Generalizing to Unseen Domains With Wasserstein Distributional Robustness Under Limited Source Knowledge

Jingge Wang, Student Member, IEEE, Liyan Xie, Yao Xie, Senior Member, IEEE, Shao-Lun Huang, Member, IEEE, and Yang Li, Member, IEEE

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 inferences for target data. Experiments on the Rotated MNIST, PACS, and 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, the 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 shifts. 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 while 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], [3], [4], [5], [6], [7], [8], natural language processing [9], [10], [11], [12], and reinforcement learning [13], etc.

Numerous methods have been developed for learning a generalizable model by exploiting data from the available source domains, where the shifts across these domains are implicitly assumed to be representative of the target shift that we will meet at test time. Achieving good performances on all given domains thus results in a generalizable model [14]. The well-known approaches include learning domain-invariant feature representations through kernel functions [11], [15], [16], [17], [18], [19], [20], by distribution alignment [21], [22], [23], and in an adversarial manner [8], [24], [25], [26], [27]. 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] divide the source domains into non-overlapping meta-train and meta-test domains, which still 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 be simulated by enhancing the diversity of data based on some data augmentations techniques [28], by generating data in an adversarial manner [7], [29], [30], or by constructing sample interpolation [31], [32]. Learning with limited labeled original samples in this way will weaken their performance since the newly 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 generalization [1], which aims at learning a model that generalizes well to target domains based on available training data from multiple source domains while 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], [3], [4], [5], [6], [7], [8], natural language processing [9], [10], [11], [12], and reinforcement learning [13], etc.

Numerous methods have been developed for learning a generalizable model by exploiting data from the available source domains, where the shifts across these domains are implicitly assumed to be representative of the target shift that we will meet at test time. Achieving good performances on all given domains thus results in a generalizable model [14]. The well-known approaches include learning domain-invariant feature representations through kernel functions [11], [15], [16], [17], [18], [19], [20], by distribution alignment [21], [22], [23], and in an adversarial manner [8], [24], [25], [26], [27]. 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] divide the source domains into non-overlapping meta-train and meta-test domains, which still 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 be simulated by enhancing the diversity of data based on some data augmentations techniques [28], by generating data in an adversarial manner [7], [29], [30], or by constructing sample interpolation [31], [32]. Learning with limited labeled original samples in this way will weaken their performance since the newly 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 generalization [1], which aims at learning a model that generalizes well to target domains based on available training data from multiple source domains while 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], [3], [4], [5], [6], [7], [8], natural language processing [9], [10], [11], [12], and reinforcement learning [13], etc.

Numerous methods have been developed for learning a generalizable model by exploiting data from the available source domains, where the shifts across these domains are implicitly assumed to be representative of the target shift that we will meet at test time. Achieving good performances on all given domains thus results in a generalizable model [14]. The well-known approaches include learning domain-invariant feature representations through kernel functions [11], [15], [16], [17], [18], [19], [20], by distribution alignment [21], [22], [23], and in an adversarial manner [8], [24], [25], [26], [27]. 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] divide the source domains into non-overlapping meta-train and meta-test domains, which still 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 be simulated by enhancing the diversity of data based on some data augmentations techniques [28], by generating data in an adversarial manner [7], [29], [30], or by constructing sample interpolation [31], [32]. Learning with limited labeled original samples in this way will weaken their performance since the newly 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) [33]. 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 [34], [35], [36], [37]. The uncertainty set contains distributions that belong to a non-parametric distribution family, which contains distributions centered around the empirical training distributions defined via some divergence metrics, e.g., Kullback-Leibler divergence [33], other \( f \)-divergences [38], [39], [40], [41], or Wasserstein distance [34], [42], [43], [44], [45], etc. These pre-defined distance constraints 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. [41] solve the DRO to learn a model within an \( f \)-divergence uncertainty set and learn 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 [41] 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 has 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 [46], under the assumption that there is no shift in the class prior distribution, i.e., the distribution \( P(Y) \) stays consistent across all domains. Furthermore, we also illustrate how our research can be readily expanded to situations that involve a shift in the class prior distributions. To be more specific, we encode the domain perturbations of each class within a class-specific Wasserstein uncertainty set. Compared with the Kullback-Leibler divergence, the Wasserstein distance is well-known for its ability to measure divergence between distributions defined on different probability spaces, 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 [47], tractability results for DRO with multiple Wasserstein uncertainty sets for each class are also available [35].

