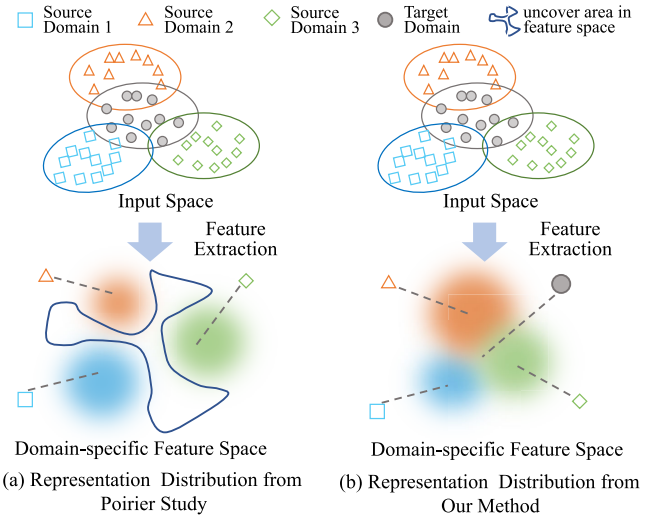
Continuous Disentangled Joint Space Learning for Domain Generalization

Zizhou Wang, Yan Wang, Yangqin Feng, *Member, IEEE*, Jiawei Du, Yong Liu, Rick Siow Mong Goh, and Liangli Zhen

***Abstract*— Domain generalization (DG) aims to learn a model on one or multiple observed source domains that can generalize to unseen target test domains. Previous approaches have focused on extracting domain-invariant information from multiple source domains, but domain-specific information is also closely tied to semantics in individual domains and is not well-suited for generalization to the target domain. In this article, we propose a novel DG method called continuous disentangled joint space learning (CJSL), which leverages both domain-invariant and domain-specific information for more effective DG. The key idea behind CJSL is to formulate and learn a continuous joint space (CJS) for domain-specific representations from source domains through iterative feature disentanglement. This learned CJS can then be used to simulate domain-specific representations for test samples from a mixture of multiple domains via Monte Carlo sampling during the inference stage. Unlike existing approaches, which exploit domain-invariant feature vectors only or aim to learn a universal domain-specific feature extractor, we simulate domain-specific representations via sampling the latent vectors in the learned CJS for the test sample to fully use the power of multiple domain-specific classifiers for robust prediction. Empirical results demonstrate that CJSL outperforms 19 state-of-the-art (SOTA) methods on seven benchmarks, indicating the effectiveness of our proposed method.**

***Index Terms*— Domain generalization (DG), feature disentanglement, robust machine learning.**



(a) Representation Distribution from Poirier Study (b) Representation Distribution from Our Method

Fig. 1. Distribution of domain-specific representations after feature disentanglement. (a) Previous methods focus on learning a universal feature extractor (e.g., using meta-learning). (b) Our method. Traditional approaches may result in the target domain falling into uncovered areas due to the gap between the target and source domains. Our method mitigates this issue by learning a CJS.

I. Introduction

MACHINE learning, especially deep learning, has enabled remarkable advances in various applications over the past decade. The common assumption in machine learning is that the test and training data are independent and identically distributed (IID) [1], [2], [3]. However, training and test data samples are typically captured in various contexts in practice. This assumption is frequently unattainable due to the widespread and pronounced distribution shifts [4], [5], [6], [7]. As a result, the trained models struggle to cope with distribution shifts and suffer from performance degradation on out-of-distribution test data. To address this issue, domain generalization (DG), as a promising solution, aims to learn a model from one or multiple different but related source domains to generalize to unseen domains.

Most DG methods are concentrated on discovering domain-invariant representations across different domains, while reducing the impact of domain-specific information. They have achieved encouraging outcomes on various tasks, such as image classification [8], [9], [10], [11], object detection [3], [12], [13], and semantic segmentation [14], [15], [16], [17]. It is noteworthy that the efficacy of these methods is contingent upon accurate disentanglement of feature vectors according to the task at hand. Ensuring that domain-specific information is distinctly separate from category information remains a challenging yet crucial aspect to attain optimal results [18]. The theoretical foundations established in prior studies [19], [20] also highlight the detrimental effects of excessive learning of domain-invariant feature vectors in the context of DG.

In recent years, numerous efforts have been dedicated to exploring the concurrent optimization of domain-invariant and domain-specific mappings to improve generalization performance. Bui et al. [21] provided a comprehensive demonstration of this optimization and articulate the conditions under which it can be effectively applied. Nevertheless, the diversity

Manuscript received 9 March 2023; revised 16 January 2024 and 19 June 2024; accepted 28 August 2024. This work was supported by the National Research Foundation of Singapore under its AI Singapore Program (AISG Award) under Grant AISG2-TC-2021-003. *(Corresponding author: Liangli Zhen.)*

The authors are with the Institute of High Performance Computing, Agency for Science, Technology and Research (A*STAR), Singapore 138632 (e-mail: wang_zizhou@ihpc.a-star.edu.sg; wangyan@ihpc.a-star.edu.sg; fengyq@ihpc.a-star.edu.sg; dujw@ihpc.a-star.edu.sg; liuyong@ihpc.a-star.edu.sg; gohsm@ihpc.a-star.edu.sg; zhenll@ihpc.a-star.edu.sg).

