Constrained Maximum Cross-Domain Likelihood for Domain Generalization

Jianxin Lin®, Yongqiang Tang®, Junping Wang, and Wensheng Zhang®

Abstract—As a recent noticeable topic, domain generalization aims to learn a generalizable model on multiple source domains, which is expected to perform well on unseen test domains. Great efforts have been made to learn domain-invariant features by aligning distributions across domains. However, existing works are often designed based on some relaxed conditions which are generally hard to satisfy and fail to realize the desired joint distribution alignment. In this article, we propose a novel domain generalization method, which originates from an intuitive idea that a domain-invariant classifier can be learned by minimizing the Kullback–Leibler (KL)-divergence between posterior distributions from different domains. To enhance the generalizability of the learned classifier, we formalize the optimization objective as an expectation computed on the ground-truth marginal distribution. Nevertheless, it also presents two obvious deficiencies, one of which is the side-effect of entropy increase in KL-divergence and the other is the unavailability of ground-truth marginal distributions. For the former, we introduce a term named maximum in-domain likelihood to maintain the discrimination of the learned domain-invariant representation space. For the latter, we approximate the ground-truth marginal distribution as an expectation computed on the ground-truth marginal distributions from different domains. An alternating optimization strategy is carefully designed to approximately solve this optimization problem. Extensive experiments on four standard benchmark datasets, i.e., Digits-DG, PACS, Office-Home, and miniDomainNet, highlight the superior performance of our method.

Index Terms—Distribution shift, domain adaptation, domain generalization, domain-invariant representation, joint distribution alignment.

I. INTRODUCTION

Deep learning methods have achieved remarkable success in computer vision tasks under the assumption that train data and test data follow the same distribution. Unfortunately, this important assumption does not hold in real-world applications [1]. The distribution shift between train data and test data, which are widespread in various vision tasks, is unpredictable and not even static, thus hindering the application of deep learning in reliability-sensitive scenarios. For example, in the field of medical image processing, image data from different hospitals follow different distributions due to discrepancies in imaging protocol, device vendors, and patient populations [2]. Hence, the models trained on data from one hospital often suffer from performance degradation when tested in another hospital owing to the distribution shift.

To tackle the distribution shift problem, considerable efforts have been made in domain adaptation and domain generalization. Domain adaptation assumes that the target domain is accessible and attempts to align the distributions between the source domain and the target domain. However, in the setting of domain adaptation, the model inevitably needs to be retrained when the distribution of the target domain changes, which can be time-consuming and cumbersome [3]. More importantly, in many cases, there is no way to access the target domain in advance. Fortunately, domain generalization has been proposed to improve the generalization ability of models in out-of-distribution scenarios given multiple source domains, where the target domain is inaccessible [4].

As an active research area, many domain generalization methods have been proposed. Let $X$ denote an input variable, i.e., an image, $Z = F(X)$ denote the feature extracted from $X$ by a feature extractor $F(\cdot)$ and $Y$ denote an output variable, i.e., a label. An effective and general solution to domain generalization is learning a domain-invariant representation space where the joint distribution $P(Z, Y)$ across all source domains keeps consistent [4], [5], [6], [7]. Along this line, some works [4], [8] try to align the marginal distribution $P(Z)$ among domains assuming that the posterior distribution $P(Y | Z)$ is stable across domains. Problematically, there is no guarantee that $P(Y | Z)$ will be invariant when aligning $P(Z)$ [9], [10]. Some methods [6] attempt to align the class-conditional distribution $P(Z | Y)$ according to $P(Z, Y) = P(Z | Y) P(Y)$. Only if the categorical distribution $P(Y)$ keeps invariant across domains, aligning the class-conditional distributions could achieve domain-invariant joint distribution [7]. But this requirement is difficult to meet in practical applications.

More recently, the domain-invariant classifier, or the invariant predictor, has attracted much interest [11], [12], [13], [14], [15]. In essence, these works are performing posterior distribution alignment. Invariant risk minimization (IRM) [12]
seeks an invariant causal predictor, which is a simultaneously optimal classifier for all environments (domains). IRM is formalized as a hard-to-solve bi-leveled optimization problem. The invariant causal predictor realizes the conditional expectation \( \mathbb{E}[Y|Z] \) alignment across domains. It is a coarse posterior distribution alignment due to the insufficiency of the conditioned expectation. Robey et al. \[9\] propose a novel definition of invariance called G-invariance, which requires that the classifier should hold invariant prediction after \( X \) is transformed to any another domain by a domain transformation model \( G \). Li et al. \[15\] propose a new formulation called invariant information bottleneck (IIB), which achieves the domain-invariant classifier by minimizing the mutual information between \( Y \) and domain label given \( Z \). Despite the brilliant achievements, the above methods do not take marginal distribution alignment into consideration and thus fail to realize the desired joint distribution alignment. To ensure that the joint distribution is invariant across domains, both \( P(Z) \) and \( P(Y|Z) \) must be considered \[16\].

In this article, we propose a novel domain generalization method that can jointly align the posterior distribution and the marginal distribution. Specifically, we formalize a general optimization objective, in which for any given sample, except for the routine empirical risk minimization, the Kullback–Leibler (KL) divergence \[17\] between posterior distributions from different domains is also minimized so that the domain-invariant classifier can be learned. To enhance the generalization ability of the learned classifier, the optimization objective is designed as an expectation computed on the ground-truth marginal distribution. Unfortunately, the above optimization problem still has two deficiencies that must be overcome. The first issue lies in the side-effect of KL-divergence which tends to enlarge the entropy of posterior distributions. To tackle this issue, we add a new term named maximum in-domain likelihood into the overall optimization objective, such that the discrimination of the learned domain-invariant feature space is reinforced. The second issue is that the ground-truth marginal distribution is not available directly. In light of this, we propose to approximate the real-world marginal distribution with source domains under a reasonable convex hull assumption. Eventually, a concise and intuitive optimization problem namely constrained maximum cross-domain likelihood (CMCL) is deduced, by solving which we can learn a domain-invariant representation space where the joint distributions across domains are naturally aligned.

The major contributions of our article can be summarized as follows.

1) We propose a new formulation for domain generalization, which minimizes the expectation of KL-divergence between posterior distributions from different domains. We innovatively compute the expectation on the ground-truth marginal distribution, such that the generalizability of the learned model can be enhanced.

2) A CMCL optimization problem is deduced by adding an objective term of maximum in-domain likelihood and a constraint of marginal distribution alignment. The former eliminates the side-effect brought by minimizing KL-divergence, and the latter makes it possible to approximate the ground-truth marginal distribution with source domains.