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 [44]. A richer uncertainty set may contain more true target distributions with higher confidence but comes with a more conservative and less practical solution. A more precise uncertainty set incentivizes higher complexity and more difficult solutions. 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 inferences 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 a heuristic magnitude of uncertainty, we design a constraint that balances the 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 compared with the state-of-the-art.

II. PRELIMINARIES AND PROBLEM SETUP

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

Given training samples \( \{(x_1, y_1), \ldots, (x_n, y_n)\} \) 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:

\[
\hat{Q}_k := \frac{1}{|i: y_i = k|} \sum_{i=1}^n \delta_{x_i, 1\{y_i = k\}}, \quad k = 1, \ldots, K.
\]

(1)

Here, \( \delta \) indicates a Dirac measure centered at \( x \) and \( 1\{\cdot\} \) is the indicator function. Therefore, \( \hat{Q}_k \) can be viewed as the empirical distribution for training samples within the class \( k \). In light of [35], [36], the test distribution of each class is likely to be distributions centered around the empirical class-conditional
distribution $\tilde{Q}_k$, within the uncertainty set defined via distribution divergence metrics, for example, the Wasserstein distance.

The Wasserstein distance \cite{48, 49} of order $p$ between any two distributions $P$ and $Q$, is defined as:

$$\mathcal{W}_p(P, Q) = \left( \min_{\gamma \in \Gamma(P, Q)} \mathbb{E}_{(x, x') \sim \gamma} \left[ \|x - x'\|^p \right] \right)^{1/p}, \quad (2)$$

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, the test distribution of each class $k$ belongs to the following distribution set:

$$\mathcal{P}_k = \left\{ P_k \in \mathcal{P}(X) : \mathcal{W}_2 \left( P_k, \tilde{Q}_k \right) \leq \theta_k \right\}, \quad k = 1, \ldots, K, \quad (3)$$

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

$$\min_{\phi : X \to \Delta_K} \max_{P_1, \ldots, P_K} \Psi \left( \phi; P_1, \ldots, P_K \right), \quad (4)$$

where $\Psi(\phi; P_1, \ldots, P_K)$ is the total risk of the classification using $\phi$ on distributions $P_1, \ldots, P_K$. The inner maximum problem refers to the worst-case risk over uncertainty sets $P_1, \ldots, P_K$. Suppose $(\phi^{*}; P_1^{*}, \ldots, P_K^{*})$ is an optimal solution pair to the saddle-point problem (4), then $P_1^{*}, \ldots, P_K^{*}$ are called the least favorable distributions (LFDs) \cite{50}, and $\phi^{*}$ 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., $1 - \phi_k(x)$ for any sample $x$ with real label $k$. Specially, when assuming the simple case with equal class prior distributions $P(y = k) = 1/K, k = 1, \ldots, K$ for all classes, the total risk of misclassifying data from all $K$ classes becomes:

$$\Psi (\phi; P_1, \ldots, P_K) = \sum_{k=1}^{K} \mathbb{E}_{x \sim P_k} [1 - \phi_k(x)]. \quad (5)$$

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 \cite{51, 52}. One of the most natural approaches is to incorporate the class prior distributions $P(y = k)$ of each class into the risk function \cite{53, 54} as follows:

$$\Psi (\phi; P_1, \ldots, P_K) = \sum_{k=1}^{K} P(y = k) \mathbb{E}_{x \sim P_k} [1 - \phi_k(x)], \quad (6)$$

which is a general form of (5).

In domain generalization problems, we have access to $R$ source domains $\{D^{s_i}_{r}, r = 1, \ldots, R\}$, with training samples $\{(x_1^{(r)}, y_1^{(r)}), \ldots, (x_{n_{r}}^{(r)}, y_{n_{r}}^{(r)})\}$ from the $r$-th source domain $D^{s_r}$, drawn i.i.d from the joint distribution $D^{s_r}$ on $X \times Y$. The goal is to learn a robust classifier that performs well on the unseen target domain $D^{t}$, which contains instances from the joint distribution $P^{t}$. For each class $k$, denote the empirical class-conditional distributions in source domain $D^{s_r}$ and target domain $D^{t}$ as $\hat{Q}^{s_r}_k$ and $\hat{Q}^{t}_k$, 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 $\hat{Q}^{s_r}_k$ from multiple source domains, which will be 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 Fig. 1(a), and the process of distributionally robust optimization is shown in Fig. 1(b). The adaptive inference for the target domain is shown in Fig. 1(c). 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. Fig. 1(a) illustrates the construction of uncertainty sets of two classes. Their closeness is further controlled by the parameter $\delta$ 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 Fig. 1(b). Fig. 1(c) shows the inference process for target samples, where optimal transport \cite{55} 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, respectively. Section III-C discusses the reformulation of the Wasserstein robust optimization. Adaptive inference for samples in the target domain is presented in Section III-D. In Section III-E, we further analyze the generalization bound of the proposed framework.