Digital Object Identifier 10.1109/TNNLS.2024.3454689

among specific feature vectors in individual domains poses a significant challenge to the learning process. Currently, methods tend to use transfer weight representations based on meta-learning to adapt to unseen target domains. Nonetheless, in accordance with the findings of these studies [21], the precision in extracting target domain-specific feature vectors becomes unattainable when these vectors are entangled with features from other categories. This suggests that a feature extractor trained on a source domain may not be effective when applied to another target domain. As mentioned in the previous work [22], the absence of constraints on specific feature vectors from various domains in free encoding methods can lead to an aggregation of feature vectors in limited locations, potentially resulting in uncovered areas, as depicted in Fig. 1(a). When subjected to test samples from the target domain, a free coding model's output may fall within these uncovered areas, compromising the precision of specific feature representation. Consequently, ensuring that the domain-specific feature representation enhances the model's performance becomes a challenging endeavor.

To tackle this challenge, in this article, we present a novel feature disentanglement method, referred to as continuous disentangled joint space learning (CJSL), to extract both domain-invariant and domain-specific information for DG. Specifically, our method involves the design of a shared feature extractor for all the source domains to perform feature disentanglement. This shared feature extractor is connected to multiple classifiers (Multi-CLS), one for each source domain, to learn a robust model through supervised learning. The key idea of CJSL is the formulation and learning of a continuous joint space (CJS) [illustrated in Fig. 1(b)] for domain-specific representations from the source domains during the iterative feature disentanglement process. To accomplish this, we use probability density estimation, a statistical tool that describes the distribution of the feature space of each domain across a continuous space. Each domain is then approximated as a Gaussian model, and the joint feature space of multiple domains is represented as a Gaussian mixture model (GMM). The learned CJS is then used to simulate multiple domain-specific representations, each as a mixture of known domain descriptions, using Monte Carlo sampling for test samples. Finally, the simulated domain-specific representation and the domain-invariant representation are concatenated for robust prediction using the ensemble of multiple trained domain-specific classifiers. Our method combines the classification outcomes from each domain to obtain the category of unseen samples. This is done by mapping the unseen target domains to the joint distribution of several known source domains.

The novelty and main contributions of this work are summarized as follows.

1) We propose a novel method, continuous joint space learning, for feature disentanglement and synthesis to enhance DG. Unlike conventional methods that rely solely on domain-invariant feature vectors or domain-specific feature extractors, CJSL uses Monte Carlo sampling in a learned CJS to simulate domain-specific representations for test samples, leveraging multiple domain-specific classifiers for robust prediction.
2) We introduce a strategy for learning a CJS for domain-specific representations derived from multiple source domains through iterative feature disentanglement. This joint space, modeled as a mixture of Gaussian distributions, allows for the simulation of representations incorporating stylistic elements from various source domains during inference, thus improving DG.
3) Extensive experiments have been conducted on seven public benchmarks to evaluate the effectiveness of our proposed method by comparing with 19 peer methods. Empirical results show that our method outperforms current state-of-the-art (SOTA) methods on these benchmarks, especially in depicting large domain shifts.

II. Related Work

As a technique for generalizing models to unseen target domains, DG has attracted increasing attention from researchers and practitioners from both academia and industry. To improve the model's generalization capability, pioneer methods propose using the data manipulation strategy to generate more data for model training since machine learning models' performance often relies on the quantity and diversity of the training data. These data manipulation-based methods primarily narrow the distance between the target and source domains by augmenting a wide variety of data [23], [24]. For instance, Yue et al. [8] proposed to use simulation images to improve models' DG abilities for semantic segmentation of real-world autonomous driving scenarios. Domain flow generation (DLOW) [25] generates a sequence of intermediate domains to bridge two distinct domains, enabling smoother domain adaptation and the creation of novel image styles by using cycle-consistent generative adversarial networks (CycleGANs) with the domainness variable. Learning to augment by optimal transport (L2A-OT) [26] addresses the DG challenge using a data generator to create diverse pseudo-novel domains, using optimal transport to maximize domain divergence and incorporating cycle-consistency and classification losses to maintain data semantics. Instead of manipulating data in input space, a series of feature vector augmentation methods [27], [28] are presented. These methods generate representations in the feature space using its statistics modification. In addition to data manipulation, different learning strategies, such as self-supervised learning, ensemble learning, meta-learning, and distributionally robust optimization, are also exploited to improve models' generalization performance [29], [30], [31], [32], [33]. For example, Carlucci et al. [34] proposed solving jigsaw puzzles as a supplemental task to induce models to learn spatially relevant concepts and extract generalized feature vectors. Zhang et al. [10] introduced a meta-learning framework to enhance the model by simulating domain transfer during training.

In recent years, the majority of DG approaches have focused on extracting domain-invariant information across various source domains. The assumption behind these methods is that these learned domain-invariant feature vectors can be applied to any domain, including unseen domains, as demonstrated in prior studies such as [35], [36], and [37]. The basic concept of these methods involves reducing the discrepancy among

multiple source domains by aligning their data distributions in a latent space through a variety of techniques [18], [38], [39]. For instance, Zhang et al. [40] introduced Style-uncertainty Augmentation (SuA) as a simple yet effective feature-based technique for domain generalizable person re-identification (DG ReID), addressing challenges in model generalization to unseen domains by perturbing instance styles with Gaussian noise during training, achieving SOTA results on four large-scale benchmarks. Xu et al. [41] proposed a multiview adversarial discriminator (MAD) for DG in object detection, addressing the limitations of previous domain adversarial learning methods by incorporating multiview adversarial training to remove noncausal factors from common features. Meng et al. [42] tackled the domain bias issue in convolutional neural networks through the distraction framework, leveraging intra- and intermodel distraction regularization to reallocate attention to diverse task-relevant features. The approach achieves SOTA performance across various benchmarks by effectively modeling domain bias, partitioning the attention graph, and aggregating it into task-relevant and domain-relevant groups.