3) An effective alternating optimization strategy with multiple optimization stages is elaborately developed to solve the maximum cross-domain likelihood problem. Comprehensive experiments are conducted on four widely used datasets and the results demonstrate that our CMCL achieves superior performance on unseen domains.

II. RELATED WORKS

In this section, we review the related works dealing with the domain (distribution) shift problem in deep learning, which can be divided into two categories, including domain adaptation and domain generalization.

A. Domain Adaptation

Domain adaptation aims to tackle the domain shift between a source domain and a particular target domain \[18\], \[19\]. The goal of domain adaptation is to train models making full use of a large amount of labeled data from a source domain to perform well on the unlabeled target domain. Most existing domain adaptation methods focus on aligning distributions between the source domain and the target domain \[20\]. They can be mainly divided into two categories: discrepancy measuring-based methods and domain adversarial-based methods.

Discrepancy measuring-based methods employ different metrics to measure the distribution disparities and then minimize them, e.g., maximum mean discrepancy (MMD) \[21\], central moment discrepancy (CMD) \[22\], and Wasserstein distance \[23\]. Deep domain confusion \[24\] employs MMD to align marginal distributions in the deep representation space. Deep CORAL \[25\] and CMD \[22\] align marginal distributions with moment matching. Joint MMD \[26\] is proposed to align the joint distributions considering the distribution shifts may stem from joint distributions. Domain adversarial based methods use domain discriminators to minimize the distance between distributions \[27\]. Feature extractors are optimized to confuse the discriminators so that the divergence of distributions is reduced. Domain-adversarial neural network \[27\] is proposed to align marginal distributions by adversarial learning. Multidomain adversarial domain adaptation \[28\] considers the alignment of multimode distributions, i.e., class-conditional distributions, instead of marginal distributions. Zuo et al. \[29\] concatenate features and corresponding labels together, and feed them into a domain classifier, then the joint distributions are aligned in an adversarial training manner.

The difference between domain adaptation and domain generalization lies in the accessibility to the target domain. The former focuses on the alignment between the given source domain and target domain, but the latter focuses more on the generalizability on unseen test domains.

B. Domain Generalization

Domain generalization aims to train models on several source domains and test them on unseen domain \[30\], \[31\], \[32\]. Existing works of domain generalization carry out the
research mainly from three aspects, including learning strategy, data augmentation, and domain invariant representation.

Learning strategy based methods mainly design special learning strategies to enhance generalizability [34]. Some works employ meta learning to address domain generalization, which randomly split the source domains into meta-train and meta-test to simulate the domain shift. Balaji et al. [35] train a regularizer through meta learning to capture the notion of domain generalization, which is parameterized by a neural network. Dou et al. [36] propose a model-agnostic learning paradigm based meta learning to enhance the generalizability of learned features. Global interclass relationships, local class-specific cohesion, and separation of sample features are also considered to regularize the semantic structure of the feature space. Gao et al. [33] introduce a novel meta-learning approach to search loss function. In addition to meta learning, distributionally robust optimization (DRO) [37] is also used for domain generalization, which trains models by minimizing the worst case loss over predefined groups. Sagawa et al. [38] find that coupling DRO with stronger regularization achieves highest worst case accuracy in the over-parameterized regime. Sener and Koltun [39] argue that the erroneous penalty term in joint optimization can cause the excess empirical risk, which leads to the failure of existing domain generalization approaches. So they choose to minimize the penalty under the constraint of optimality of the empirical risk instead of jointly minimization. Ding et al. [40] design multiple classifiers for each domain to learn domain-specific features, and the domain-invariant model can be learned by removing those domain-specific features. Yang et al. [41] also propose to enhance generalization ability with domain-specific and domain aggregation networks.

The core idea of data augmentation-based methods is to increase the diversity of training data. MixStyle [42] is motivated that the visual domain is closely related to image style, which is encoded by feature statistics. The domain diversity can be increased by randomly combining feature statistics between two training instances. Deep domain-adversarial image generation (DDAIG) [43] is proposed to fool the domain classifier by augmenting images. A domain transformation network is designed to automatically change image style. Seo et al. [44] propose a domain-specific optimized normalization (DSON) to remove the domain-specific style. Wang et al. [45] design a feature-based style randomization module, which randomizes image style by introducing random noise into feature statistics. These style augmentation-based methods actually exploit the prior knowledge about domain shift, that is, the difference across source domains lies in image style [46]. Though they work well in existing benchmarks, style augmentation-based methods would probably fail when the domain shift is caused by other potential factors. Methods that do not rely on prior knowledge deserve further study.

Domain-invariant representation-based methods often achieve domain invariance by aligning distributions of different domains as they did in domain adaptation [47]. Li et al. [5] impose MMD to an adversarial autoencoder to align the marginal distributions $\mathcal{P}(Z)$ among domains, and the aligned distribution is matched with a predefined prior distribution by adversarial training. Motiian et al. [48] try to align the class-conditional distributions $\mathcal{P}(Z|Y)$ for finer alignment. However, class-conditional distributions alignment-based methods hardly deal with the domain shift caused by the label shift, which requires that categorical distribution $\mathcal{P}(Y)$ remains unchanged among domains. Another important branch attempts to achieve domain-invariant representation via domain-invariant classifier learning. IRM [12] tries to learn a domain-invariant classifier by constraining that the classifier is simultaneously optimal for all domains. But this optimization problem is hard to solve. Our method CMCL learns the domain-invariant classifier via posterior distribution alignment, an effective alternating optimization strategy is proposed to solve our optimization problem leading to excellent performance. Zhao et al. [50] propose an entropy regularization term to align posterior distributions. According to our analysis, the proposed entropy term is a side-effect of minimizing KL-divergence, severely damaging classification performance. In our method, a term of maximum in-domain likelihood is proposed to eliminate this side-effect.

### III. Proposed Method

#### A. Overview

In this article, we focus on domain generalization for image classification. Suppose the sample and label spaces are represented by $\mathcal{X}$ and $\mathcal{Y}$, respectively, then a domain can be represented by a joint distribution defined on $\mathcal{X} \times \mathcal{Y}$. There are $N$ datasets $\mathcal{D} = \{\mathcal{S}^i = \{(x^i_j, y^i_j)\}_{j=1}^{M_i}\} = \mathcal{N}_{i=1}^{M_i}$ sampled from domains with different distributions $\{\mathcal{P}(X, Y)\}_{i=1}^{M_i}$, where $M_i$ denotes the number of samples of dataset $\mathcal{S}^i$, $X \in \mathcal{X}$, and $Y \in \mathcal{Y}$. Let $P(X, Y)$ denote the ground-truth joint distribution in the real world. As shown in Fig. 1, we suppose that $P(X, Y)$ yields distributions of training domains $\{\mathcal{P}(X, Y)\}_{i=1}^{M_i}$ and distribution of unseen domain $\mathcal{P}^u(X, Y)$, with different domain shift due to different selection bias.