#### A. Construction of Uncertainty Sets

To measure distributionally divergence, we choose 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 utilize Wasserstein barycenter \cite{56} as the reference distribution, which is the average of multiple given distributions and is capable of leveraging the inherent geometric relations among them \cite{21}. Given empirical class-conditional distributions $\hat{Q}^{s_1}_k, \ldots, \hat{Q}^{s_R}_k$ for each class $k$ from $R$ different source domains, the Wasserstein barycenter for class $k$ is defined as:

$$B_k = \arg \min_{B_k} \sum_{r=1}^{R} \frac{1}{R} \mathcal{W}_2 \left( B_k, \hat{Q}^{s_r}_k \right), \quad k = 1, \ldots, K, \quad (7)$$

which could be a proxy of the reference distribution for each uncertainty set. Suppose each barycenter supports on $b$ samples uniformly, i.e., $B_k = \sum_{i=1}^{b} \frac{1}{b} \delta_{x_i^{(s_k)}}$, where $\{x_i^{(s_k)}\}_{i=1}^{b}$ are $b$ barycenter samples for class $k$ and $b$ is fixed, then (7) only optimizes over the locations $x_i^{(s_k)}$ of the uniform distribution on
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.

The feature space, which could be efficiently computed using POT package [57].

To ensure that the uncertainty sets are large enough to avoid misclassification for unseen target samples, the maximum of all Wasserstein distances between class-conditional distributions of each source domain \( \tilde{Q}_{sr}^r \) and the barycenter \( B_k^* \) is used as the radius for each class \( k \) as follows:

\[
\theta_k^* = \max_{r=1,\ldots,R} W_2^2(B_k^*, \tilde{Q}_{sr}^r). \tag{8}
\]

In this way, we can construct the Wasserstein uncertainty set \( P_k \) of radius \( \theta_k^* \) centered around \( B_k^* \) for each class \( k \) following (3) as:

\[
P_k = \left\{ P_k \in \mathcal{P}(\tilde{X}) : W_2(P_k, B_k^*) \leq \theta_k^* \right\}. \tag{9}
\]

Fig. 1(a) 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 (8) tends to be overly large, and the uncertainty sets of different classes may overlap with each other, leading to indistinguishable LFDs for the optimization problem (4). As shown in Fig. 2, overlap between each pair of class-specific uncertainty sets exists, 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 (4), which indirectly controls the discriminability of data from different classes. We thus add...
constraints to ensure significantly different LFDs that are discriminable, characterized by the Wasserstein distance between all pairs of LFDs \( (P^u, P^v) \) among the \( K \) categories as follows:
\[
\mathcal{W}_2(P^u, P^v) \geq \delta, \quad 1 \leq u < v \leq K,
\]
where \( \delta > 0 \) is the threshold that indicates the discriminability, which is a hyperparameter that can be tuned on the validation set. In this way, robustness is ensured by large enough Wasserstein uncertainty sets, and the threshold \( \delta \) further guarantees discriminability among the uncertainty sets.

C. Distributionally Robust Optimization

Incorporating the constraints (10) into the optimization problem (4), our goal becomes solving the following minimax problem:
\[
\min_{\phi : X \to \delta K} \max_{P_k \in P_k, \quad 1 \leq k \leq K} \Psi \left( \phi; P_1, \ldots, P_K \right). \tag{11}
\]
We now establish the following theorem, stating a convex approximation of problem (11).