Recent research has shown that domain-specific feature vectors are also crucial and can enhance DG performance while learning invariant feature vectors [18], [19], [21]. For example, Bui et al. [21] performed a theoretical analysis, which provides a fundamental foundation for domain-specific representation learning in DG. Our proposed method also leverages both domain-invariant and domain-specific information to improve DG performance. However, it differs from the existing methods by learning a CJS for all the domains and then simulating domain-specific domains for target test samples, enabling the full utilization of multiple domain-specific classifiers, rather than learning a universal domain-specific feature extractor for all the domains.

III. Our Proposed Method

A. Problem Formulation

We consider the problem of DG, where we are given a set of N source domains $D^{\text{train}} = \{D^1, D^2, \ldots, D^N\}$. Each domain D^i is a distribution of the joint space of the input data x and the corresponding label y: $P^i(x, y) \sim \mathcal{PX} \times \mathcal{Y}$. The joint distributions of each pair of domains are different: $P^i(x, y) \neq P^\tau(x, y)$ when $i \neq \tau \in [1, N]$. The aim of DG is to learn a generalizable inference function $f: \mathcal{X} \to \mathcal{Y}$ from the source domains to achieve a minimum prediction error on an unseen target domain D^t with $P^t(x, y) \sim \mathcal{PX} \times \mathcal{Y}$ and $P^{\text{test}}(x, y) \neq P^i(x, y)$ for $i \in [1, N]$. Note that D^t is not accessible for model training. In other words, we aim to learn a model that can generalize to unseen target domains by exploiting the commonalities across the observed source domains.

B. Framework of CJSL

As shown in Fig. 2, we present the overview of our CJSL method, which includes the training and inference phases. In the training phase, we design a feature extractor containing two subbranches: a domain-invariant feature vector learning branch and a domain-specific feature vector learning branch. On the top of the feature extractor, we connect Multi-CLS (each for one source domain) to achieve image classification, a coupled feature domain discriminator (CFDD) to distinguish which domain the input sample comes from, and an image reconstruction (IR) module to recover the input sample using the feature representation. Specifically, the feature extractor will map the input sample to domain-specific and domain-invariant feature spaces with $F_S: \mathcal{X} \to \hat{\mathcal{Z}}_S$ and $F_I: \mathcal{X} \to \mathcal{Z}_I$. The subbranch F_S is enforced to map the input samples into a CJS defined by a mixture of Gaussian distributions. Furthermore, the domain-invariant and domain-specific feature vectors are disentangled using adversarial learning between the feature extractor and CFDD. Once the domain-invariant and domain-specific representations are obtained, we concatenate them and classify the concatenated representation into one source domain. Then, we select the corresponding domain-specific classifier to conduct the classification for the concatenated representation. At the same time, using the concatenated representation, we adopt IR to recover the input sample, thus encouraging the feature extractor to learn more discriminative feature vectors.

In the inference phase, we use the trained domain-invariant feature extraction branch and the learned CJS (characterized by the learned mixture of Gaussian distributions) to classify the test samples. To be specific, we use F_I to obtain the domain-invariant representation z_I for the input test sample. Simultaneously, we use Monte Carlo sampling on the learned CJS to simulate multiple domain-specific representations for each test sample. Each simulated representation will be concatenated with the domain-invariant representation for robust classification with CFDD and the Multi-CLS.

C. Continuous Disentangled Joint Space

In the following, we will introduce how to formulate and learn the CJS for domain-specific representation learning and simulation in feature disentanglement.

Assumption 1: The domain-specific feature vectors correlate with the semantics of their corresponding domains, and they can be mapped into a continuous joint latent space. In addition, the target domain-specific feature vectors can be represented as a mixture of domain-specific feature vectors of all the observed source domains.

Assumption 1 indicates that: 1) the source domains are relevant and each domain-specific feature space is a subspace of latent joint space and 2) the target domain belongs to the style as a mixture of observed source domains. Based on this assumption, we can learn the CJS for capturing the domain-specific feature vectors of source domains and use it to simulate domain-specific representations for the test samples.