Given several training domains following different distributions, domain generalization aims to learn a model which is expected to overcome the domain shift and maintain its performance on unseen domains. To overcome the distribution shift across domains, we try to learn a domain-invariant representation space in which the joint distributions of different domains are aligned.

**Definition 1 (Domain-Invariant Representation):** Let $\mathcal{E}$ be a set of all possible domains. $F(\cdot) : \mathcal{X} \to \mathbb{R}^d$ is a feature mapping function that transforms raw input to the
domain-invariant representation space. A representation space is domain-invariant if

\[ \forall i \neq j \in \mathcal{E} \quad P^i(Z, Y) = P^j(Z, Y) \]  

(1)

where \( Z = F(X) \).

To obtain the domain-invariant representation space, we first focus on aligning the posterior distribution from the perspective of domain-invariant classifier learning.

**Definition 2 (Domain-Invariant Classifier):** Given a particular representation space, a domain-invariant classifier is simultaneously Bayes optimal classifier on any domain, which can be obtained when posterior distributions of different domains are aligned

\[ \forall i \neq j \in \mathcal{E} \quad P^i(Y|Z) = P^j(Y|Z). \]  

(2)

We propose an optimization problem to learn the domain-invariant classifier, which minimizes the KL-divergence between posterior distributions of different domains and maximizes the discrimination of the in-domain feature space (see Section III-B1). The optimization objective is formalized as an expectation of the KL-divergence computed on ground-truth marginal distribution \( P(Z) \) to enhance the generalizability of the domain-invariant classifier on unseen domains. Due to the unavailability of ground-truth marginal distribution \( P(Z) \), a marginal distribution alignment constraint is proposed to approximate \( P(Z) \) by \( \{P^i(Z)\}_{i=1}^N \) under a convex hull assumption. Finally, a CMCL optimization problem is formalized (see Section III-B2). Joint distributions are naturally aligned after solving this constrained optimization problem. An alternating optimization strategy is proposed to solve this constrained optimization problem (see Section III-C).

**B. Constrained Maximum Cross-Domain Likelihood**

The typical classifier of traditional deep learning assumes that samples follow independent identical distribution and tries to minimize the following objective:

\[ \min_F \sum_{i=1}^{N} \mathbb{E}_{P^i(Z,Y)}[\log P^i(Y|Z)] \]  

(3)

where \( Z = F(X) \) denotes the feature of raw input \( X \) and \( P^i(Y|Z) \) denotes the global classifier trained with the data in all source domains. \( \mathbb{E}_{P(Z,Y)}[\cdot] \) denotes an expectation computed over the distribution \( P(Z, Y) \), i.e., \( \int P(Z, Y)[\cdot] \mathrm{d}Z \mathrm{d}Y \).

Equation (3) is a regular empirical risk and can be regarded as a term of maximum likelihood, which ignores the fact that the data collected from different environments (domains) generally present distribution shifts.

To learn a domain-invariant classifier with better generalization ability on unseen domains, in this study, we propose to minimize the KL-divergence between posterior distributions of different domains as follows:

\[ \min_F \mathbb{E}_{P(Z)} \left[ \sum_{i \neq j} \text{KL}(P^i(Y|Z) \parallel P^j(Y|Z)) \right] 
- \sum_{i=1}^{N} \mathbb{E}_{P^i(Z,Y)}[\log P^i(Y|Z)] \]  

(4)

where \( P(Z) \) denotes the ground-truth marginal distribution in the real world. In this article, \( \text{KL}(\cdot|\cdot) \) means the original nonsymmetric KL divergence, but note that symmetric KL divergence has been equivalently used in (4) because \( P^i(Y|Z) \) is computed twice due to \( \sum_i \neq j \). We provide the symmetric KL-divergence form of (4) in supplementary materials. Besides, we also analyze what will happen if symmetric KL-divergence is replaced with Jensen–Shannon divergence in supplementary materials. The first term of the above formula means that a representation space is optimized hoping that all domain-specific posterior distributions can be the same for any given sample sampled from the ground-truth marginal distribution.

Note that the expectation calculated on the ground-truth marginal distribution makes the optimization objective more general instead of being limited to source domains. If the expectation is calculated on source domains, the alignment of posterior distribution can only be achieved on limited several source domains. To generalize to unseen domains, the ideal optimization object should be an expectation calculated on unseen test distributions. An unseen test domain, which is yielded from the ground-truth distribution with a selection bias leading to domain shift, is naturally near to the ground-truth distribution. The distribution shift between the unseen test domain and the ground-truth marginal distribution may be small than that between the unseen domain and source domains. So the ground-truth marginal distribution is a reasonable substitute for arbitrary unseen test distributions, and hope that the learned classifier can generalize well to unseen test domains.

1) **Maximum In-Domain Likelihood:** However, minimizing the KL-divergence directly would produce a side-effect that can seriously damage the classification performance. To illustrate this more clearly, we divide KL-divergence into two terms as follows:

\[ \text{KL}(P^i(Y|Z) \parallel P^j(Y|Z)) 
= \mathbb{E}_{P^i(Y|Z)}[\log P^i(Y|Z)] - \mathbb{E}_{P^j(Y|Z)}[\log P^j(Y|Z)]. \]  

(5)

When minimizing the KL-divergence, the first term is also minimized, which is essentially maximum entropy. Greater entropy means greater prediction uncertainty, which is contrary to the goal of the classification task. To solve this problem, another optimization objective is proposed

\[ \min_F \sum_{i \neq j} \left( \mathbb{E}_{P(Z)} \left[ \text{KL}(P^i(Y|Z) \parallel P^j(Y|Z)) \right] 
- \mathbb{E}_{P^j(Z,Y)}[\log P^j(Y|Z)] \right) 
- \sum_{i=1}^{N} \mathbb{E}_{P^i(Z,Y)}[\log P^i(Y|Z)]. \]  

(6)

A new term is proposed, which maximizes the posterior probability of the labeled data \( (Z, Y) \) sampled from each domain. This term aims to maintain the discrimination of the learned representation space. Actually, it is essentially a maximum in-domain likelihood objective. This term is
obviously different from the third term, which is a maximum global likelihood objective. The former measures the in-domain likelihood on domain-specific distributions, while the latter measures the global likelihood on the global distribution by aggregating images from all source domains. Next, we introduce the following necessary definition for further analyzing the optimization problem in (6).

