \sum_{k=1}^{K} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim P_k} \left[1 - \phi_k(\boldsymbol{x})\right]. \tag{4}$$

However, in a more general classification problem, to compensate for the possible class imbalance scenario, a series of class-weighting methods assign different weights to misclassifying samples from different classes [50], [51]. One of the most natural approaches is to incorporate the class prior distributions $\mathbb{P}(y = k)$ of each class into the risk function [52], [53] as

$$\Psi\left(\phi; P_1, \ldots P_K\right) = \sum_{k=1}^{K} \mathbb{P}(y = k) \mathop{\mathbb{E}}_{\boldsymbol{x}\sim P_k} \left[1 - \phi_k(\boldsymbol{x})\right], \tag{5}$$

which is a general form of (4).

In domain generalization problems, we have access to R source domains $\{\mathcal{D}^{s_r}\}_{r=1}^{R}$, with training samples $\left\{\left(\boldsymbol{x}_1, \boldsymbol{y}_1\right), \ldots, \left(\boldsymbol{x}_{n_r}, \boldsymbol{y}_{n_r}\right)\right\}$ from the r-th source domain drawn i.i.d from the joint distribution P^{s_r} on $\mathcal{X} \times \mathcal{Y}$. The goal is to learn a robust classifier that performs well on the unseen target domain $\mathcal{D}^t$, which contains instances from the joint distribution P^t. For each class k, denote the empirical class-conditional distributions in source domain $\mathcal{D}^{s_r}$ and target domain $\mathcal{D}^t$ as $\widehat{Q}_k^{s_r}$ and $\widehat{Q}_k^t$, respectively. Instead of constructing uncertainty sets relative to the empirical (training) distributions of a single domain as in the classic DRO formulation, we need to set the uncertainty sets using distributions $\widehat{Q}_k^{s_r}$ from multiple source domains, which is detailed in the next section.

III. Wasserstein Distributionally Robust Domain Generalization

In this section, we present our proposed framework for domain generalization that leverages the empirical distributions from multiple source domains as shown in Figure 1a, and the process of distributionally robust optimization is shown in Figure 1b. The adaptive inference for the target domain is shown in Figure 1c. Here we show binary classification for simplicity.

More specifically, we first extrapolate the class-conditional source distributions to a Wasserstein uncertainty set for each class. Figure 1a illustrates the construction of uncertainty sets of two classes. Their closeness is further controlled by the parameter δ to ensure discriminability. A convex solver then solves the distributionally robust optimization over these uncertainty sets, obtaining the least favorable distributions (LFDs), which are represented as probability mass vectors depicted in Figure 1b. Figure 1c shows the inference process for target samples, where optimal transport [54] is used to re-weight LFDs adaptively.

Details of the construction of uncertainty sets and the additional Wasserstein constraints could be found in Sections III-A and III-B. Section III-C discusses the re-formulation of the Wasserstein robust optimization. Adaptive inference for samples in the target domain is presented in section III-D. In III-E, we further analyze the generalization bound of the proposed framework.

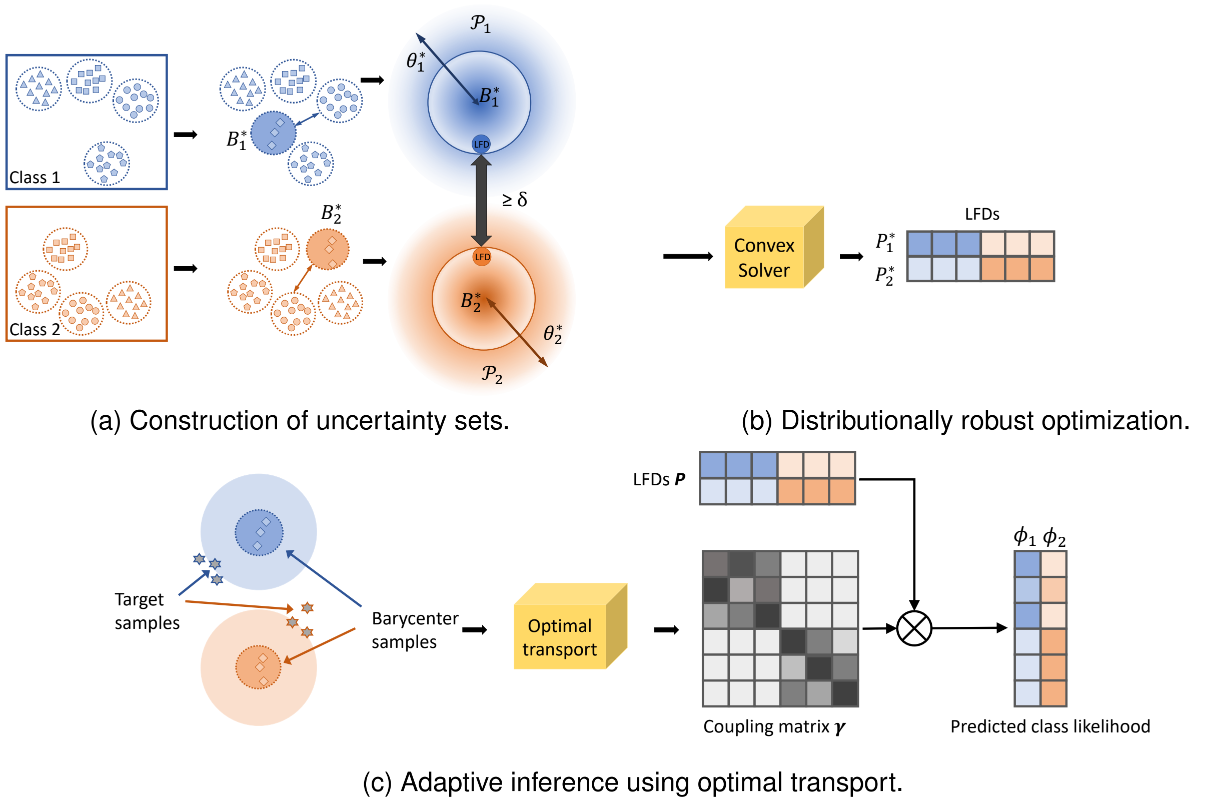
A. Construction of Uncertainty Sets

To measure distributionally divergence, we chose Wasserstein distance since it can handle divergences between discrete and continuous distributions, which is essential for our use of empirical (discrete) distributions as the center of the uncertainty sets. We construct the uncertainty sets controlled mainly by two terms: the reference distribution that represents the center of the uncertainty set, and the radius parameter that controls the size of the set, i.e., an upper bound of the divergence between the reference distribution and other distributions in the set. We use *Wasserstein barycenter* [55] as the reference distribution, which is the average of multiple given distributions and is capable of leveraging the inherent geometric relations among them [20]. Given empirical class-conditional distributions $\widehat{Q}_k^{s_1}, \ldots, \widehat{Q}_k^{s_R}$ for each class k from R different source domains, the Wasserstein barycenter for class k is defined as

$$B_k^* = \arg\min_{B_k} \sum_{r=1}^{R} \frac{1}{R} \mathcal{W}_2(B_k, \widehat{Q}_k^{s_r}), k = 1, \ldots, K, \tag{6}$$

which could be a proxy of the reference distribution for each uncertainty set. Suppose each barycenter supports on b samples uniformly, i.e., $B_k = \sum_{i=1}^{b} \frac{1}{b} \delta_{\boldsymbol{x}_i^{(k)}}$, where $\{\boldsymbol{x}_i^{(k)}\}_{i=1}^{b}$ are b barycenter samples for class k, then (6) only optimizes over the locations $\boldsymbol{x}_i^{(k)}$ of the uniform distribution on the feature space, which could be efficiently computed using POT package [56].