**Theorem 1:** Suppose the Wasserstein barycenter \( B_\gamma \) for each class as defined in (7) is supported on \( b \) samples. Let \( S_b \) be the union of all the supported samples for all barycenters \( \{B_1, \ldots, B_K\} \), which contains \( K \) \( b \) samples in total, i.e., \( \{x_b^i\}, i = 1, \ldots, nb \). The class prior distributions of each class is denoted as \( P(y = k) \). Denote each distribution within the uncertainty set \( P_k \) as \( P_k \in P_k^b \). Let \( C \in \mathbb{R}_+^{nb \times nb} \) be the pairwise distance matrix of \( nb \) samples, \( C_{ij} = \| x_b^i - x_b^j \|^2 \), \( \gamma_k \in \mathbb{R}_+^{nb \times nb} \) be the coupling matrix between \( B_k^* \) and \( P_k \), and \( \beta_u,v \in \mathbb{R}_+^{nb \times nb} \) 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 (11) could be obtained by solving:
\[
\max_{P_u, \ldots, P_K \in P_k^b} \min_{\gamma_1, \ldots, \gamma_K \in C, \beta_u,v \in \mathbb{R}_+^{nb \times nb}} 1 - \sum_{i=1}^{nb} \max_{1 \leq k \leq K} \mathbb{P}(y = k) P_k \left( x_b^i \right)
\]
\[
\quad \text{s.t.} \quad \langle \gamma_k, C \rangle_F \leq (\theta_k^*)^2, \quad \langle \beta_u,v, C \rangle_F \leq \delta^2,
\]
\[
\gamma_k 1_{nb} = B_k^* 1_{nb}, \quad \beta_u,v 1_{nb} = P_v 1_{nb},
\]
\[
\forall 1 \leq k \leq K, 1 \leq u < v \leq K,
\]
and the optimal prediction function satisfies \( \phi_k^* (x_b^i) = P_k^* (x_b^i) / \sum_{k=1}^{K} P_k^* (x_b^i) \) for any \( x_b^i \in S_b \).

D. Adaptive Inference by Test-Time Adaptation

Since the barycenters are a weighted average of distributions of multiple source domains, the barycenter samples in the support set \( S_b \) can be viewed as samples from a generalized source domain denoted as \( D^b \). For any sample in domain \( D^b \), the likelihood that it is assigned to each class can be decided based on \( \phi_k^* (\cdot) \) by non-parametric inference method such as KNN [36] with regard to \( S_b \). Nevertheless, when making predictions for samples from an unseen target domain \( D^t \), the domain shift between \( D^b \) 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 \( \mu_b = \sum_{i=1}^{nb} \frac{1}{nb} \delta x_b^i \) and \( \mu_t = \sum_{j=1}^{nt} \frac{1}{nt} \delta x_t^j \) are the empirical marginal distributions of the feature vectors from the generalized source domain \( D^b \) and the target domain \( D^t \), respectively. Denote the coupling matrix of transporting from the target to the generalized source distribution using optimal transport [55] as \( \gamma = [\gamma_1, \ldots, \gamma_{nt}]^T \in \mathbb{R}_+^{nt \times nb} \), where each vector \( \gamma_j \in \mathbb{R}_+^{nb}, j = 1, \ldots, nt \), represents the transported mass from the \( j \)-th target sample to each of the \( nb \) barycenter samples. In most classical optimal transport-based domain adaptation methods, each target sample \( x_t^j \), \( j = 1, \ldots, nt \), is transported to \( \tilde{x}_b^j \) in the generalized source domain \( D^b \) by the barycenter mapping as follows:
\[
\tilde{x}_b^j = \sum_{i=1}^{nb} n_t \gamma_{j,i} x_b^i, \quad j = 1, \ldots, nt, \tag{13}
\]
then having its label inferred based on a classifier learned on the labeled samples. Instead of such a two-step process, we propose an equivalent single-step adaptive inference process. The following proposition states the equivalence and the detailed proof can be found in the supplementary material.

**Proposition 1:** Given the coupling matrix \( \gamma \in \mathbb{R}_+^{nt \times nb} \). Suppose we transport each target sample \( x_t^j \) from the empirical target distribution \( \mu_t = \sum_{j=1}^{nt} \frac{1}{nt} \delta x_t^j \) to the empirical distribution \( \tilde{\mu}_t = \sum_{i=1}^{nb} \frac{1}{nb} \delta x_b^i \) of the generalized source domain by the barycenter mapping, as in (13). The likelihood assignment for each class can be obtained by re-weighting the likelihood of all barycenter samples, i.e., \( \phi_k^* (x_t^j), x_t^j \in S_t \), using the transportation weight \( w(\tilde{x}_b^j, x_t^j) = n_t \gamma_{j,i} \). Then the resulting adaptive classifier is equivalent to directly re-weighting LFDs on the barycenter samples using the coupling matrix \( \gamma_{j,i} \), i.e., the equivalent classification result is:
\[
\Phi \left( x_t^j \right) = \arg \max_{1 \leq k \leq K} \sum_{i=1}^{nb} \gamma_{j,i} P_k^* \left( x_b^i \right). \tag{14}
\]
This proposition illustrates that the domain difference between the target domain and the 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 samples from solving (11) and based on Proposition 1, the classification of each target sample can be obtained by assigning each class with probability based on the re-weighted LFDs as \( \sum_{i=1}^{nb} n_t \gamma_{j,i} P_k^* (x_b^i) \). The final decision \( \Phi (x_t^j) \) is made by choosing the class that maximizes the probability.
Denote the LFDs for all classes as $P = [P_1, \ldots, P_K]$ $\in \mathbb{R}^{K \times n_b}$. Based on Proposition 1, the predicted class likelihood of each target sample $x^t_j$ can be rewritten as