Definition 3 (Marginal Distribution Error): In the representation space, let \( P(Z) \) be the ground-truth marginal distribution. For the marginal distribution \( P^i(Z) \) in each source domain, \( 0 \leq i \leq N \), there exists a distribution error \( \Delta^i(Z) \) such that \( \Delta^i(Z) = P(Z) - P^i(Z) \).

The formulation in (6) can be further decomposed as

\[
\sum_{i \neq j} \left( -E_{P^j(Z,Y)}[\log P^i(Y|Z)] + \int \Delta^i(Z)KL(P^i(Y|Z) \parallel P^j(Y|Z))dZ \right) - \sum_{i=1}^{N} E_{P^i(Z,Y)}[\log P^i(Y|Z)].
\]

(7)

We provide the detailed derivation of (7) in supplementary materials. As shown above, the proposed new term of maximum in-domain likelihood eliminates the side-effect of minimizing KL-divergence. Original optimization objective in (6) is transformed into a new form in (7).

2) Marginal Distribution Alignment Constraint: Because of the unavailability of the ground-truth marginal distribution, there is no way to optimize the integral term \( \int \Delta^i(Z)KL(P^i(Y|Z) \parallel P^j(Y|Z))dZ \) in (7) directly. Hence we introduce a new reasonable assumption which is critical for distribution alignment-based domain generalization.

Assumption 1 (Inner Point of the Convex Hull): Let a set of marginal distributions of source domains in representation space be denoted as \( \mathcal{M} = \{P^i(Z)\}_{i=1}^{N} \). The convex hull of the set \( \mathcal{M} \) is a set of all convex combinations of distributions in \( \mathcal{M} \)

\[
\Lambda = \left\{ \sum_{i} \pi_i P^i(Z) \middle| P^i(Z) \in \mathcal{M}, \pi_i \geq 0, \sum_{i=1}^{N} \pi_i = 1 \right\}.
\]

The ground-truth marginal distribution is always an inner point of the convex hull

\[
P(Z) \in \Lambda.
\]

As shown in Fig. 2, it is reasonable that the ground-truth marginal distribution should lie inside of the convex hull of source domains for domain generalization. Under this assumption, the ground-truth marginal distribution can be depicted by source domains. Otherwise, the generalization on any possible unseen domain given several source domains cannot be guaranteed, and domain generalization would be an unattainable goal. Similar assumptions are also covered in [8], [10], [38], and [51]. Albuquerque et al. [8] and Sagawa et al. [38] assume that the distributions of the unseen domain stay inside the convex hull of source domains. Sagawa et al. [38] tries to optimize the worst case expected loss over an uncertainty set of distributions, which encodes the possible test distributions. The uncertainty set is defined as a set of convex combinations of source domains. Even though Krueger et al. [10] and Rosenfeld et al. [51] try to handle scenarios that unseen domains are extrapolations of source domains, they still admit that many existing researches are based on the basic assumption that unseen domains can be seen as interpolations of source domain and it is an important scenario for consideration.

Under the above assumption, we try to align marginal distributions across different source domains so that the convex hull shrinks to a certain point. In this case, the ground-truth marginal distribution would be aligned to domain-specific marginal distributions, and the integral term in (7) would approach 0. In other words, we hope that \( \Delta^i(Z) \) is negligibly small after aligning \( \{P^i(Z)\}_{i=1}^{N} \). We can get the following proposition by adding a constraint to (7).

Proposition 1 (CMCL): Under Assumption 1, if the marginal distributions of source domains are aligned, the original optimization objective in (6) can be achieved by solving the following constrained optimization problem:

\[
\max_F \sum_{i \neq j} E_{P^j(Z,Y)}[\log P^i(Y|Z)] + \sum_{i=1}^{N} E_{P^i(Z,Y)}[\log P^i(Y|Z)]
\]

s.t. \( \forall 1 \leq i \neq j \leq N, P^i(Z) = P^j(Z) \).

(10)

Proof: Under Assumption 1, if \( \forall i \neq j, P^i(Z) = P^j(Z) \), we can get that \( \forall i, P(Z) = P^i(Z) \), and then \( \Delta^i(Z) \equiv 0 \). Hence \( \int \Delta^i(Z)KL(P^i(Y|Z) \parallel P^j(Y|Z))dZ = 0 \), and then we can get that (10) is equivalent to (6).

Optimizing both KL-divergence and maximum in-domain likelihood generates a constrained optimization problem, containing a term of maximum cross-domain likelihood under the condition of marginal distribution alignment, which means that the data sampled from one domain should have high posterior probability even though measured in the posterior distribution of another domain. This optimization objective of maximum cross-domain likelihood realizes the alignment of posterior distributions while improving the discrimination of representation space, and extends the traditional maximum likelihood to the domain shift setting. Marginal distributions and posterior distributions in the representation space will be aligned by solving this constrained optimization problem, and thus
joint distributions will be aligned naturally. Furthermore, the marginal distribution alignment is non-trivially coupled with posterior distribution alignment, which is indeed designed for the purpose of enhancing the generalization ability of the domain-invariant classifier.

3) Practical Operation: The non-convex constrained optimization problem described in (10) is hard to be solved. For simplicity, we transform it into an unconstrained optimization problem by adding a penalization term

\[
\max_F \sum_{i \neq j} \left( \mathbb{E}_{P(y|z)} \left[ \log P^j(y|z) \right] - \lambda \text{Dis}(P^j(z), P^i(z)) \right) + \sum_{i=1}^N \mathbb{E}_{P(y|z)} \left[ \log P^i(y|z) \right]
\]

(11)

where \( \lambda \) is a parameter controlling the intensity of the penalization term, and \( \text{Dis}(\cdot) \) denotes the distance between two distributions. We adopt the moment matching loss [22] to implement the penalization term \( \text{Dis}(\cdot) \). The first-order raw moment and second-order central moment of marginal distributions can be calculated as follows:

\[
\bar{z}^i = \frac{1}{|S_i|} \sum_{x \in S_i} F(x)
\]

(12)

\[
C^i = \frac{1}{|S_i| - 1} \sum_{x \in S_i} \left( (F(x) - \bar{z}^i) (F(x) - \bar{z}^i)^T \right).
\]

(13)

Moment matching loss functions are designed as

\[
\mathcal{L}_{\text{mean}} = -\frac{2}{N(N-1)d} \sum_{i \neq j} \| z^i - z^j \|^2_F
\]

(14)

\[
\mathcal{L}_{\text{cov}} = -\frac{2}{N(N-1)d^2} \sum_{i \neq j} \| C^i - C^j \|^2_F
\]

(15)

where \( d \) denotes the dimension of features used to rescale the loss value, and \( \| \cdot \|^2_F \) denotes the squared matrix Frobenius norm.