To ensure that the uncertainty sets are large enough to avoid misclassification for unseen target samples, the maximum of



(a) Construction of uncertainty sets.

(b) Distributionally robust optimization.

(c) Adaptive inference using optimal transport.

Fig. 1. An overview of our WDRDG framework, consisting of three components: (a) Wasserstein uncertainty set construction for each class based on the empirical Wasserstein barycenters and radius obtained from given source domains. One constraint is added to control the discriminability of LFDs; (b) distributionally robust optimization to solve for the least favorable distributions; (c) adaptive inference for target testing samples based on probability mass on LFDs and coupling matrix from optimal transportation between barycenter samples and target samples.

all R Wasserstein distances between class-conditional distributions of each source domain $\widehat{Q}_k^{s_r}$ and the barycenter B_k^*, is used as the radius for each class k:

$$\theta_k^* = \max_{r=1,\dots,R} \mathcal{W}_2\left(B_k^*, \widehat{Q}_k^{s_r}\right). \tag{7}$$

In this way, we can construct the Wasserstein uncertainty set $\mathcal{P}_k$ of radius θ_k^* centered around B_k^* for each class k following (2):

$$\mathcal{P}_k = \left\{P_k \in \mathscr{P}(\widehat{\mathcal{X}}) : \mathcal{W}_2\left(P_k, B_k^*\right) \leq \theta_k^*\right\}. \tag{8}$$

Figure 1a shows the construction process of the uncertainty sets for two classes.

B. Balance Robustness and Discriminability

When the source training samples are limited, the class-conditional distributions may vary widely in practice. In this situation, the radius computed from (7) tends to be overly large, and the uncertainty sets of different classes may overlap with each other, leading to indistinguishable LFDs for optimization problem (3). As shown in Figure 2, overlap between each pair of class-specific uncertainty sets exist as the sum of their radius is larger than the Wasserstein distance between the corresponding barycenters.

Discriminability of LFDs is necessary since this leads to a well-defined problem of (3), which indirectly controls the discriminability of data from different classes. We add one more constraint to obtain significantly different LFDs that are discriminable, characterized by the Wasserstein distance between each pair of LFDs (P_u^*, P_v^*) within K classes:

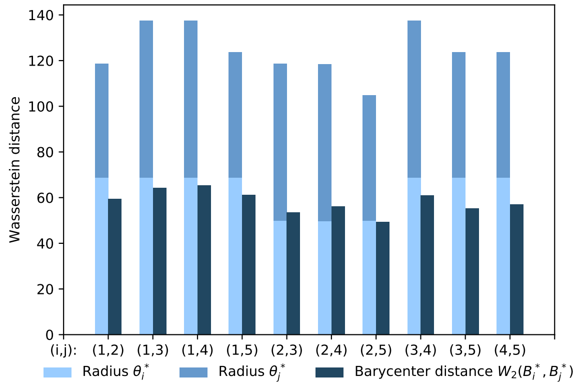


Fig. 2. Comparison between $\theta_i^* + \theta_j^*$ and the Wasserstein distance $W_2(B_i^*, B_j^*)$ for all 10 unique pairs (i, j) among all 5 classes of the VLCS dataset. The sum of uncertainty radius of any two classes is larger than the Wasserstein distance between the corresponding barycenters. The oversized radius will lead to overlapping class-specific uncertainty sets, and the distributions within them will be indistinguishable.

$$\mathcal{W}_2\left(P_u^*, P_v^*\right) \geq \delta,\ \ 1 \leq u < v \leq K, \tag{9}$$

where $\delta > 0$ is the threshold that indicates the discriminability,

Algorithm 1 Wasserstein distributionally robust domain generalization.

```
Input: $\{\widehat{Q}_k^{s_r}\}_{r=1}^R$ - empirical class-conditional distributions
    for each class $k$ in all $K$ classes from source domains
    $\{\mathcal{D}^{s_r}\}_{r=1}^R$;
    $b$ - number of barycenter samples for each class;
    $\delta$ - discriminability threshold parameter.
Output: $\phi(\boldsymbol{x}_j^t)$ - predictions for each of the unseen target
    samples $\{\boldsymbol{x}_j^t\}_{j=1}^{n_t}$.
1: for each class $k$ do
2:     Obtain barycenter $B_k^*$ by (6);
3:     Obtain radius $\theta_k^*$ using (7).
4:     Construct uncertainty sets $\mathcal{P}_k$ centered around $B_k^*$ with
       radius $\theta_k^*$ as formed in (2).
5: end for
6: Solve the optimization (11) for the optimal LFDs $P_k^*$.
7: The inference for each target sample is given by (14).
```

which is a hyperparameter that could be tuned on the validation set. In this way, robustness is ensured by large enough Wasserstein uncertainty sets, and the threshold δ guarantees discriminability among the uncertainty sets.

C. Distributionally Robust Optimization

Incorporating the constraints (9) into (3), we aim to solve the following minimax problem

$$\min_{\phi:\mathcal{X}\to\Delta_K} \max_{\substack{P_k\in\mathcal{P}_k,\ 1\le k\le K \\ \mathcal{W}(P_u,P_v)\ge\delta,\ 1\le u<v\le K}} \Psi\left(\phi; P_1,\ldots,P_K\right). \tag{10}$$

We establish the following theorem, stating a convex approximation of problem (10).

Theorem 1: Suppose the Wasserstein barycenter B_k^* for each class as defined in (6) is supported on b samples. Let S_b be the union of the support of $\{B_1^*,\ldots,B_K^*\}$ which contains $n_b = Kb$ samples $\{\boldsymbol{x}_i^b, i=1,\ldots,n_b\}$ in total. The class prior distributions of each class is denoted as $\mathbb{P}(y=k)$. Denote each distribution within the uncertainty set $\mathcal{P}_k$ as $P_k\in\mathbb{R}_+^{n_b}$. Let $\boldsymbol{C}\in\mathbb{R}_+^{n_b\times n_b}$ be the pairwise distance matrix of n_b samples, $\boldsymbol{C}_{i,j} = \|\boldsymbol{x}_i^b - \boldsymbol{x}_j^b\|^2$, $\gamma_k\in\mathbb{R}_+^{n_b\times n_b}$ be the coupling matrix between B_k^* and P_k, and $\beta_{u,v}\in\mathbb{R}_+^{n_b\times n_b}$ be the coupling matrix between any two distributions P_u, P_v in different classes. When using the Wasserstein metric of order 2, the least favorable distributions P_k^* of the problem (10) could be obtained by solving:

$$\begin{aligned}
\max_{\substack{P_1,\ldots,P_K\in\mathbb{R}_+^{n_b} \\ \gamma_1,\ldots,\gamma_K\in\mathbb{R}_+^{n_b\times n_b} \\ \beta_{u,v}\in\mathbb{R}_+^{n_b\times n_b}}} \quad & 1-\sum_{i=1}^{n_b}\max_{1\le k\le K}\mathbb{P}(y=k)P_k\left(\boldsymbol{x}_i^b\right) \\
\text{s.t.} \quad & \langle\gamma_k,\boldsymbol{C}\rangle_F \le (\theta_k^*)^2, \langle\beta_{u,v},\boldsymbol{C}\rangle_F \ge \delta^2, \\
& \gamma_k\mathbf{1}_{n_b} = B_k^*, \gamma_k^T\mathbf{1}_{n_b} = P_k, \\
& \beta_{u,v}\mathbf{1}_{n_b} = P_u, \beta_{u,v}^T\mathbf{1}_{n_b} = P_v, \\
& \forall 1\le k\le K, 1\le u<v\le K,
\end{aligned} \tag{11}$$

and the optimal prediction function of (10) satisfies $\phi_k^*(\boldsymbol{x}_i^b) = P_k^*\left(\boldsymbol{x}_i^b\right)/\sum_{k=1}^K P_k^*\left(\boldsymbol{x}_i^b\right)$ for any $\boldsymbol{x}_i^b\in S_b$.