$$
\phi (x^t_j) = \left[ \phi_1 (x^t_j), \ldots, \phi_K (x^t_j) \right] = \frac{\gamma_j^T P^T}{\gamma_j^T P^T 1_K},
$$

where $0 \leq \phi_k (x^t_j) \leq 1$, $\sum_{k=1}^{K} \phi_k (x^t_j) = 1$. The algorithm is summarized in Algorithm 1. By integrating the previously discussed optimal-transport-based adaptive inference, we establish our framework named Wasserstein Distributionally Robust Domain Generalization (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 (15), the classification decision for the test sample $x^t_j$ in the target domain is based on the following weighted average:

$$
\arg \max_{1 \leq k \leq K} \sum_{i=1}^{n_b} w \left( \tilde{x}^b_i, x^b_i \right) P^*_k \left( x^b_i \right).
$$

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

**Theorem 2:** Suppose the distributionally robust prediction function $\phi^{S_0}$ learned from the sample set $S_0$ is $M$-Lipschitz continuous for some $M \geq 0$. Let $\mu_b$ and $\mu_t$ be the probability distributions for the generalized source and target domain, respectively. Then the risk on the target distribution $\mu_t$ follows:

$$
\epsilon_{\mu_t} (\phi^{S_0}) \leq \epsilon_{\mu_b} (\phi^{S_0}) + 2M \cdot \mathcal{W}_1 (\mu_b, \mu_t) + \lambda,
$$

where $\lambda = \min_{\phi: X \rightarrow [0,1], \|\phi\|_{Lip} \leq M} (\epsilon_{\mu_b} (\phi) + \epsilon_{\mu_t} (\phi))$.

The first term is the risk on the barycenter distribution $\mu_b$. The second term shows 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 [58], we can measure the generalization risk based on empirical Wasserstein distances similar to Theorem 3 in [59]. Under the assumption of Theorem 2, if the two probability distributions $\mu_b$ and $\mu_t$ satisfy $T_1 (\xi)$ inequality [58], then for any $d' > d$ and $\xi' < \xi$, there exists some constant $N_0$ depending on $d'$ such that for any $\varepsilon > 0$ and $\min(n_b, n_t) \geq N_0 \max (\varepsilon - (d' + 2), 1)$, with probability at least 1 $- \varepsilon$ the following holds for the risk on the target domain:

$$
\epsilon_{\mu_t} (\phi^{S_k}) \leq \epsilon_{\mu_b} (\phi^{S_k}) + 2M \cdot \mathcal{W}_1 (\hat{\mu}_b, \hat{\mu}_t) + \lambda
$$

$$
+ 2M \sqrt{2 \log \left( \frac{1}{\varepsilon} \right) / \xi'} \left( \sqrt{\frac{T}{n_b}} + \sqrt{\frac{T}{n_t}} \right).
$$

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 indicates that reducing the Wasserstein distance between the barycenters and target distributions leads to a tighter upper bound for the risk of the learned model on the target domain. This provides theoretical evidence for our design of test-time adaptation, which aims to reduce the domain gap using optimal transport. Details of the proof can 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 [60] dataset, the PACS [61] dataset, and the Rotated MNIST [62] dataset.

**VLCS** This domain generalization benchmark contains images from four image classification datasets: PASCAL VOC2007 (V), LabelMe (L), Caltech-101 (C), and SUN09 (S) [63], denoted as domains $D_V$, $D_L$, $D_C$, and $D_S$, respectively. There are five fine-grained categories: bird, car, chair, dog, and person.