Then the final moment matching loss function can be defined as

\[
\mathcal{L}_{\text{mm}} = \lambda_1 \mathcal{L}_{\text{mean}} + \lambda_2 \mathcal{L}_{\text{cov}}
\]

(16)

where \( \lambda_1 \) and \( \lambda_2 \) are trade-off parameters.

C. Alternating Optimization Strategy

In this section, we propose an alternating optimization strategy to approximately solve (11). In this elaborately designed optimization process, the posterior distribution estimation and the posterior distribution alignment are decoupled and carried out at different stages, and the difference among domains is explicitly explored and then minimized effectively.

1) Parameterization of Posterior Distribution: It is primary to calculate the posterior probability given a sample \( P(Y|Z = z) \) when optimizing the objective of maximum cross-domain likelihood. We adopt the softmax classifier to parameterize the posterior distribution

\[
P(Y = y|Z = z) = \frac{\exp(w_y z)}{\sum_{y \in Y} \exp(w_y z)}
\]

(17)

where \( w_y \) and \( w_y \) denote the corresponding row of the parameter matrix \( W \in \mathbb{R}^{K \times d} \) of the softmax classifier and \( K \) is the number of classes. In the process of optimizing the maximum cross-domain likelihood objective described in (10) and (11), posterior distributions of all domains need to be estimated separately. Hence \( N \) domain-specific classifiers \( \{P^i(y|z; W^i)\}_{i=1}^N \) are introduced to parameterize the posterior distribution of each domain. In addition to domain-specific classifiers, we need to train a global classifier \( W^g \) with all samples based on the learned representation, which is required by the second term in (11).

2) Alternating Optimization: In order to maximize the cross-domain likelihood, we should estimate the posterior distributions of all domains \( \{P^i(y|z; W^i)\}_{i=1}^N \) before updating the feature extractor \( F(\cdot) \). After \( F(\cdot) \) is updated, the representation space has been changed and \( \{P^i(y|z; W^i)\}_{i=1}^N \) need to be re-estimated. Therefore, an alternating optimization strategy for domain-invariant classifier learning is designed to approximately solve the constrained optimization problem:

a) Stage A: The feature extractor and all classifiers are jointly trained through vanilla empirical risk minimization to maintain the classification ability of the classifiers and further enhance the discrimination ability of the learned representation extracted by \( F(\cdot) \) during the alternating training process. The loss function can be calculated as

\[
\mathcal{L}_{\text{ce}} = -\frac{1}{|D|} \sum_{(x,y) \in D} \left( \sum_{i=1}^N \log P^i(y|F(x); W^i) \right)
\]

(18)

which is essentially a cross-entropy loss function. Additionally, at this stage, the penalization term in (11) is implemented by aligning marginal distributions by moment matching. The loss function at this training stage can be defined as

\[
\mathcal{L}_{\text{cemm}} = \mathcal{L}_{\text{ce}} + \mathcal{L}_{\text{mm}}.
\]

(19)

b) Stage B: The feature extractor is frozen, providing a deterministic representation space for estimating the posterior distributions, which is denoted by \( \bar{F}(\cdot) \). Given the fixed representations, the domain-specific classifiers are trained with data sampled from respective domains. The loss function at this training stage can be defined as

\[
\mathcal{L}_{\text{dsc}} = -\sum_{i=1}^N \frac{1}{|S_i|} \sum_{(x,y) \in S_i} \log P^i(y|\bar{F}(x); W^i).
\]

(20)

As mentioned earlier, the domain-specific classifiers tend to be consistent as the alternating training goes on. Then the optimal global classifier can be obtained at the convergence point of the domain-specific classifiers. Hence, at this stage, we set the parameters of the global classifier as the mean of all domain-specific classifiers to accelerate the convergence of the training process and improve the stability of the training process

\[
W^g = \frac{1}{N} \sum_{i=1}^N W^i.
\]

(21)
loss function at this training stage can be defined as likelihood is maximized by updating the feature extractor. The into the classifier of another domain. Then the cross-domain

Estimate the posterior distribution of each domain in a fixed representation space, that is, the feature extractor is frozen. Stage C: Update the feature extractor to align the posterior distributions parameterized by frozen domain-specific classifiers via maximum cross-domain likelihood.

c) Stage C: The domain-specific classifiers are frozen, providing measurements of the posterior distributions for updating the feature extractor. Given the fixed domain-specific classifiers \( \{\hat{W}^i\}_{i=1}^N \), the data sampled from one domain are fed into the classifier of another domain. Then the cross-domain likelihood is maximized by updating the feature extractor. The loss function at this training stage can be defined as

\[
\mathcal{L}_{\text{cdd}} = -\sum_{i=1}^N \frac{1}{|S^i|} \sum_{(x,y) \in S^i} \left( \sum_{j \neq i} \log P_j^i(y|F(x); \hat{W}^j) \right. \\
\left. + \log P^\delta(y|F(x); W^\delta) \right) .
\]  

(22)

At this stage, the initial point of parameters of the global classifier \( W^\delta \) is the average of all domain-specific classifiers as mentioned at stage B. The global classifier is trained together with the feature extractor alleviating the problem of over-adjustment when maximizing the cross-domain likelihood.

As described above, we carry out three stages of the training process alternately and this process keeps cycling. To improve the stability of the training process and facilitate generalization, in addition to the online model which is updated along the negative gradient direction, we introduce an extra target model which is updated along the differential direction of the parameters between the online model and target model. It is essentially the exponential moving average (EMA) of parameters of the online model

\[
\Theta^\text{target}_t = \Theta^\text{target}_t + \alpha (\Theta^\text{online}_t - \Theta^\text{target}_t)
\]  

(23)

where \( \Theta = \{\hat{W}^i, F\} \), \( \Theta^\text{target}_t \) and \( \Theta^\text{online}_t \) denote the parameters of the target model and online model at step \( t \), respectively, and \( \alpha \) denotes the step size of EMA. In this article, \( \alpha \) is set to 0.001 for all experiments.

As Fig. 3 shows, we optimize \( \mathcal{L}_{\text{cmm}}, \mathcal{L}_{\text{disc}} \) and \( \mathcal{L}_{\text{cdd}} \) alternately to align marginal distributions and posterior distributions so that the constrained optimization problem described in (10) can be solved approximately. To illustrate the training process clearly, the pseudo-code of our algorithm is provided in Algorithm 1.