The constraints on γ_k restrict each target class-conditional distribution to its respective uncertainty set of radius θ_k^*. The constraints on $\beta_{u,v}$ restrict the Wasserstein distance between each pair of class-conditional distributions in the target domain following (9). Based on the above theorem, the classification for any sample in the sample set S_b is given by $\Phi(\boldsymbol{x}_i^b) = \arg\max_{1\le k\le K} P_k^*(\boldsymbol{x}_i^b)$. The proof can be found in the supplementary material.

D. Adaptive Inference by Test-time Adaptation

Since the barycenters are the weighted average of distributions from multiple source domains, the barycenter samples in the support set S_b could be viewed as samples from a *generalized source domain* denoted as $\mathcal{D}^b$. For any sample in $\mathcal{D}^b$, the likelihood that it is assigned to each class could be decided based on ϕ^* by a non-parametric inference method such as KNN [35]. When making predictions for samples from an unseen target domain D^t, the domain shift between $\mathcal{D}^b$ and D^t needs to be considered. We adopt optimal transport to reduce the domain shift adaptively by the following test-time adaptation process.

Suppose $\widehat{\mu}_b = \sum_{i=1}^{n_b}\frac{1}{n_b}\delta_{\boldsymbol{x}_i^b}$ and $\widehat{\mu}_t = \sum_{j=1}^{n_t}\frac{1}{n_t}\delta_{\boldsymbol{x}_j^t}$ are the empirical marginal distributions of the feature vectors from the generalized source domain $\mathcal{D}^b$ and a target domain $\mathcal{D}^t$, respectively. Denote the coupling matrix of transporting from target to the generalized source distribution using optimal transport [54] as $\boldsymbol{\gamma} = [\gamma_1,\ldots,\gamma_{n_t}]^T\in\mathbb{R}^{n_t\times n_b}$, where each vector $\gamma_j\in\mathbb{R}^{n_b}$, $j=1,\ldots,n_t$, represents the transported mass from the j-th target sample to each of the n_b barycenter samples. In most optimal transport-based domain adaptation methods, each target sample $\boldsymbol{x}_j^t$, $j=1,\ldots,n_t$, is first transported to $\widehat{\boldsymbol{x}}_j^t$ in the generalized source domain $\mathcal{D}^b$ by the barycentric mapping:

$$\widehat{\boldsymbol{x}}_j^t = \sum_{i=1}^{n_b} n_t\boldsymbol{\gamma}_{j,i}\boldsymbol{x}_i^b,\ j=1,\ldots,n_t, \tag{12}$$

then having its label inferred based on the classifier learned on the labeled samples. Instead of such a two-step process, we propose an equivalent single-step inference process. The following proposition states the equivalence, and the proof can be found in the supplementary.

Proposition 1: Given the coupling matrix $\boldsymbol{\gamma}\in\mathbb{R}^{n_t\times n_b}$. Suppose we transport the target sample $\boldsymbol{x}_j^t$ from the empirical target distribution $\widehat{\mu}_t = \sum_{j=1}^{n_t}\frac{1}{n_t}\delta_{\boldsymbol{x}_j^t}$ to the generalized source domain empirical distribution $\widehat{\mu}_b = \sum_{i=1}^{n_b}\frac{1}{n_b}\delta_{\boldsymbol{x}_i^b}$ by the barycentric mapping as shown in (12), and obtain the class likelihood by re-weighting $\phi_k^*(\boldsymbol{x}_i^b)$ of all the samples $\boldsymbol{x}_i^b\in S_b$ using the weight function $w\left(\widehat{\boldsymbol{x}}_j^t,\boldsymbol{x}_i^b\right) = n_t\boldsymbol{\gamma}_{j,i}$. Then the resulting classifier is equivalent to directly re-weighting LFDs on the barycenter samples using the coupling matrix. The equivalent classification result is:

$$\Phi(\boldsymbol{x}_j^t) = \arg\max_{1\le k\le K}\sum_{i=1}^{n_b}\boldsymbol{\gamma}_{j,i}P_k^*(\boldsymbol{x}_i^b). \tag{13}$$

This proposition illustrates that domain difference between target domain and generalized source domain can be eliminated by adaptively applying the coupling matrix in the inference stage, without actually transporting the target samples to the generalized source domain. With LFDs P_k^* supported on barycenter sampless from solving (11) and Proposition 1, the classification of each target sample could be obtained by assigning each class with probability based on the re-weighted LFDs $\sum_{i=1}^{n_b} n_t \boldsymbol{\gamma}_{j,i} P_k^*(\boldsymbol{x}_i^b)$. The final decision $\Phi(\boldsymbol{x}_j^t)$ is made by choosing the class that maximizes the probability.

Denote the LFDs for all classes as $\boldsymbol{P} = [P_1^*, \dots, P_K^*]^T \in \mathbb{R}^{K \times n_b}$. Based on Proposition 1, the predicted class likelihood of each target sample $\boldsymbol{x}_j^t$ can be rewritten as

$$\phi(\boldsymbol{x}_j^t) = \frac{{\boldsymbol{\gamma}_j}^T \boldsymbol{P}^T}{{\boldsymbol{\gamma}_j}^T \boldsymbol{P}^T \mathbf{1}_K} = \left[\phi_1(\boldsymbol{x}_j^t), \dots, \phi_K\left(\boldsymbol{x}_j^t\right)\right], \tag{14}$$

where $0 \leq \phi_k\left(\boldsymbol{x}_j^t\right) \leq 1, \sum_{k=1}^K \phi_k\left(\boldsymbol{x}_j^t\right) = 1$. The algorithm is summarized in Algorithm 1. Further adding the optimal-transport based adaptive inference leads to our complete framework Wasserstein **D**istributionally **R**obust **D**omain **G**eneralization (**WDRDG**).

E. Generalization Analysis

We further analyze the generalization risk of our proposed method. Our analysis considers the domain shift between the target domain and the generalized source domain.

Based on (14), the classification decision for the test sample $\widehat{\boldsymbol{x}}_j^t$ in the target domain is based on the weighted average

$$\underset{1 \leq k \leq K}{\arg\max} \sum_{i=1}^{n_b} w\left(\widehat{\boldsymbol{x}}_j^t, \boldsymbol{x}_i^b\right) P_k^*(\boldsymbol{x}_i^b). \tag{15}$$

Consider a binary classification problem with label set $\{0, 1\}$. Let $\phi(\boldsymbol{x}) = [\phi_0(\boldsymbol{x}), \phi_1(\boldsymbol{x})]$ represents the prediction vector of $\boldsymbol{x}$ belonging to either classes. The true labeling function is denoted as $f : \mathcal{X} \to \{0, 1\}$. Considering the simple case that all classes are balanced, the expected risk that the correct label is not accepted for samples in any distribution μ is denoted as $\epsilon_\mu(\phi) = \mathbb{E}_{\boldsymbol{x}\sim\mu}[1 - \phi_{f(\boldsymbol{x})}(\boldsymbol{x})]$. We now present the following theorem stating the generalization bound.