**PACS** The PACS dataset contains images from four domains: Photos (P), Art painting (A), Cartoon (C), and Sketch (S) [61]. There are in total 7 types of object in this classification task, i.e., dog, elephant, giraffe, guitar, horse, house, and person.

**Rotated MNIST** We constructed the Rotated MNIST dataset with four domains, $r_{90}$, $r_{30}$, $r_{60}$ and $r_{90}$ following the common settings [62]. $r_0$ denotes the domain containing original images from the MNIST dataset, and we rotated images 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 Fig. 3. We randomly sampled among digits [1, 2, 3] for each domain.

### 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., \( D_L, D_C, D_S \rightarrow D_V \), and the average of test accuracies of four cross-domain experiments is taken as the final generalization result.

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 [64] pretrained on the ImageNet is used to extract 512-dimensional features as the inputs. For the VLCS dataset, the pretrained 4096-dimensional DeCAF features [65] are employed as the inputs of our algorithm following previous works [23], [66]. For the PACS dataset, we use the ImageNet pre-trained AlexNet [67] as the backbone network to extract the 9216-dimensional features. The Wasserstein distance of order 2 is used for all experiments, and the discriminability threshold \( \delta \) 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 [71] to solve the Wasserstein distributionally robust optimization problem in an end-to-end trainable neural networks architecture, following [36]. Instead of static optimization with fixed hyperparameters using the convex solver, the solution to the optimization problem can be backpropagated using the optimization layer, allowing for flexible updating of parameters during optimization. In this extended version denoted as WDRDG++, we implement the differentiable convex optimization layers based on cvxpylayers package [72] to make differentiable optimization possible. Two learnable parameters, the uncertainty set radius \( \theta_k \) and the discriminability threshold parameter \( \delta \) become trainable now. The radius computed in (8) is used as the initialization for the parameter \( \theta_k \).

Additionally, we evaluate our pipeline by benchmarking it against some state-of-the-art domain generalization methods, including MLDG [13], ADA [7], GroupDRO [73], VREx [74], and EQRM [75]. For these methods, a simple multi-layer perceptron network is adopted as the trainable classifier on the pretrained features.

### 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 Fig. 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 Fig. 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 illustrate that our method is efficient for few-shot cases.

The second row of Fig. 4 reports the classification accuracy results for the PACS dataset. 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 results for the Rotated MNIST dataset in the third row of Fig. 4 also yield similar conclusions. As the training sample...
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.

When the training sample size increases, almost all methods converge to the same accuracy for different target domains. 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 bigger, e.g., 25 per class for each source domain, the advantage of our framework is not as pronounced when there are fewer training samples available. This is consistent as in the above two datasets.

Fig. 5 reports the average performance of different target domains on the three datasets. Overall, WDRDG++ is more stable under different numbers of training samples, with a narrower shadow band of standard deviation. As the size of training samples gets bigger, all methods have the tendency to perform better. In most cases, WDRDG++ achieves the best average performance under different training sample sizes compared with other methods with smaller standard deviations. In addition, our method shows more advantages 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 capability than others on the VLCS, PACS and Rotated MNIST dataset, respectively. We also did a further exploration of how larger training sample sizes (e.g., 40, 60, 80, 100, 120, 140, 160) impact the generalization capability. More detailed experimental results can be found in 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 first 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
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**

| #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 | 0.304 | 0.350 | 0.471 | 0.237 | 0.391 | 0.593 | 0.640 | 0.577 | 0.553 | 0.591 |
| 10               | 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 |
| 15               | WDRDG (w. TTA)  | 0.540 | 0.402 | 0.516 | 0.334 | 0.448 | 0.559 | 0.374 | 0.480 | 0.259 | 0.418 | 0.567 | 0.695 | 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 |
|                  | WDRDG (w. 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 |

Adding the TTA module results in better performance.

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

We compare the results of using a training sample size of 5,10,15 per class for each source domain.

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 from given source domains, which shows larger unseen distribution shifts. Similar experiments are conducted on the Rotated MNIST dataset with regard to WDRDG++, as shown in Table II. TTA also brings significant performance improvements for WDRDG++. This validates the robustness of our adaptive inference module for even harder, unseen target domains.