3) Complexity Analysis: Let \( \mathcal{O}(F_t) \) and \( \mathcal{O}(F_p) \) denote the complexity of forward and backward propagation of the feature extractor, \( \mathcal{O}(C_t) \) and \( \mathcal{O}(C_p) \) denote the complexity of forward and backward propagation of each classifier, \( \mathcal{O}(M) \)
denotes the complexity of computing mean classifier, and $O(E)$ denotes the complexity of EMA. So the complexity of alternating optimization strategy is

$$
O\left(n(n_{A}(F_f + F_b + (N + 1)(C_f + C_b)) + n_{B}(F_f + N(C_f + C_b)) + M + n_{C}(F_f + F_b + NC_f) + E)\right).
$$

(24)

Though there are multiple classifiers during training, only one feature extractor and one global classifier are retained in test time. And the complexity in test time is $O(F_f + C_f)$, which is equal to that of vanilla empirical risk minimization methods.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we conduct extensive experiments on four popular domain generalization datasets to validate the effectiveness of the proposed CMCL. Ablation studies and further analysis are carried out to analyze the characteristics of the proposed CMCL approach.

A. Datasets and Settings

The proposed method is evaluated on four commonly used datasets. Following are details of these datasets.

1) Digits-DG [43]: is a digit recognition benchmark, which is consisted of four classical datasets MNIST [52], MNIST-M [27], SVHN [53], and SYN [27]. Each dataset is treated as a domain, and the domain shift across the four datasets is mainly reflected in font style, background, and image quality. Each domain contains ten categories, and each class contains 600 images. The original train-validation split in [43] is adopted for a fair comparison.

2) PACS [54]: is an object recognition benchmark which is consisted of four domains namely Photo, Art, Cartoon and Sketch. The main domain shift is reflected in a large discrepancy in image style. There are 9991 images and seven classes in PACS. The original train-validation split provided by Li et al. [54] is adopted.

3) Office-Home [55]: is also an object recognition benchmark which consists of four domains, i.e., Art, Clipart, Product, and Real-world. The main domain shift is reflected in image style and viewpoint, with less diversity than PACS. There are around 15,500 images in total, and each domain contains 65 classes. We randomly split each domain into 90% for training and 10% for validation following [56].

4) miniDomainNet [57]: is a larger object recognition benchmark, which takes a subset of DomainNet [58] containing four domains namely Clipart, Painting, Real, and Sketch. There are 140,006 images and 126 classes. The original train-validation split provided by Zhou et al. [57] is adopted.

Images are resized to $32 \times 32$ for Digits-DG, and $224 \times 224$ for PACS, Office-Home and miniDomainNet. All experiments are conducted following the commonly used leave-one-domain-out protocol [54]. Specifically, one domain is specified as an unseen test domain, and the remaining domains are treated as source domains to train the model. To fairly compare with published methods, our models are trained using data only from the training split, and the models with the best result on the validation split are selected for testing. All results are reported based on the average top-1 classification accuracy over three repetitive runs.

B. Implementation Details

For all experiments, the models are constructed by a feature extractor, three domain-specific classifiers and a global classifier. The classifiers are implemented by softmax layers. The domain-specific classifiers are trained by an AdamW optimizer with a learning rate of $1e^{-3}$ and weight decay of $5e^{-4}$. The number of iterations of stage A, i.e., $n_A$, is set to 1.

For Digits-DG, the feature extractor is constructed by four $3 \times 3$ conv layers (64 kernels), each followed by ReLU and $2 \times 2$ max-pooling, following [43]. The feature extractor and global classifier are trained by SGD optimizer with learning rate of 0.05, batch size of 64 for each domain, momentum of 0.9, and weight decay of $5e^{-4}$. The number of outer loops and iteration of each stage, i.e., $n$, $n_B$, and $n_C$, are set to 4000, 8, and 6, respectively. In particular, $\lambda_1$ and $\lambda_2$ are set to 0.001 and 0.01 for Digits-DG. For PACS, Office-Home and miniDomainNet, ResNet-18 pretrained on ImageNet is used as a feature extractor removing the last layer, and $\lambda_1$ and $\lambda_2$ are set to 10 and 100, respectively. Experiments using pretrained ResNet-50 as the feature extractor are also conducted on PACS and Office-Home. The feature extractor and global classifier are trained by AdamW optimizer with an initial learning rate of $1e^{-3}$, batch size of 64 for each domain and weight decay of $5e^{-4}$. We set $n = 800$, $n_B = 8$, and $n_C = 2$ for Office-Home, $n = 6000$, $n_B = 6$, and $n_C = 2$ for miniDomainNet. Standard data augmentation including crops of random size, random horizontal flips, random color jitter, and randomly converting images to grayscale are used, following [59] and [60].

C. Performance Comparison

In this section, we compare our method with a series of domain generalization methods, presenting reported accuracy on each dataset. In existing domain generalization datasets, the domain shift is mainly reflected by the image style as shown in Fig. 4. Hence some works develop their methods based on the prior knowledge about domain shift, e.g., the style randomization. For a fair comparison, methods are divided into two groups according to whether the prior knowledge of domain shift is used. Our method deals with domain generalization without the need for prior knowledge and can be applied to various datasets with more general domain shifts.

1) Evaluation on Digits-DG: The domain generalization results on the Digits-DG benchmark are reported in Table I. For all compared approaches, we summarize the results reported in their original papers. Though some comparison methods do not apply data augmentation for data preprocessing on Digits-DG dataset, their algorithms still involve some operations which increase the diversity of training data [42], [43], [61]. For a fair comparison, we provide the
out-of-distribution generalization performance of the vanilla Empirical risk minimization method DeepAll w/o aug and our proposed CMCL w/o aug, which do not use data augmentation, and the out-of-distribution generalization performance of DeepAll w/ aug and CMCL w/ aug which use standard data augmentation. When without data augmentation, we can see that our proposed CMCL w/o aug achieves a 4.44% performance improvement over DeepAll w/o aug. DeepAll w/ aug already surpasses many comparison methods. But our proposed CMCL w/ aug still achieves the best performance in average accuracy and significantly surpasses the second-best competitor, DeepAll w/ aug, by a large margin of 1.97%. Specifically, CMCL w/ aug outperforms competitors on MNIST, SVHN and SYN. The improvement on MNIST-M is not as significant as those on other domains, mainly due to its dissimilarity with source domains as shown in the second row in Fig. 4. On the contrary, image augmentation-based methods, DDAIG, L2A-OT, and feature augmentation-based method, SFA-A, obtain larger improvement on MNIST-M but perform worse than CMCL on other ones. Probably because the domain shift of MNIST, SVHN, and SYN are independent of image style and texture, the proposed CMCL which does not rely on any prior knowledge of domain shift works better on these domains. Note that the proposed CMCL has great advantages over CCSA and MMD-AAE, which are all domain-invariant representation-based methods.