Theorem 2: Suppose the distributionally robust prediction function ϕ^{S_b} learned from the sample set S_b is M-Lipschitz continuous for some $M \geq 0$. Let μ_b and μ_t be the probability distributions for the generalized source and target domain, respectively. Then the risk on the target distribution μ_t follows

$$\epsilon_{\mu_t}(\phi^{S_b}) \leq \epsilon_{\mu_b}(\phi^{S_b}) + 2M \cdot \mathcal{W}_1\left(\mu_b, \mu_t\right) + \lambda, \tag{16}$$

where $\lambda = \min_{\phi:\mathcal{X}\to[0,1], \|\phi\|_{\text{Lip}} \leq M} (\epsilon_{\mu_t}(\phi) + \epsilon_{\mu_b}(\phi))$.
The first term is the risk on the barycenter distribution μ_b. The second term shows that the divergence between the barycenter distribution and target distribution, measured by the Wasserstein distance (of order 1). This theorem shows that the generalization risk on the target domain is affected by the Wasserstein distance between the barycenter distribution and the target distribution, which represents the gap between the generalized source domain and the target domain.

By applying the concentration property of the Wasserstein distance [57], we can measure the generalization risk based on empirical Wasserstein distances similar to Theorem 3 in [58]. Under the assumption of Theorem 2, if the two probability distributions μ_b and μ_t satisfy $T_1(\xi)$ inequality [57], then for any $d' > d$ and $\xi' < \xi$, there exists some constant N_0 depending on d' such that for any $\varepsilon > 0$ and $\min(n_b, n_t) \geq N_0 \max\left(\varepsilon^{-(d'+2)}, 1\right)$, with probability at least $1-\varepsilon$ the following holds for the risk on the target domain

$$\begin{aligned} \epsilon_{\mu_t}(\phi^{S_b}) \leq & \epsilon_{\mu_b}(\phi^{S_b}) + 2M\mathcal{W}_1\left(\widehat{\mu}_b, \widehat{\mu}_t\right) + \lambda \\ & + 2M\sqrt{2\log\left(\frac{1}{\varepsilon}\right)/\xi'}\left(\sqrt{\frac{1}{n_b}} + \sqrt{\frac{1}{n_t}}\right). \end{aligned}$$

Here d denotes the dimension of the feature space. The last term illustrates the importance of getting more labeled samples from the generalized source domain. This result show that reducing the Wasserstein distance between the barycenters and target distributions will lead to tighter upper bound for the risk of the learned model on the target domain. Therefore, it provides a theoretical motivation to our design of the test-time adaptation, which reduces such domain gap by optimal transport. Details of the proof could be found in the supplementary material.

IV. Experiments

A. Datasets

To evaluate the effectiveness of our proposed domain generalization framework, we conduct experiments on three datasets: the VLCS [59] dataset, the PACS [60] dataset, and the Rotated MNIST [61] dataset.

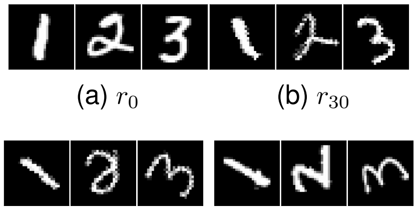
VLCS dataset This domain generalization benchmark contains images from four image classification datasets: PASCAL VOC2007 (V), LabelMe (L), Caltech-101 (C), and SUN09 (S), denoted as domains D_V, D_L, D_C, and D_S, respectively [62]. There are five common categories: bird, car, chair, dog and person.

PACS dataset The PACS dataset contains images of four domains: Photos (P), Art painting (A), Cartoon (C) and Sketch (S) [60]. There are in total 7 types of objects in this classification task, i.e., dog, elephant, giraffe, guitar, horse, house, and person.

Rotated MNIST dataset We constructed the Rotated MNIST dataset with four domains, r_0, r_{30}, r_{60} and r_{90} following the common settings [61]. r_0 denotes the domain containing original images from the MNIST dataset, and we rotated each image in the original MNIST dataset by 30, 60 and 90 degrees clockwise, respectively to generate the dataset of r_{30}, r_{60} and r_{90}. Some example images are shown in Figure 3. We randomly sampled among digits $[1, 2, 3]$.

B. Experimental Configuration

We evaluate each method on the multi-domain datasets via the leave-one-domain-out experiments, i.e., we train a model based on the source domains and test on the hold-out unseen target domain. For example, when the target domain is D_V, then the transfer direction is from three source domains to a target domain, i.e., $D_L, D_C, D_S \to D_V$, and the average of test accuracies of four cross-domain experiments is taken as the average generalization result.



(a) r_0 (b) r_{30}

(c) r_{60} (d) r_{90}

Fig. 3. Visualization of example images from four domains of the Rotated MNIST dataset with rotation angles of 0°, 30°, 60°, 90°.

We mainly consider the scenario when we have only limited labeled data from the source domains. For each domain, we randomly select some images to form the training set and validation set for the cross-domain classification. For the training set, we set the number of training images per category per domain to be a number in the set $\{2, 3, 5, 10, 15, 20, 25\}$. We randomly sample 10 images per category for the validation set of each source domain. We repeat the above sampling process 5 times for all datasets, so that the experiments are based on 5 trials. The average results of all 5 trials are finally reported.

Frozen features pretrained on neural networks are taken as our input. For the Rotated MNIST dataset, the Resnet-18 [63] pretrained on the ImageNet is used to extract 512-dimensional features as the inputs. For the VLCS dataset, the pretrained 4096-dimensional DeCAF features [64] are employed as the inputs of our algorithm following previous works [22], [65]. For the PACS dataset, we use the ImageNet pre-trained AlexNet [66] as the backbone network to extract the 9216-dimensional features. The Wasserstein distance of order 2 is used for all experiments, and calculated with the POT package [56].

We conduct a comparative experiments of the proposed WDRDG pipeline against some traditional baselines (CIDG [18], MDA [19]) that base on learning the domain-invariant feature transformation. DrkNN [67] is compared since it also tackles the challenging problem of learning a robust classifier from a few samples. For CIDG, MDA, and DrkNN, a simple 1-NN is adopted as a classifier. The k-NN learned on the pretrained feature is compared as another simple baseline. For the proposed WDRDG, we use the CVXPY package [68] with the MOSEK [69] solver to solve the constrained distributionally robust optimization problem (11). The discriminability threshold δ is taken as a hyperparameter chosen via validation.

To alleviate the dependence on hyperparameters learned from a limited validation dataset, we further extend our approach by integrating a differentiable optimization layer [70] to solve the Wasserstein distributionally robust optimization problem in an end-to-end trainable neural networks architecture, following [35]. Instead of relying on the static optimization with fixed hyperparameters using the convex solver, the solution to the optimization problem can now be backpropagated, allowing for dynamic updating of parameters during optimization. In this extended version denoted as **WDRDG++**, we implement the differentiable convex optimization layers based on cvxpylayers package [71] to make differentiable optimization. Two learnable parameters, uncertainty set radius θ_k and discriminability threshold parameter δ become trainable now. The radius in equation (7) is used as the initialization for the parameter θ_k.

Additionally, we evaluate our pipeline, by benchmarking it against some state-of-the-art methods, including MLDG [13], ADA [7], GroupDRO [72], VREx [73], and EQRM [74]. For these methods, a simple multi-layer perceptron network is adopted to be the trainable classifier on the pretrained feature.

C. Results and Discussion

In this section, we present the results for domain generalization on all three datasets. When each domain serves as the target domain, the results are shown in Figure 4, with the plotted lines representing the average performance over 5 trials and the shaded area representing the corresponding standard deviation.