Domain-specific features, tailored to each domain, pose challenges for precise estimation, particularly in the context of an unfamiliar target domain. In this investigation, we conceptualize the domain-specific feature space for each source domain as a Gaussian distribution. To enhance simplicity and mitigate unnecessary complexities, we model the distribution of each domain-specific feature on the subspace as a Gaussian distribution with mean and variance, effectively encapsulating their

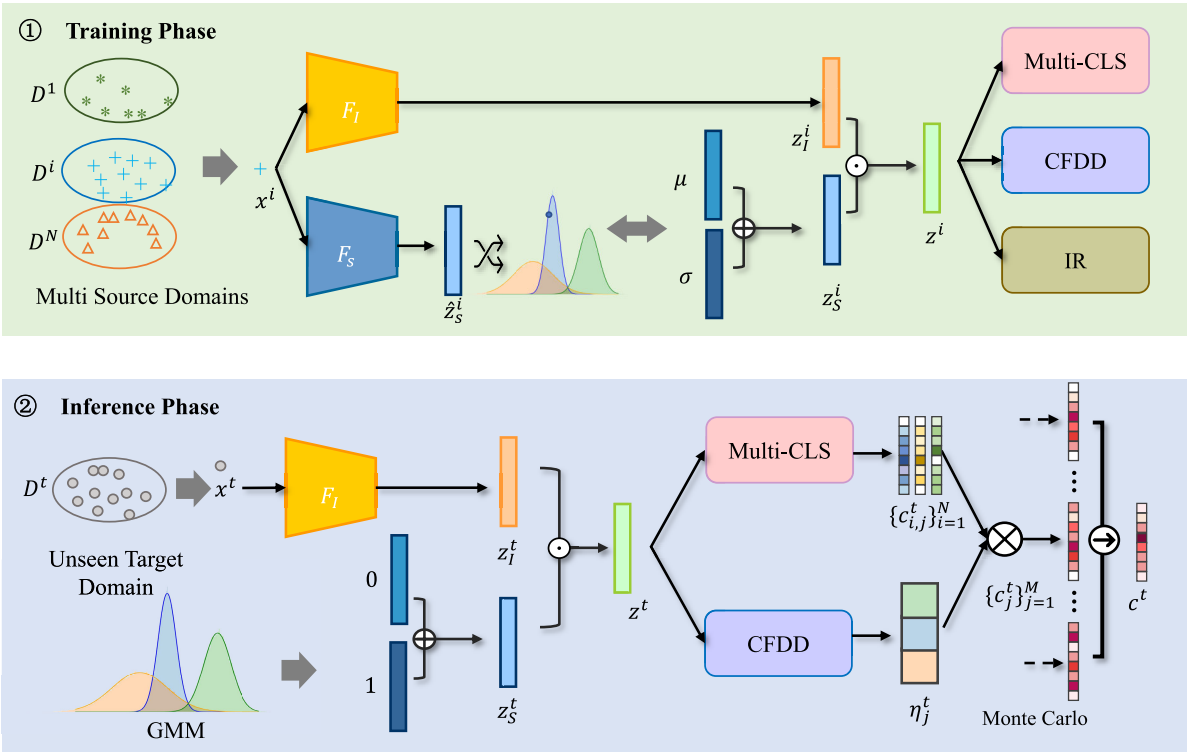


Fig. 2. Training phase and inference phase of our proposed CJSL. It contains two feature extractors F_S and F_I, an IR module, a domain discriminator (CFDD), and multiple domain-specific classifiers (Multi-CLS). z_S and z_I are derived through F_S and F_I, and then they are disentangled via CFDD. Next, the two feature vectors can be concatenated as the input for Multi-CLS. Moreover, we develop an IR to guarantee the completeness of the information. During inference, we only need to extract the invariant feature vector and sample a specific feature vector from the mixture distribution. Then we can obtain the final result c^t via the output of classifier i for the jth sampling (denoted as $c^t_{i,j}$) and the output of CFDD for the jth sampling (denoted as η_j) according to the Monte Carlo with Bayesian.

domain-relevant characteristics rather than semantic information. This streamlined approach is meticulously designed to achieve a balance between simplicity and effectiveness in capturing domain-specific attributes. As illustrated in Fig. 2, the joint space for domain-specific representations can be aptly described as a mixture of Gaussian distributions. Note that it is more efficient to stylize at the feature level than at the image level, which is in line with the current research on style transfer [43]. Specifically, for the ith source domain, we use a Gaussian model with mean μ^i and variance σ^i to fit its domain-specific representations. Then, we can define a GMM to model all the source domains. Finally, we can use the expectation–maximization (EM) algorithm based on maximum likelihood estimation to learn the parameter values of GMM. As a result, the domain-specific latent space can be continuous, and each location has meaningful outputs that can be readily interpolated across observed source domains. The process is illustrated in Fig. 3, where the unseen target domain D^t can be represented by mixing up several known source domains as follows:

$$\{D^1, D^2, \ldots, D^N\} \xrightarrow{\text{GMM}} D^t. \tag{1}$$

We implement domain-specific feature learning using a technique akin to variational autoencoder (VAE)-type variational inference. A continuous distribution of specific feature vectors of multiple domains can be obtained in the following manner. The feature extractor is enforced to map

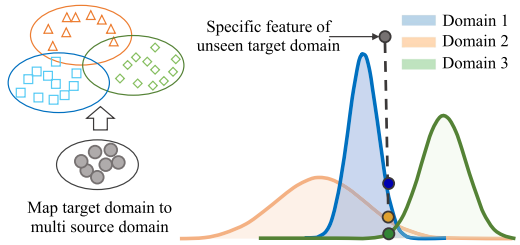


Fig. 3. Link between the target domain with source domains. In the continuous latent space of domain-specific representations, we can map a given unseen target domain sample to a mixture of several known source domains. We use sampling to simulate the feature vectors we need since we are unable to extract accurate specific feature vectors for unknown domains. We believe that the specific feature vectors of each domain belong to a common Gaussian mixture distribution, which forms the basis of our use of sampling.