2) Evaluation on PACS: Results on PACS with ResNet-18 and ResNet-50 are presented in Tables II and III, respectively.
For all competitors, we summarize the results reported in their original papers. We can observe that CMCL outperforms all comparison approaches on average accuracy with ResNet-18 as the feature extractor and obtains comparable performance with the reported best approach with ResNet-50 as the feature extractor. The experiments on feature extractors of different sizes further prove the effectiveness of our CMCL. Specifically, CMCL achieves the best accuracy on Sketch and the second-best accuracy on Art in Table II and obtains the best performance on Sketch and Cartoon in Table III. We notice that there is a performance drop on Photo compared to the vanilla empirical risk minimization method DeepAll. It is probably because of the ImageNet pretraining. As explained in [80], models pretrained on ImageNet may be biased toward texture, and finetuning those models on PACS using empirical risk minimization may inherit this bias, thus leading to better performance on Photo which is similar to ImageNet.

3) Evaluation on Office-Home: Experimental results with ResNet-18 as the feature extractor are reported in Table IV. For all comparison methods, we summarize the results reported in their original papers. We also report the results with ResNet-50 in Table V. For all comparison methods in Table V, we summarize the results reported in a published work [59], which uses the same experiment settings as ours, including data augmentation, model selection, and data splitting. From Table IV, we can see that our method again achieves the best average accuracy compared to the competitors, though the domain discrepancy of Office-Home is less than other datasets, which is unfavorable for CMCL to eliminate the spurious correlation in datasets and enhance generalization. Due to the similarity to ImageNet, DeepAll, which combines all data from source domains to train a model, acts as a strong baseline and beats a series of DG methods, e.g., DSON, MetaNorm, SagNet, MMD-AAE, CrossGrad, and RSC. Nevertheless, our method still lifts the performance of DeepAll by a margin of 4.12% on Clipart, and 0.99% on average accuracy. Besides, comparable results with other competitors are also obtained in other domains. From Table V, we can observe that CMCL exceeds all comparison methods and achieves the best results on all domains. When using a larger backbone, which replaces ResNet-18 with ResNet-50, the performance of CMCL gets significantly improved with a large margin of 5.15%, demonstrating that our method has a nontrivial improvement in the generalization on unseen domains.

4) Evaluation on miniDomainNet: We additionally carry out experiments on a large-scale dataset, miniDomainNet and report results in Table VI. For a fair comparison, we cite the results of comparison methods from a published work [86]. We can observe that CMCL achieves the best performance on all domains and outperform the second-best method by a large margin of 3.22% on average accuracy. Our method obtains a more significant improvement on the baseline when the dataset gets larger, which further proves the superiority of CMCL.

### D. Further Analysis

In this section, we conduct a series of experiments to further analyze our method.
TABLE VII
ABLATION STUDY OF DIFFERENT COMPONENTS OF OUR CMCL WITH RESNET-18 AS THE BACKBONE

| Method       | PACS | Office-Home | miniDomainNet |
|--------------|------|-------------|---------------|
|              | Art  | Cartoon     | Photo         | Sketch | Avg  | Product | Real | World | Avg  | Clipart | Painting | Real | Sketch | Avg  |
| CMCL         | 84.55 | 80.08 | 94.95 | 82.84 | 85.61 | 60.14 | 53.52 | 73.57 | 75.53 | 65.69 | 69.59 | 63.37 | 68.42 | 63.29 | 66.17 |
| - w/o \(L_{\text{mean}}\) & \(L_{\text{cov}}\) | 83.02 | 77.63 | 94.63 | 81.45 | 84.18 | 56.35 | 51.83 | 72.66 | 74.65 | 63.88 | 69.52 | 62.10 | 67.29 | 62.42 | 65.33 |
| - w/o \(L_{\text{mean}}\)            | 83.51 | 78.07 | 94.97 | 80.74 | 84.29 | 58.51 | 51.84 | 72.88 | 74.43 | 64.54 | 69.56 | 62.33 | 67.62 | 62.46 | 65.50 |
| - w/o \(L_{\text{cov}}\)          | 83.74 | 80.05 | 94.87 | 81.52 | 85.65 | 59.28 | 53.03 | 73.58 | 75.86 | 65.29 | 69.81 | 62.79 | 67.25 | 62.48 | 65.70 |
| - w/o Mean Classifier             | 84.93 | 79.11 | 94.33 | 81.78 | 84.81 | 58.48 | 51.59 | 73.42 | 76.20 | 64.77 | 69.65 | 62.86 | 67.12 | 62.50 | 65.53 |
| - w/o EMA                        | 83.33 | 79.12 | 93.83 | 82.80 | 84.74 | 54.36 | 51.10 | 71.55 | 72.71 | 62.43 | 66.54 | 59.24 | 65.76 | 60.33 | 62.97 |

The best result is in bold face. Underlined ones represent the second-best results.

1) Rationality of Assumption 1: In Section III-B, an assumption, i.e., the ground-truth marginal distribution lies in the convex hull of source domains, is proposed as the basis of problem formalization. Under this assumption, a test domain sampled from the ground-truth distribution with a selection bias is naturally near to the ground-truth distribution and lies inside of the convex hull. Here we empirically analyze the rationality of the assumption. As shown at the second row in Fig. 4(a), MNIST-M is obviously different from other domains, and the domain shift in it is obviously different from that of others. MNIST-M probably does not lie inside the convex hull of other domains, which means that the assumption is not well met. From Table I, we can observe that all reported domain generalization methods perform worst in MNIST-M among all test domains. Hence we can conclude that Assumption 1 is necessary and reasonable for distribution alignment-based domain generalization.

2) Effectiveness of Each Component of CMCL: We discuss the effectiveness of \(L_{\text{mean}}\) and \(L_{\text{cov}}\) in (16), Mean Classifier in (21), and EMA in (23). The results on PACS, Office-Home, and miniDomainNet with ResNet-18 as the feature extractor are reported in Table VII. As shown in Table VII, we can observe that removing any component of CMCL can lead to significant performance degradation, demonstrating the effectiveness of our design.

CMCL w/o \(L_{\text{mean}}\) & \(L_{\text{cov}}\) is a variant which optimizes the objective in (10) without considering the constrained condition of marginal distribution alignment. Unless the marginal distributions of source domains are naturally aligned, the optimization objective of CMCL w/o \(L_{\text{mean}}\) & \(L_{\text{cov}}\) is obviously different from CMCL. The former considers minimizing the KL-divergence between domain-specific posterior distributions given samples from source domains, while the latter tries to minimize the KL-divergence given any samples from the real-world distribution, which is more general. Obviously, the latter works better as shown in Table VII.