For the VLCS dataset, we report the results in the first row in Figure 4. In all four cases when each domain serves as the unseen target domain, our method achieves better classification accuracy and standard deviation than other methods when the training sample size for each class is very few, i.e., 2, 3, or 5. The advantage of WDRDG over MLDG then levels off as the sample size reaches to over 10 per class. The performance improvement of WDRDG against MLDG reaches as high as 6.53%, 11.89%, 46.79%, 22.54% with only 2 training samples for each class when the target domain is PASCAL VOC2007, LabelMe, Caltech-101, and SUN09, respectively. For WDRDG++, it achieves at least 21.64%, 4.61%, 15.05%, 22.43% better performance than other sota baselines when there are only 2 samples per class. These results confirm that our method is efficient for few-shot cases.

The second row of Figure 4 reports the classification accuracy results for the PACS dataset. The proposed WDRDG achieves the best results in accuracy and standard deviation when the target domain is Art Painting, Cartoon, or Sketch using different training sample size, and MLDG outperforms WDRDG when the target domain is Photos with the sample size 15 for each class. WDRDG outperforms MLDG by up to 19.81%, 20.95%, 18.68%, 20.35% for each target domain when the training sample size is 2, while WDRDG++ outperforms other baselines by at least 9.24%, 14.65%, 35.26%, 12.39%. This validates the effect of our method when the training sample size is limited. The improvement of WDRDG over other methods on the PACS dataset is relatively larger compared with the improvements on the VLCS dataset. This improvement is especially obvious over MDA and CIDG when the target domain is Sketch, shown in the fourth column of the second row in Figure 4. This may because that the differences among domains are greater in PACS where the image styles are obviously different compared with in VLCS, where samples from different domains are real-world images collected from different perspectives or scales. This demonstrates that our WDRDG could better handle scenarios with larger unseen domain shift.

The results for the Rotated MNIST dataset in the third row of Figure 4 also yield similar conclusions. As the training

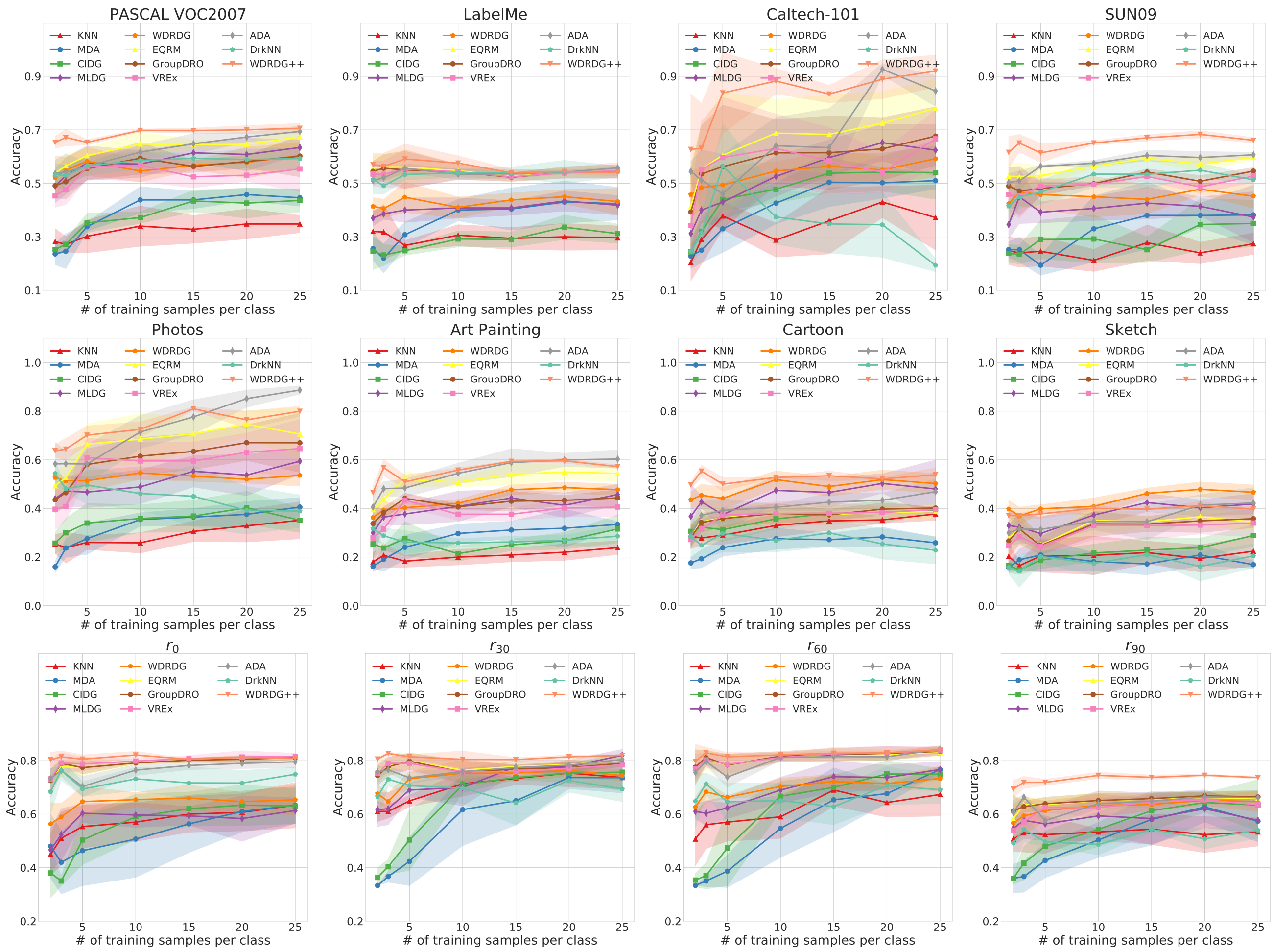


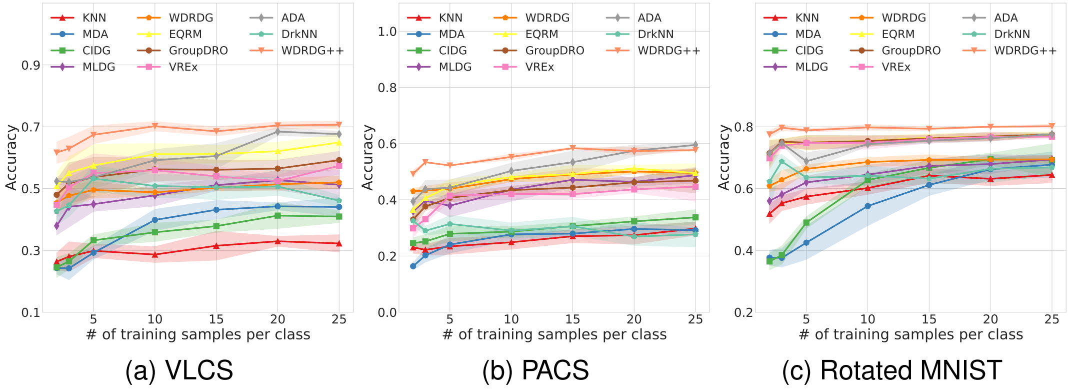
Fig. 4. Performance comparison for the VLCS, PACS and Rotated MNIST dataset under different size of training samples per class. Each row shows the results for a dataset, and each column shows the generalization result for a certain target domain. Average performance of five methods are represented by different colors, and the corresponding shadow shows the standard deviation of 5 experimental trials. Our WDRDG framework outperforms KNN, MDA and CIDG with higher accuracy and smaller standard deviation. Also, it has more advantage over MLDG especially when the source training sample size is limited. For example, WDRDG outperforms MLDG by up to 46.79% when the target domain is Caltech-101 in the VLCS dataset, by up to 20.95% for target domain Art Painting in the PACS dataset, and by up to 20.71% for target domain r_0 in the Rotated MNIST dataset with training sample size of 2 for each class.

sample size increases, almost all methods converges to the same accuracy for different target domain. When the training sample size is smaller, i.e., the training sample per class for each source domain is $2, 3, 5$, the advantage of our proposed framework is more obvious. WDRDG outperforms MLDG by 20.71%, 9.73%, 2.73%, 3.66% when the training sample size is 2 for each class for target domain r_0, r_{30}, r_{60}, and r_{90}, respectively. WDRDG++ outperforms others by at least 9.5%, 6.29%, 2.83%, 13.49%. When the training sample size is big, e.g., the training sample per class for each source domain is 25, even simple KNN method performs well. This is consistent with the analysis in the above two datasets.