domain-specific representations $\hat{z}_S$ to a Gaussian distribution with the mean μ and variance σ of the feature space. Then, we can sample a feature vector z_S from the distribution $\mathcal{N}(\mu, \sigma)$. As a result, the model can learn domain-specific feature vectors more efficiently and diversely. In addition, the model can link different domains to the feature space, making every point in the feature space valid. The learned Gaussian distribution is enforced to be close to a standard Gaussian distribution by minimizing the following loss function:

$$\mathcal{L}_J = \mathrm{KL}\big(\mathcal{N}(\mu, \sigma)||\mathcal{N}(0, 1)\big) \tag{2}$$

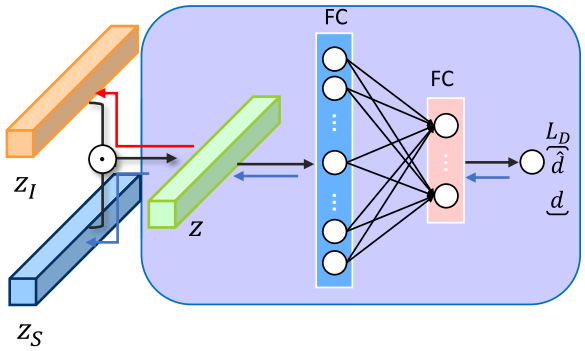


Fig. 4. CFDD architecture is designed with the expectation that z_S and z_I convey distinct information. In our augmentation strategy, we not only incorporate domain-invariant features but also include domain-unique features, enhancing the classifier's discriminative capacity. To delineate specific roles for these features, we introduce a mechanism wherein domain-invariant features prompt a gradient reversal during backpropagation. Consequently, in optimizing the CFDD module, we guide z_S and z_I in divergent directions. During the backpropagation process, GRL is introduced to invert the gradient of z_I (illustrated by red arrows), while the gradient of z_S remains unaltered (depicted by blue arrows), as outlined in (3). This gradient reversal, fundamental to our adversarial learning approach, is formulated in (4).

where KL$(\cdot)$ is the Kullback–Leibler divergence. Through this constraint, the diversity of features can be well-guaranteed, and the continuous space will not shrink to a small area. Following the analogy to VAE, both the methods share a foundational assumption, allowing our method to sample domain-specific features from a standard Gaussian distribution $N(0, 1)$ to guarantee the continuity of feature distribution. Thus, we transform the process of feature extraction into feature generation. Using GMM, it is feasible to generate known specific feature vectors for an unknown domain by sampling. Based on the previous training, the known feature vectors generated from the GMM, when combined with the invariant feature vectors, can improve the accuracy (ACC) of the model by providing additional information for the unseen domains.

D. Coupled Feature Domain Discriminator

The disentanglement relationship between the two feature vectors, z_S and z_I, is not guaranteed by the two feature extraction branches. To ensure disentanglement, we validate the preservation of information through a data reconstruction module. Subsequently, disentanglement conditions are specified to differentiate between the two feature representations. Motivated by image generation research, we use adversarial learning to optimize the domain discriminator and feature extractor in opposite directions, facilitating the learning of feature vectors that remain invariant across domains.

Specifically, we propose CFDD to discriminate which domain the feature vectors come from as

$$\hat{d} = \phi(z_S, \text{GRL}(z_I)) \tag{3}$$

where $\hat{d}$ denotes the domain prediction. The model architecture is delineated in Fig. 4. By concatenating z_S and z_I, we guide them toward distinct optimization directions using the gradient reversal layer (GRL) [44]. As previously elucidated, z_S signifies the domain style, presumed to manifest significant interdomain differences, while z_I encapsulates invariant feature information that is intended to remain undifferentiated. The adversarial relationship between F_I and ϕ is established through GRL, facilitating F_S and F_I to acquire domain-specific and invariant feature vectors, respectively. The disentanglement objective is realized through a min–max task expressed as follows:

$$\min_{F_S,\phi} \max_{F_I} \{\mathcal{L}_D := -\mathbb{E}[d \log(\hat{d})]\} \tag{4}$$

where d denotes the true domain of the sample. This strategy enhances the discriminative power of the classifier by concurrently introducing two classes of features, thereby increasing the available features for the classifier compared with traditional methods.

By merging F_S and F_I in ϕ, we can separate the feature vectors in terms of domain discrimination. This technique provides the same feature space for decomposition and stems from a unified point of view. The model can acquire a more precise representation as a result.

E. Monte Carlo Sampling-Based Bayesian

After acquiring the continuous disentangled joint space through the disentanglement process under CFDD, obtaining a domain-specific representation requires sampling due to the inaccuracies and biases associated with a single sampling, as dictated by the law of large numbers. Motivated by Bayesian networks, we introduce a Monte Carlo process based on the Bayesian theorem, involving M stochastic samplings on GMMs. This yields M domain-specific representations $(z_{S,j})$ for $j = 1, 2, \ldots, M$. Combining each domain-specific representation $z_{S,j}$ with the invariant feature vector z_I results in z_j, and then passing z_j to CFDD ϕ and Multi-CLS, we obtain the classification prediction probability c_j as

$$c_j = \frac{c^*_{i,j} \times \eta^*_{i,j}}{\sum_i \eta^*_{i,j}} = c^*_{i,j} \times \eta_j$$
$$\text{s.t. } \eta_j = \frac{\eta^*_{i,j}}{\sum_i \eta^*_{i,j}} \tag{5}$$

where $c^*_{i,j}$ is considered as the prior probability, which represents the probability of jth sampling for classification in the domain i perspective, and $\eta^*_{i,j}$ represents likelihood of the jth sampling for classification in the domain i with $\eta_j = \phi(z_{S,j}, z_{I,j})$ is the standardized likelihood of the continuous latent space, and the one domain classification c_j of the image is calculated according to the formula, which is the posterior probability we expect. After that, we perform forward passes of the model under random dropout and obtain a set of classification prediction probability: (c_j) for $j = 1, 2, \ldots, M$. Consequently, the classification results from the M samplings are then averaged using the ensemble algorithm, providing the final classification result.