As shown in (16), we adopt \(\lambda_1\) and \(\lambda_2\) to control the penalty intensity of \(L_{\text{mean}}\) and \(L_{\text{cov}}\), respectively. To illustrate the significance of \(\lambda_1\) and \(\lambda_2\), parameter sensitivity analysis is also conducted as shown in Fig. 5. We update the global classifier \(W_x\) by the mean of domain-specific classifiers, denoted as Mean Classifier, at Stage B. As the training progresses, the domain-specific classifiers tend to be consistent. So the Mean Classifier is a reasonable prediction of the convergence of domain-specific classifiers. From Table VII, we can see that Mean Classifier makes a significant contribution to the final performance. As demonstrated in Section III-C2, EMA helps to improve the generalization of the trained model and the stability of the training process. The former can be verified by CMCL w/o EMA in Table VII. The latter is further illustrated in Fig. 6. The training accuracy curves of target models, which are updated by EMA, are smoother than that of online models.

3) Effectiveness of Maximum In-Domain Likelihood: The term of maximum in-domain likelihood in (6) is proposed to eliminate the side-effect of KL-divergence as mentioned in Section III-B1, and then the CMCL optimization problem can be deduced. To evaluate the effectiveness of the term of maximum in-domain likelihood, CMCL-KL, a variant of CMCL, is constructed by removing the term of maximum in-domain likelihood in (6). CMCL-KL is also optimized by an alternating optimization strategy. Specifically, the first term of (22) is removed and the KL-divergence between
Comparison Among Different Variants With ResNet-18 as the Backbone

| Method   | Art | Cartoon | Photo | Sketch | Avg. | Artistic | Clipart | Product | Real World | Avg. | Clipart | Painting | Real | Sketch | Avg. |
|----------|-----|---------|-------|--------|------|----------|---------|---------|------------|------|---------|----------|------|--------|------|
| CMCL-KL  | 83.36 | 79.32  | 94.49 | 80.76  | 84.48 | 58.32    | 52.26   | 73.33   | 75.31      | 64.38 | 69.85   | 62.61    | 67.45| 62.91  | 65.66|
| E2E-KL   | 83.53 | 79.31  | 95.01 | 80.77  | 84.66 | 57.38    | 52.42   | 72.26   | 75.33      | 63.30 | 69.27   | 62.35    | 67.14| 62.93  | 65.57|
| CMCL     | 84.55 | 80.98  | 84.95 | 85.61  | 86.14 | 53.52    | 73.57   | 75.53   | 65.69      | 62.53 | 63.37   | 68.42    | 63.29| 66.17  |

The best result is in bold face. Underlined ones represent the second-best results.

Fig. 7. Average entropy values in validation and test sets of models trained with and without maximum in-domain likelihood term. CMCL: models trained with maximum in-domain likelihood, CMCL-KL: models trained without maximum in-domain likelihood.

5) Effect of the Number of Source Domains and the Number of Training Samples: To analyze the effect of the number of source domains, we provide results of changing the number of source domains in Table IX. In Table IX, “C + P” means that images of the Cartoon domain and Photo domain are aggregated to construct a new domain, and the same goes for others. As shown in Table IX, we can see that models trained with three source domains can achieve higher accuracy than those trained with two source domains. What is more, the diversity of source domains is also important. For example, as shown in Fig. 4(b), we can empirically observe that the difference between the “C + P” (Cartoon + Photo)” domain and the “S (Sketch)” domain is more significant than that between the “C (Cartoon)” domain and the “P + S (Photo + Sketch)” domain. So models trained with the “C + P” domain and “S” domain perform better than those trained with the “C” domain and “P + S” domain.

In Table X, we provide test accuracy of models trained with different proportions of raw training samples. We can see that the models trained with more training samples achieve higher accuracy, which is to be expected.

6) Feature Visualization: To qualitatively assess the ability of CMCL in learning the domain-invariant classifier,
The constrained optimization problem is transformed into an unconstrained one by adding a penalty term and approximately solved by an alternating optimization strategy. CMCL naturally realizes the joint distribution alignment by solving this optimization problem. Comprehensive experiments on four datasets demonstrate that our method can obtain excellent domain generalization performance.

In this work, we propose an important convex hull assumption, under which the domain-invariant classifier could generalize to unseen domains. In the future, generative methods can be coupled with CMCL to diversify training domains so that the marginal distribution of the real world is more likely to be located in the convex hull of that of training domains.

### References

[1] M. Long, H. Zhu, J. Wang, S. Wang, and M. I. Jordan, “Unsupervised domain adaptation with residual transfer networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 136–144.

[2] H. Li, Y. Wang, R. Wan, S. Wang, T. Li, and A. C. Kot, “Domain generalization for medical imaging classification with linear-dependency regularization,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 3118–3129.

[3] L. Zhang et al., “Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation,” *IEEE Trans. Med. Imaging*, vol. 39, no. 7, pp. 2531–2540, Jul. 2020.

[4] K. Muandet, D. Balduzzi, and B. Scholkopf, “Domain generalization via invariant feature representation,” in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 10–18.

[5] H. Li, S. J. Pan, S. Wang, and A. C. Kot, “Domain generalization with adversarial feature learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5400–5409.

[6] Y. Li et al., “Deep domain generalization via conditional invariant adversarial networks,” in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 624–639.

[7] Y. Li, M. Gong, X. Tian, T. Liu, and D. Tao, “Domain generalization via conditional invariant representations,” in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 579–587.

[8] I. Albuquerque, J. Monteiro, M. Darvishi, T. H. Falk, and I. Mitliagkas, “Generalizing to unseen domains via distribution matching,” 2019, arXiv:1911.08084.

[9] A. Robey, G. J. Pappas, and H. Hassani, “Model-based domain generalization,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 20210–20229.

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

[11] C. Heinz-Deml, J. Peters, and N. Meinshausen, “Invariant causal prediction for nonlinear models,” *J. Causal Inference*, vol. 6, no. 2, Sep. 2018, Art. no. 20170016.

[12] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, “Invariant risk minimization,” 2019, arXiv:1907.02893.

[13] M. Koyama and S. Yamaguchi, “Out-of-distribution generalization with maximal invariant predictor,” 2020, arXiv:1907.02893.

[14] K. Ahuja et al., “Invariance principle meets information bottleneck for out-of-distribution generalization,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 3438–3450.

[15] B. Li et al., “Invariant information bottleneck for domain generalization,” in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 7399–7407.

[16] H. Zhao, R. T. D. Combes, K. Zhang, and G. J. Gordon, “On learning invariant representations for domain adaptation,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7523–7532.

[17] S. Kullback and R. A. Leibler, “On information and sufficiency,” *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[18] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, “Heterogeneous domain adaptation through progressive alignment,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1381–1391, May 2019.

[19] Z. Wang, B. Du, and Y. Guo, “Domain adaptation with neural embedding matching,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2387–2397, Jul. 2020.