Figure 5 reports the average performance of different target domains on the three datasets. Overall, our method is the most stable under different numbers of training samples, with narrower shadow band of standard deviation. As the size of training samples gets bigger, all methods have the tendency of performing better. In most cases, WDRDG++ achieves the best average performance under different training sample size compared with other methods with smaller standard deviation. In addition, our method shows more advantage over others in few-shot settings. When given training samples are limited to less than 10 (i.e., 2, 3, 5 in our experiments) per class, WDRDG++ provides at least 17.72%, 24.92%, 8.52% better generalization ability than others on the VLCS, PACS and Rotated MNIST dataset, respectively. We also did further exploration of how larger training sample sizes impact the generalization capability. More details could be found in the the supplementary material.

D. Ablation Study for the Test-time Adaptation

To explore the effectiveness of the test-time adaptation based on optimal transport, we compare WDRDG with and without this adaptive inference module. For the non-adaptive inference, the nearest neighbor for any test sample from the target domain is found by the simple 1-NN over barycenter samples. We compare the results of using training sample size of $5, 10, 15$ per class for each source domain.



(a) VLCS (b) PACS (c) Rotated MNIST

Fig. 5. Average generalization performance of different methods on the VLCS, PACS and Rotated MNIST dataset. As the training sample size increases, all methods obtain better performance. Our WDRDG++ framework outperforms other baselines, especially in few-shot settings. When the sample size is less than 10 per class, WDRDG++ provides at least 17.72%, 24.92%, 8.52% better generalization ability than others on the VLCS, PACS and Rotated MNIST dataset, respectively.

TABLE I
THE EFFECT OF THE OPTIMAL TRANSPORT-BASED TEST-TIME ADAPTATION (TTA) FOR ADAPTIVE INFERENCE. ADDING THE TTA MODULE RESULTS IN BETTER PERFORMANCE.

#training sample	Method	V	L	C	S	Average	P	A	C	S	Average	r_0	r_{30}	r_{60}	r_{90}	Average
5	WDRDG (w/o. TTA)	0.516	0.372	0.554	0.356	0.450	0.504	0.350	0.471	0.237	0.391	0.593	0.640	0.577	0.553	0.591
	WDRDG (w. TTA)	0.582	0.448	0.494	0.458	**0.496**	0.514	0.403	0.441	0.399	**0.439**	0.647	0.732	0.663	0.613	**0.664**
10	WDRDG (w/o. TTA)	0.540	0.402	0.516	0.334	0.448	0.559	0.374	0.480	0.259	0.418	0.567	0.690	0.647	0.557	0.615
	WDRDG (w. TTA)	0.546	0.410	0.546	0.450	**0.488**	0.556	0.421	0.519	0.409	**0.476**	0.654	0.753	0.703	0.633	**0.686**
15	WDRDG (w/o. TTA)	0.510	0.378	0.67	0.39	0.487	0.549	0.404	0.491	0.251	0.424	0.567	0.653	0.677	0.533	0.608
	WDRDG (w. TTA)	0.568	0.438	0.564	0.440	**0.503**	0.533	0.477	0.475	0.462	**0.487**	0.660	0.753	0.721	0.636	**0.693**

From the results in Table I, we can make several observations. Our WDRDG framework with the adaptive inference module results in better average performance for all three datasets, with up to 10.22% higher mean accuracy for the VLCS dataset with 5 training samples per class, 14.86% performance improvement for the PACS dataset with 15 training samples per class, and 13.98% improvements for the Rotated MNIST dataset with 15 training samples per class. Note that when the target domain is Sketch on the PACS dataset, the improvements are especially obvious compared with other targets, reaching 68.35%, 57.92%, and 84.06% when the training sample size for each class is $5, 10, 15$, respectively. Similar results could be found on the Rotated MNIST dataset when the target domain is r_0 or r_{90} when the training sample size per class is 10 or 15, with up to 19.32% performance improvements. This improvement is more obvious compared with other targets r_{30} or r_{60}, which obtains up to 15.31% performance improvements using the adaptive inference module. One thing they share in common is these target domains are more different with given source domains, which shows larger unseen distribution shifts. Similar experiments are conducted on Rotated MNIST dataset with regard to WDRDG++, as show in Table II, TTA brings significant performance improvements for WDRDG++. This validates the robustness of our adaptive inference module for even harder, unseen target domains.

E. Impact of the Discriminability Threshold δ

We conducted analysis on the Rotated MNIST dataset to evaluate the impact of the threshold parameter δ on the robustness of our algorithm's performance, particularly concerning varying sample sizes. Specifically, we recorded the optimal δ values with the uncertainty radius parameter fixed. The results demonstrate that as the sample size increases, the required δ stabilizes and decreases, indicating that the necessity for strict δ constraints diminishes with larger sample sizes. This

TABLE II
THE EFFECT OF THE OPTIMAL TRANSPORT-BASED TEST-TIME ADAPTATION (TTA) ON THE ROTATED MNIST DATASET. TTA MODULE PROVIDES PERFORMANCE IMPROVEMENTS FOR WDRDG++.

# sample	Method	r_0	r_{30}	r_{60}	r_{90}	Average
5	WDRDG++ (w/o. TTA)	0.738	0.771	0.753	0.576	0.710
	WDRDG++ (w. TTA)	0.806	0.814	0.814	0.719	**0.788**
10	WDRDG++ (w/o. TTA)	0.774	0.728	0.771	0.600	0.718
	WDRDG++ (w. TTA)	0.821	0.805	0.821	0.745	**0.798**
15	WDRDG++ (w/o. TTA)	0.779	0.758	0.755	0.603	0.724
	WDRDG++ (w. TTA)	0.806	0.804	0.828	0.737	**0.794**

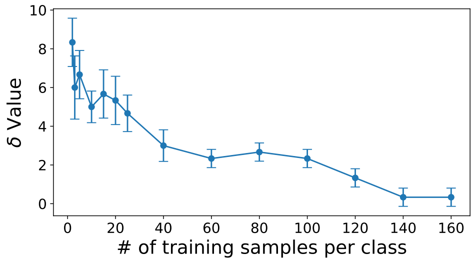


Fig. 6. Optimal δ values for different sample sizes. As the sample size increases, the optimal δ decreases, indicating more strict constraint requirements for the challenging scenarios with few training samples.