During training, Monte Carlo dropout is used to simulate multiple samplings without the need for multiple feedforwards. The classification objective function $\mathcal{L}_M$ is defined as the negative expected value of the loss $L_{i,j}$, ensuring the same expectation despite dropout, as given by the following

equation:

$$\mathcal{L}_M = -\mathbb{E}[L_{i,j}] = -\mathbb{E}\left[y \log\left(c^*_{i,j}\right)\right] \tag{6}$$

where $L_{i,j}$ denotes the loss of the ith classifier in the jth sampling and y is the true label for the image. Such individual training can improve the relevance and sensitivity of each classifier for its relevant domain. This enhances the relevance and sensitivity of each classifier for its respective domain. To ensure similarity in the sampling of domain-specific representations for model stability, we introduce an entropy measure to quantify the differences. The entropy, as defined in the following equation, is designed to limit the consistency of expressions across domains and sampling:

$$\begin{aligned} \mu_i &= \frac{1}{M}\sum_j c^*_{i,j} \\ \mathcal{L}_u &= -\sum_i \mu_i \log \mu_i \end{aligned} \tag{7}$$

where minimizing the entropy can limit the consistency of expressions across domains and sampling. Both the losses, $\mathcal{L}_M$ and $\mathcal{L}_u$, are combined to create a new overall objective function $\mathcal{L}_C$, optimizing F_S, F_I, and Multi-CLS, as expressed in the following equation:

$$\mathcal{L}_C = \mathcal{L}_M + \mathcal{L}_u. \tag{8}$$

During the inference phase, Monte Carlo simulation is used to predict test samples. The process involves calculating the invariant feature representation using the domain-invariant feature extractor F_I and randomly sampling a point from the learned continuous disentangled joint space as a domain-specific feature vector z^t_S. The likelihood is computed based on $\phi(z^t_S, z^t_S)$, and the classification results are obtained by feeding the concatenation of z^t_S and z^t_I (i.e., z^t) to Multi-CLS via (5). For increased ACC, Monte Carlo simulation is executed T times, and the results are ensemble-averaged.

F. Overall Loss Function

To supervise the entire training process, we construct an objective function $\mathcal{L}$ that integrates various components, encompassing the classification loss $\mathcal{L}_M$, disentanglement loss $\mathcal{L}_D$, joint space loss $\mathcal{L}_J$, and IR loss $\mathcal{L}_R$. These defined loss functions have a one-to-one correspondence with the previously mentioned modules. The CFDD module aligns with $\mathcal{L}_D$ and uses adversarial learning to decompose features. The CJS corresponds to both $\mathcal{L}_J$ with a design rooted in the principles of VAE, ensuring IR and diverse feature sampling. Finally, the Multi-CLS module corresponds to $\mathcal{L}_C$ and comprises two components: cross-entropy $\mathcal{L}_M$ and entropy $\mathcal{L}_u$, facilitating smooth learning of semantic categories through feature optimization.

In addition, we minimize the IR loss to ensure that the model retains essential information during feature disentanglement, as expressed by the following equation:

$$\mathcal{L}_R = \|x - \hat{x}\|_2 \tag{9}$$

where $\|\cdot\|_2$ is the L_2-norm, and $\hat{x}$ is the reconstruction of input sample x using (z_S, z_I) as the decoder's input.

In summary, we integrate (2), (4), (9), and (8) to construct the objective function $\mathcal{L}$ that supervises the entire training. It is defined as follows:

$$\mathcal{L} = \alpha\mathcal{L}_C + \beta\mathcal{L}_D + \gamma\mathcal{L}_J + \mathcal{L}_R \tag{10}$$

where the hyperparameters α, β, and γ trade off the impact of the four losses during training. The training process involves optimizing the overall objective function $\mathcal{L}$, guiding the model to disentangle features, smooth the optimization process through Monte Carlo simulation, and achieve accurate domain-specific representations for improved classification performance.

IV. Experimental Study

A. Datasets and Implementation

1) Datasets: Seven popular DG benchmarks are used to evaluate the performance of the proposed method, including **Colored-MNIST** [45]: containing 70 000 samples in binary classification problem with noisy labels, over three noisy rate domains; **Rotated-MNIST** [46]: including 70 000 samples and ten classes, rotated from MNIST over six domains; **VLCS** [47]: consisting of 10 729 samples and five categories, over four domains with a small domain distribution; **PACS** [48]: consisting of 9991 images of seven categories, over four domains, with large domain discrepancy; **Office-Home** [49]: having more categories than the first two datasets, which has 15 500 daily images of 65 categories, over four domains; **Terra Incognita** [50]: containing 24 778 camera-trap image and ten categories, over four domains; and **DomainNet** [51]: having 586 575 images and 345 categories, over six domains. The last two datasets are significantly more complex than the first three ones.