**TABLE II**

| # 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** |

TTA module provides performance improvements for WDRDG++.
E. Impact of the Discriminability Threshold $\delta$

We conducted analysis on the Rotated MNIST dataset to evaluate the impact of the threshold parameter $\delta$ on the robustness of our algorithm’s performance, particularly concerning varying sample sizes. Specifically, we recorded the optimal $\delta$ values in WDRDG++ under different training sample sizes, as shown in Fig. 6. The results demonstrate that as the sample size increases, the required $\delta$ stabilizes and decreases, indicating that the necessity for strict $\delta$ constraints diminishes with larger sample sizes. This trend can be attributed to smaller sample sizes producing less accurate barycenters, which are more prone to overlap, thus necessitating a larger optimal $\delta$ to ensure discriminability. With more training samples, the uncertainty set becomes a better estimate of the true distribution, reducing the importance of the distinguishable threshold parameter.

F. Analysis of Imbalanced Classes Among Source Domains

In previous experiments, we assume the training sample size per class in the source domains is 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 the 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 Fig. 7. There are cases when different classes have similar sample numbers, 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 WDRDG with some baseline methods, and the results are shown in Fig. 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 a 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 inferences for unseen target data. Experimental results on Rotated MNIST, PACS, and VLCS datasets demonstrate the effectiveness of our proposed framework in learning 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 relationships among source domains to better leverage the source distributions to model possible target distributions.
REFERENCES

[1] G. Blanchard, G. Lee, and C. Scott, "Generalizing from several related classification tasks to a new unlabeled sample," in Proc. Int. Conf. Adv. Neural Inf. Process. Syst., 2011, vol. 24, pp. 2178–2186.

[2] R. Gong, W. Li, Y. Chen, and L. V. Gool, “DLOW: Domain flow for adaptation and generalization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 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 Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6817–6826.

[4] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, "Learning to generate novel domains for domain generalization," in Proc. Eur. Conf. Comput. Vis., 2020, pp. 561–578.

[5] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Learning generalisable Omni-scale representations for person re-identification," IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 9, pp. 5056–5069, Sep. 2022.

[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 Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 2100–2110.

[7] R. Volpi, H. Namkoong, O. Sener, J. C. Duchi, V. Murino, and S. Suvarese, "Generating source domains via adversarial data augmentation," in Proc. 32nd Int. Conf. Adv. Neural Inf. Process. Syst., 2018, pp. 5339–5349.

[8] R. Shao, X. Lan, J. Li, and P. C. Yuen, “Multi-adversarial discriminative deep domain generalization for face presentation attack detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 10023–10031.

[9] Y. Balaji, S. Sankaranarayanan, and R. Chellappa, “MetaReg: Towards domain generalization using meta-regularization,” in Proc. Int. Conf. Adv. Neural Inf. Process. Syst., 2018, vol. 31, pp. 998–1008.

[10] G. Marzinotto, G. Damanti, F. Bécét, and B. Favre, "Robust semantic parsing with adversarial learning for domain generalization," in Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol., Volume 2 (Industry Papers), 2019, pp. 166–173.

[11] E. Stepnowy and G. Riccardi, "Towards cross-domain PDTB-style discourse parsing," in Proc. 5th Int. Workshop Health Text Mining Inf. Anal., 2014, pp. 30–37.

[12] D. Fried, N. Kitaev, and D. Klein, "Cross-domain generalization of neural constituency parsers," in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, 2019, pp. 323–330.

[13] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, “Learning to generalize: Meta-learning for domain generalization,” Proc. 32nd AAAI Conf. Artificial Intell., vol. 32, no. 1, pp. 3490–3497, 2018.

[14] Y. Tan, E. Zhang, Y. Li, S.-L. Huang, and X.-P. Zhang, “Transferability-guided cross-domain cross-task transfer learning,” IEEE Trans. Neural Netw. Learn. Syst., pp. 1–14 2024.

[15] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, “Scatter component analysis: A unified framework for domain adaptation and domain generalization,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 7, pp. 1414–1430, Jul. 2017.

[16] G. Blanchard, A. A. Deshmukh, U. Dogan, G. Lee, and C. Scott, "Domain generalization by marginal transfer learning," J. Mach. Learn. Res., vol. 22 no. 2, pp. 1–55, 2021.

[17] T. Grubinger, A. Birlutiu, H. Schönner, T. Natschlüger, and T. Heeske, "Domain generalization based on transfer component analysis," in Proc. Int. Work.-Conf. Artif. Neural Netw., 2015, pp. 325–334.

[18] Y. Tan, A. Birlutiu, T. Lin, and Y. Zhou, “Domain generalization via conditional invariant representations,” in Proc. AAAI Conf. Artificial Intell., vol. 32, no. 1, pp. 3579–3587, 2018.