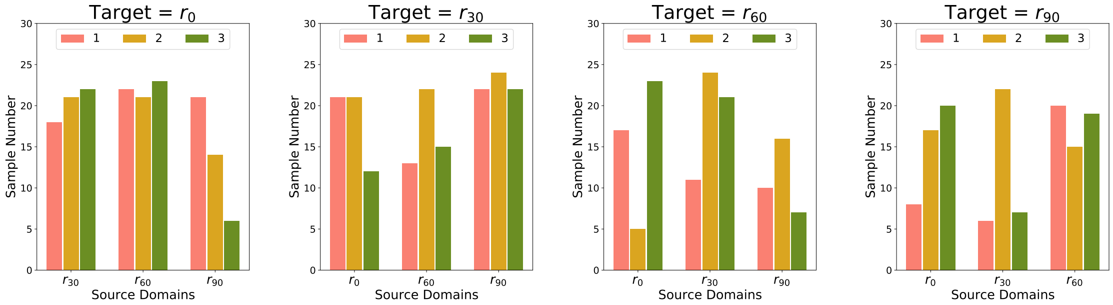


Fig. 7. Visualization of random sample size for each class in source domains when a different domain serves as the target domain in the Rotated MNIST dataset. For each source domain, the number of samples for different classes are shown in different colors. There are cases when different classes have similar sample number, e.g., Class 1 and 2 of source domain r_0 when target domain is r_{30}, and also cases when different classes have quite different number of samples, e.g., in source domain r_{90} when target domain is r_0.

trend can be attributed to smaller sample sizes producing less accurate barycenters, which are more prone to overlap, thus necessitating a larger optimal δ. With more training samples, the uncertainty set becomes a better estimate of the true distribution, reducing the importance of the distinguishable threshold parameter.

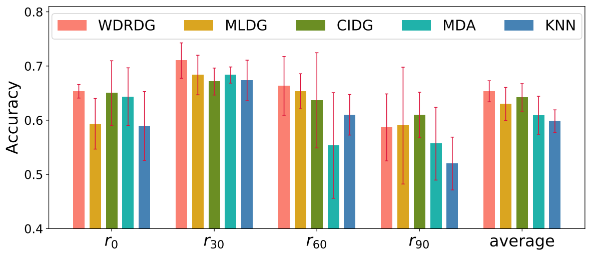


Fig. 8. The performance of WDRDG and four compared methods on the Rotated MNIST dataset with different class prior distributions across source domains. WDRDG outperforms other baselines by at least 0.51%, 3.90%, 1.53% when the target domain is r_0, r_{30}, r_{60}, respectively, and achieves similar accuracies with MLDG but with smaller deviation when the target domain is r_{90}.

F. Analysis of Imbalanced Classes among Source Domains

In previous experiments, we actually assume the training sample size per class in the source domains are the same under the setting of no class prior distribution shift, i.e., the distribution of $P(Y)$ is the same across all source domains. To show the feasibility of extending our framework to scenarios with class prior distribution shift, we further conduct experiments when the categories in source domains are imbalanced, i.e., there are shifts among $P(Y)$ of different domains.

We randomly sample the training sample size for each class from $[5, 25)$ on the Rotated MNIST dataset here. The distribution of sample number for each class when each domain is chosen as the target domain is shown in Figure 7. There are cases when different classes have similar sample number, e.g., in source domain r_{90} when the target domain is r_{30}, or in source domain r_{60} when the target domain is r_0. In other source domains, different classes may have quite different number of samples, e.g., in source domain r_{90} when target domain is r_0, or in source domain r_0 when target domain is r_{60}. We compare our framework WDRDG with other methods, and the results are shown in Figure 8. When the target domain is r_{90}, our method achieves similar accuracies with MLDG but with smaller deviation, while in other cases WDRDG outperforms other baselines by at least 0.51%, 3.90%, 1.53% when the target domain is r_0, r_{30}, r_{60}, respectively. Our framework outperforms other methods on average with smaller standard deviation, which validates the generalization ability of our framework when the source domains have class prior distribution shift.

V. Conclusion

In this research, we proposed a novel framework for domain generalization to enhance model robustness when labeled training data of source domains are limited. We formulated the distributional shifts for each class with class-specific Wasserstein uncertainty sets and optimized the model over the worst-case distributions residing in the uncertainty sets via distributionally robust optimization. To reduce the difference between source and target domains, we proposed a test-time domain adaptation module through optimal transport to make adaptive inference for unseen target data. We found that our domain generalization framework with this adaptive inference module works better when target domains are more different compared with source domains. Experimental results on Rotated MNIST, PACS and VLCS datasets demonstrate that our proposed WDRDG framework could learn a robust model for unseen target domains based on limited source data, and we also showed that its advantage is more obvious in few-shot settings. To perfect this work in the future, we would study the usage of class priors in constructing more realistic uncertainty sets, and explore measurable relationship among source domains to better leverage the source distributions to model possible target distributions.

Acknowledgment

This work is supported in part by the Natural Science Foundation of China (Grant 62371270) and Shenzhen Key Laboratory of Ubiquitous Data Enabling (No.ZDSYS20220527171406015).

References

[1] G. Blanchard, G. Lee, and C. Scott, "Generalizing from several related classification tasks to a new unlabeled sample," *Advances in neural information processing systems*, vol. 24, pp. 2178–2186, 2011.

[2] R. Gong, W. Li, Y. Chen, and L. V. Gool, "Dlow: Domain flow for adaptation and generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2477–2486.

[3] Y. Shi, X. Yu, K. Sohn, M. Chandraker, and A. K. Jain, "Towards universal representation learning for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6817–6826.

[4] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, "Learning to generate novel domains for domain generalization," in *European conference on computer vision*. Springer, 2020, pp. 561–578.

[5] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Learning generalisable omni-scale representations for person re-identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[6] X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong, "Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2100–2110.

[7] R. Volpi, H. Namkoong, O. Sener, J. C. Duchi, V. Murino, and S. Savarese, "Generalizing to unseen domains via adversarial data augmentation," in *NeurIPS*, 2018.

[8] R. Shao, X. Lan, J. Li, and P. C. Yuen, "Multi-adversarial discriminative deep domain generalization for face presentation attack detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 023–10 031.

[9] Y. Balaji, S. Sankaranarayanan, and R. Chellappa, "Metareg: Towards domain generalization using meta-regularization," *Advances in Neural Information Processing Systems*, vol. 31, pp. 998–1008, 2018.

[10] G. Marzinotto, G. Damnati, F. Béchet, and B. Favre, "Robust semantic parsing with adversarial learning for domain generalization," *arXiv preprint arXiv:1910.06700*, 2019.

[11] E. Stepanov and G. Riccardi, "Towards cross-domain pdtb-style discourse parsing," in *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)*, 2014, pp. 30–37.

[12] D. Fried, N. Kitaev, and D. Klein, "Cross-domain generalization of neural constituency parsers," *arXiv preprint arXiv:1907.04347*, 2019.

[13] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[14] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *International Conference on Machine Learning*. PMLR, 2013, pp. 10–18.

[15] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1414–1430, 2016.

[16] G. Blanchard, A. A. Deshmukh, U. Dogan, G. Lee, and C. Scott, "Domain generalization by marginal transfer learning," *arXiv preprint arXiv:1711.07910*, 2017.

[17] T. Grubinger, A. Birlutiu, H. Schöner, T. Natschläger, and T. Heskes, "Domain generalization based on transfer component analysis," in *International Work-Conference on Artificial Neural Networks*. Springer, 2015, pp. 325–334.

[18] Y. Li, M. Gong, X. Tian, T. Liu, and D. Tao, "Domain generalization via conditional invariant representations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[19] S. Hu, K. Zhang, Z. Chen, and L. Chan, "Domain generalization via multidomain discriminant analysis," in *Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 292–302.