2) Implementation Details: We follow up the setup of the previous methods [DomainBed and meta-Domain Specific-Domain Invariant (mDSDI)] to develop the model [21], [52]. We report the ACC of all the methods for comparing the models' performance. We use the MNIST-CNN and ResNet-50 pretrained on ImageNet [53] as the backbones and Adam [54] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as the optimizer. Their initial learning rates are 5×10^{-5} for large-size image datasets and 1×10^{-3} for small-size image datasets. Both of them introduce weight decay to avoid overfitting and set early-stop to terminate training when the performance on the validation set is stable. Following the setting in [21], all large-size images are resized to 224×224, and small-size images are resized to 28×28. In addition, we use the same training hyperparameters and data augmentation. Meanwhile, motivated by the high performance of stochastic weight averaging densely (SWAD) in a complex situation, we also introduce it as a plug-and-play regularization for our method [33] on the DomainNet dataset. This is in line with several recent SOTA methods [55]. The values of the hyperparameters have been set to $\alpha = 5$, $\beta = 1$, and $\gamma = 1$ for most experimentation. And the number of sampling M is 8.

B. Comparison With Peer Methods

To evaluate the effectiveness of our method, we compare it with 16 peer methods, mainly including empirical risk

minimization (ERM [56]), group distributionally robust optimization (GDRO [57]), marginal transfer learning (MTL [58]), adaptive risk minimization (ARM [59]), meta-learning for DG (MLDG [60]), invariant risk minimization (IRM [45]), deep correlation alignment (CORAL [61]), maximum mean discrepancy (MMD [36]), domain adversarial neural networks (DANNs [62]), class-conditional DANN (CDANN [63]), risk extrapolation (VREx [64]), interdomain mixup (Mixup [65]), style-agnostic networks (SagNets [37]), representation self-challenging (RSC [66]), mDSDI [21], SWAD [33], proxy-based contrastive learning for DG (PCL [55]), invariant DAG searching for DG (iDAG [67]), and a general multidomain generalization objective (GMDG [68]). We follow up the setup of the previous methods to develop the model [21], [52]. The results reported in this study for these peer methods are sourced from the papers by Gulrajani and Lopez-Paz [52], Bui et al. [21], Cha et al. [33], and Yao et al. [55].

1) Results on the Seven Benchmarks: The results of our method on seven benchmarks while comparing several SOTA methods are reported in Table I, from which we have the following observations.

1) Our CJSL method outperforms all the peer methods in terms of the average ACC on the seven datasets. For instance, CJSL achieves an average ACC of 70.4%, which improves the current SOTA by 2.5%. Among these benchmarks, CJSL achieves the highest average ACC on VLCS, PACS, OfficeHome, and DomainNet and the second-highest average ACC on other datasets. Compared with a popular domain-specific feature-based method (mDSDI), our method improves the average ACC from 67.9% to 70.4%. In comparison to mDSDI, we leverage continuous latent space and the Monte Carlo procedure to acquire domain-specific feature vectors rather than the previous meta-learning-based algorithm, which indicates the effectiveness of CJSL. More detailed results on each of the benchmarks are available in Tables II–V.
2) CJSL achieves only the second-highest results on two datasets with low-resolution images. For CMNIST, our method does not improve the generalization since it is a noise labeling problem that requires prior knowledge of noise ratio to adjust the distribution for better generalization. As reported in Table II, a large ratio of labels flip can cause a significant ACC drop for all the tested methods. Moreover, the methods that focus on invariant feature vectors perform well on it, such as ARM (56.2%). For RMNIST, all the methods achieve an ACC higher than 97%, and the difference among results from these methods is not significant. The detailed results in Table III show that the rotation degree does not have a significant impact on the ACC scores for all the tested methods.
3) For the five datasets with higher resolution images, CJSL achieves the highest ACC scores on four of them and comparable results with the current SOTA on the rest (i.e., Terranlnc). Moreover, there is a significant improvement over the second-best methods on Office-Home and DomainNet, which have large domain gaps compared with other datasets.
4) We investigate the limited improvement observed in certain datasets as follows. First, for COLORED-MNIST, the presence of label noise poses a challenge. Our method relies on domain-related information, making it prone to learning this noise pattern, hindering optimal performance. In the case of the TERRA dataset, our method, while comparable to leading approaches, exhibits a decrease of 0.1% in terms of ACC. This outcome may be attributed to the inherent characteristics of the dataset, which predominantly consists of nighttime imagery. In such images, objects occupy a minimal portion of the frame and are characterized by low luminance. Consequently, the extraction of critical invariant information from these challenging images is more vulnerable to interference from environmental factors, leading to a negligible enhancement in the performance of the method.

2) Evaluation on VLCS, Office-Home, and PACS: First, we evaluate and analyze more detailed results of our method on VLCS, Office-Home, and PACS. These three datasets all have four domains and contain about 10k images. From the results in Tables IV–VI, we can see that our method has a smaller improvement on the VLCS dataset than on the PACS dataset compared with the baseline method, mDSDI. As images in VLCS are gathered from real-world data and contain a wealth of domain-specific information in background and style, the domain-specific-based method can commonly produce higher ACC scores on it. Compared with VLCS, we find that Office-Home and PACS have a wider interdomain gap due to the images from different domains having completely different styles. The improvement of our method is more significant on PACS, which is consistent with the design core of our model. Our method can obtain more accurate domain-specific feature vectors, thus improving the performance of the baseline (mDSDI) from 86.2% to 88.7%. On Office-Home, CJLS boosts ACC from 69.2% to 73.5% compared with mDSDI. This more significant improvement stems from the larger gaps between different domains in Office-Home.