[19] S. Hu, K. Zhang, Z. Chen, and L. Chan, “Domain generalization via multidomain discriminant analysis,” in Proc. 35th Int. Conf. Uncertainty Artif. Intell., 2020, pp. 292–302.

[20] F. Zhou, Z. Jiang, C. Shi, B. Wang, and B. Chaib-draa, “Domain generalization with optimal transport and metric learning,” Neurocomputing, vol. 456, pp. 469–480, 2021.

[21] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, “Moment matching for multi-source domain adaptation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 1406–1415.

[22] S. Motiani, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 5715–5725.
G. Peyré et al., “Computational optimal transport: With applications to data science,” Found. Trends Mach. Learn., vol. 11, no. 5/6, pp. 355–607, 2019.

P. J. Huber, “A robust version of the probability ratio test,” Ann. Math. Statist., vol. 34, no. 6, pp. 1753–1758, 1963.

Z.-H. Zhou and X.-Y. Liu, “Training cost-sensitive neural networks with methods addressing the class imbalance problem,” IEEE Trans. Knowl. Data Eng., vol. 18, no. 1, pp. 63–77, Jan. 2006.

C. Scott, “Calibrated asymmetric surrogate losses,” Electron. J. Statist., vol. 6, pp. 958–992, 2012.

J. Shen, Y. Qu, W. Zhang, and Y. Yu, “Wasserstein distance guided classification,” in Proc. Int. Conf. Scale Space Variational Methods Comput. Vis., 2011, pp. 435–446.

R. Flamary et al., “POT: Python optimal transport,” J. Mach. Learn. Res., vol. 22, no. 78, pp. 1–8, 2021.

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

J. Shen, Y. Qu, W. Zhang, and Y. Yu, “Wasserstein distance guided representation learning for domain adaptation,” in Proc. 32nd AAAI Conf. Artif. Intell., 2018, pp. 4058–4065.

C. Fang, Y. Xu, and D. N. Rockmore, “Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias,” in Proc. IEEE Int. Conf. Comput. Vis., 2013, pp. 1657–1664.

D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, “Deeper, broader and attier domain generalization,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 3542–3550.

M. Ghifary, W. Bastiaan Kleijn, M. Zhang, and D. Balduzzi, “Domain generalization for object recognition with multi-task autoencoders,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 2551–2559.

A. Torralba and A. A. Efros, “Unbiased look at dataset bias,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2011, pp. 1521–1528.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

J. Donahue et al., “DECAF: A deep convolutional activation feature for generic visual recognition,” in Proc. Int. Conf. Mach. Learn., 2014, pp. 647–655.

Q. Dou, D. Coelho de Castro, K. Kamnitas, and B. Glocker, “Domain generalization via model-agnostic learning of semantic features,” in Proc. Int. Conf. Adv. Neural Inf. Process. Syst., 2019, vol. 32, pp. 6450–6461.

A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Int. Conf. Adv. Neural Inf. Process. Syst., 2012, vol. 25, pp. 1097–1105.

S. Zhu, L. Xie, M. Zhang, R. Gao, and Y. Xie, “Distributionally robust weighted K-nearest neighbors,” in Proc. Int. Conf. Adv. Neural Inf. Process. Syst., 2022, vol. 35, pp. 29088–29100.

S. Diamond and S. Boyd, “CVXPY: A python-embedded modeling language for convex optimization,” J. Mach. Learn. Res., vol. 17, no. 1, pp. 2909–2013, 2016.

M. ApS, “MOSEK optimization toolbox for MATLAB,” User’s Guide and Reference Manual, Version 4.1, 2019.

A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, “Differentiable convex optimization layers,” in Proc. Int. Conf. Adv. Neural Inf. Process. Syst., 2019, vol. 32, pp. 9562–9574.

A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and Z. Kolter, “Differentiable convex optimization layers,” in Proc. Int. Conf. Adv. Neural Inf. Process. Syst., 2019, vol. 32, pp. 9562–9574.

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,” in Proc. Int. Conf. Learn. Representations, 2019, pp. 1–19.

D. Krueger et al., “Out-of-distribution generalization via risk extrapolation (rex),” in Proc. Int. Conf. Mach. Learn., 2021, pp. 5815–5826.

C. Eastwood et al., “Probable domain generalization via quantile risk minimization,” in Proc. Int. Conf. Adv. Neural Inf. Process. Syst., 2022, vol. 35, pp. 17340–17358.