[20] F. Zhou, Z. Jiang, C. Shui, B. Wang, and B. Chaib-draa, "Domain generalization with optimal transport and metric learning," *arXiv preprint arXiv:2007.10573*, 2020.

[21] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1406–1415.

[22] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5715–5725.

[23] H. Li, S. J. Pan, S. Wang, and 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.

[24] R. Gong, W. Li, Y. Chen, and L. V. Gool, "Dlow: Domain flow for adaptation and generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[25] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao, "Deep domain generalization via conditional invariant adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 624–639.

[26] M. M. Rahman, C. Fookes, M. Baktashmotlagh, and S. Sridharan, "Correlation-aware adversarial domain adaptation and generalization," *Pattern Recognition*, vol. 100, p. 107124, 2020.

[27] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and 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.

[28] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi, "Generalizing across domains via cross-gradient training," *arXiv preprint arXiv:1804.10745*, 2018.

[29] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, "Deep domain-adversarial image generation for domain generalisation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 13 025–13 032.

[30] Y. Wang, H. Li, and A. C. Kot, "Heterogeneous domain generalization via domain mixup," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3622–3626.

[31] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," *ICLR*, 2021.

[32] J. A. Bagnell, "Robust supervised learning," in *AAAI*, 2005, pp. 714–719.

[33] A. Sinha, H. Namkoong, R. Volpi, and J. Duchi, "Certifying some distributional robustness with principled adversarial training," *arXiv preprint arXiv:1710.10571*, 2017.

[34] R. Gao, L. Xie, Y. Xie, and H. Xu, "Robust hypothesis testing using wasserstein uncertainty sets." in *NeurIPS*, 2018, pp. 7913–7923.

[35] S. Zhu, L. Xie, M. Zhang, R. Gao, and Y. Xie, "Distributionally robust k-nearest neighbors for few-shot learning," *arXiv e-prints*, pp. arXiv–2006, 2020.

[36] H. Rahimian and S. Mehrotra, "Distributionally robust optimization: A review," *arXiv preprint arXiv:1908.05659*, 2019.

[37] A. Ben-Tal, D. Den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen, "Robust solutions of optimization problems affected by uncertain probabilities," *Management Science*, vol. 59, no. 2, pp. 341–357, 2013.

[38] J. Duchi, P. Glynn, and H. Namkoong, "Statistics of robust optimization: A generalized empirical likelihood approach," *arXiv preprint arXiv:1610.03425*, 2016.

[39] H. Namkoong and J. C. Duchi, "Stochastic gradient methods for distributionally robust optimization with f-divergences," *Advances in neural information processing systems*, vol. 29, 2016.

[40] J. C. Duchi and H. Namkoong, "Learning models with uniform performance via distributionally robust optimization," *The Annals of Statistics*, vol. 49, no. 3, pp. 1378–1406, 2021.

[41] J. Blanchet, Y. Kang, and K. Murthy, "Robust wasserstein profile inference and applications to machine learning," *Journal of Applied Probability*, vol. 56, no. 3, pp. 830–857, 2019.

[42] J. Lee and M. Raginsky, "Minimax statistical learning with wasserstein distances," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[43] P. Mohajerin Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations," *Mathematical Programming*, vol. 171, no. 1, pp. 115–166, 2018.

[44] M. Staib and S. Jegelka, "Distributionally robust deep learning as a generalization of adversarial training," in *NIPS workshop on Machine Learning and Computer Security*, vol. 1, 2017.

[45] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," in *International Conference on Machine Learning*. PMLR, 2013, pp. 819–827.

[46] D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh, "Wasserstein distributionally robust optimization: Theory and applications in machine learning," in *Operations Research & Management Science in the Age of Analytics*. INFORMS, 2019, pp. 130–166.

[47] C. Villani, *Optimal transport: old and new*. Springer, 2009, vol. 338.

[48] G. Peyré, M. Cuturi *et al.*, “Computational optimal transport: With applications to data science,” *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[49] P. J. Huber, “A robust version of the probability ratio test,” *Annals of Mathematical Statistics*, vol. 36, no. 6, pp. 1753–1758, 1965.

[50] Z.-H. Zhou and X.-Y. Liu, “Training cost-sensitive neural networks with methods addressing the class imbalance problem,” *IEEE Transactions on knowledge and data engineering*, vol. 18, no. 1, pp. 63–77, 2005.

[51] C. Scott, “Calibrated asymmetric surrogate losses,” *Electronic Journal of Statistics*, vol. 6, pp. 958–992, 2012.

[52] Y. S. Aurelio, G. M. de Almeida, C. L. de Castro, and A. P. Braga, “Learning from imbalanced data sets with weighted cross-entropy function,” *Neural processing letters*, vol. 50, no. 2, pp. 1937–1949, 2019.

[53] Z. Xu, C. Dan, J. Khim, and P. Ravikumar, “Class-weighted classification: Trade-offs and robust approaches,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 10 544–10 554.

[54] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, “Optimal transport for domain adaptation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1853–1865, 2016.

[55] J. Rabin, G. Peyré, J. Delon, and M. Bernot, “Wasserstein barycenter and its application to texture mixing,” in *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 2011, pp. 435–446.

[56] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier *et al.*, “Pot: Python optimal transport,” *Journal of Machine Learning Research*, vol. 22, no. 78, pp. 1–8, 2021.

[57] F. Bolley, A. Guillin, and C. Villani, “Quantitative concentration inequalities for empirical measures on non-compact spaces,” *Probability Theory and Related Fields*, vol. 137, no. 3-4, pp. 541–593, 2007.

[58] J. Shen, Y. Qu, W. Zhang, and Y. Yu, “Wasserstein distance guided representation learning for domain adaptation,” in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[59] C. Fang, Y. Xu, and D. N. Rockmore, “Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1657–1664.

[60] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, “Deeper, broader and artier domain generalization,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5542–5550.

[61] M. Ghifary, W. Bastiaan Kleijn, M. Zhang, and 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.

[62] A. Torralba and A. A. Efros, “Unbiased look at dataset bias,” in *CVPR 2011*. IEEE, 2011, pp. 1521–1528.

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

[64] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” in *International conference on machine learning*. PMLR, 2014, pp. 647–655.

[65] Q. Dou, D. Coelho de Castro, K. Kamnitsas, and B. Glocker, “Domain generalization via model-agnostic learning of semantic features,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[66] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.

[67] S. Zhu, L. Xie, M. Zhang, R. Gao, and Y. Xie, “Distributionally robust weighted k-nearest neighbors,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 29 088–29 100, 2022.

[68] S. Diamond and S. Boyd, “Cvxpy: A python-embedded modeling language for convex optimization,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

[69] M. ApS, “Mosek optimization toolbox for matlab,” *User’s Guide and Reference Manual, Version*, vol. 4, no. 1, 2019.

[70] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, “Differentiable convex optimization layers,” *Advances in neural information processing systems*, vol. 32, 2019.

[71] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and Z. Kolter, “Differentiable convex optimization layers,” in *Advances in Neural Information Processing Systems*, 2019.

[72] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, “Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization,” *arXiv preprint arXiv:1911.08731*, 2019.

[73] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville, “Out-of-distribution generalization via risk extrapolation (rex),” in *International Conference on Machine Learning*. PMLR, 2021, pp. 5815–5826.

[74] C. Eastwood, A. Robey, S. Singh, J. Von Kügelgen, H. Hassani, G. J. Pappas, and B. Schölkopf, “Probable domain generalization via quantile risk minimization,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 17 340–17 358, 2022.