3) Evaluation on Terra and DomainNet: Subsequently, we analyze the results in the more complex scenario. Terra is a photograph of wildlife taken at different trap locations. Since the locations have not changed for each camera, the captured images have very similar backgrounds and styles for each domain. Thus, domain-invariant-based methods can achieve better results, as shown in Table VII. For example, SagNet achieves an ACC score of 48.6% and outperforms the baseline (mDSDI) that concatenates domain-invariant and domain-specific feature representations for learning. Our method achieves a slightly lower ACC score than the best-performing method SWAD at 52.0% versus 52.1%. In the spectrum of seven benchmark datasets, DomainNet stands out as the most expansive in terms of both image count and categories. Our method also achieves the highest ACC score at 48.4%. It is imperative to scrutinize the influence of SWAD on results and address potential fairness concerns. Table VIII delineates

TABLE I
COMPARISON OF ACC (%) WITH THE SOTA METHODS ON SEVEN BENCHMARKS

Method	CMNIST	RMNIST	VLCS	PACS	OfficeHome	Terralnc	DomainNet	Avg
ERM	51.5	98.0	77.5	85.5	66.5	46.1	40.9	66.6
IRM	52.0	97.7	78.5	83.5	64.3	47.6	33.9	65.4
GDRO	52.1	98.0	76.7	84.4	66.0	43.2	33.3	64.8
Mixup	52.1	98.0	77.4	84.6	68.1	47.9	39.2	66.7
MLDG	51.5	97.9	77.2	84.9	66.8	47.7	41.2	66.7
CORAL	51.5	98.0	78.8	86.2	68.7	47.6	41.5	67.5
MMD	51.5	97.9	77.5	84.6	66.3	42.2	23.4	63.3
DANN	51.5	97.8	78.6	83.6	65.9	46.7	38.3	66.1
CDANN	51.7	97.9	77.5	82.6	65.8	45.8	38.3	65.6
MTL	51.4	97.9	77.2	84.6	66.4	45.6	40.6	66.2
SagNets	51.7	98.0	77.8	86.3	68.1	48.6	40.3	67.2
ARM	**56.2**	**98.2**	77.6	85.1	64.8	45.5	35.5	66.1
VREx	51.8	97.9	78.3	84.9	66.4	46.4	33.6	65.6
RSC	51.7	97.6	77.1	85.2	65.5	46.6	38.9	66.1
mDSDI	52.2	98.0	79.0	86.2	69.2	48.1	42.8	67.9
SWAD	-	-	79.1	88.1	70.6	50.0	46.5	66.9
PCL	-	-	-	88.7	71.6	**52.1**	47.7	65.0
iDAG	-	-	76.2	**88.8**	71.8	46.1	47.7	66.1
GMDG	-	-	79.2	85.6	70.7	50.1	44.6	66.0
CJSL	52.2	98.1	**80.0**	88.7	**73.5**	52.0	**48.4**	**70.4**

TABLE II
COMPARISON OF ACC (%) WITH THE SOTA METHODS ON COLORED-MNIST WITH DIFFERENT RATIOS OF LABELS FLIP

Method	10% flip	20%flip	90% flip	Avg.
ERM	71.7±0.1	72.9±0.2	10.0±0.1	51.5
IRM	72.5±0.1	73.3±0.5	10.2±0.3	52.0
GDRO	73.1±0.3	73.2±0.2	10.0±0.2	52.1
Mixup	72.7±0.4	73.4±0.1	10.1±0.1	52.1
MLDG	71.5±0.2	73.1±0.2	9.8±0.1	51.5
CORAL	71.6±0.3	73.1±0.1	9.9±0.1	51.5
MMD	71.4±0.3	73.1±0.2	9.9±0.3	51.5
DANN	71.4±0.9	73.1±0.1	10.0±0.0	51.5
CDANN	72.0±0.2	73.0±0.2	10.2±0.1	51.7
MTL	70.9±0.2	72.8±0.3	**10.5±0.1**	51.4
SagNets	71.8±0.2	73.0±0.2	10.3±0.0	51.7
ARM	**82.0±0.5**	**76.5±0.3**	10.2±0.0	**56.2**
VREx	72.4±0.3	72.9±0.4	10.2±0.0	51.8
RSC	71.9±0.3	73.1±0.2	10.0±0.2	51.7
mDSDI	73.4±0.2	73.1±0.3	10.1±0.2	52.2
CJLS	74.5	71.8	10.2	52.2

TABLE III
COMPARISON OF ACC (%) WITH THE SOTA METHODS ON ROTATED-MNIST WITH DIFFERENT IMAGE ROTATION DEGREES

Method	0	15	30	45	60	75	Avg
ERM	95.9±0.1	006.86	98.8±0.0	98.9±0.0	98.9±0.0	96.4±0.0	98.0
IRM	95.5±0.1	98.8±0.2	98.7±0.1	98.6±0.1	98.7±0.0	95.9±0.2	97.7
GDRO	95.6±0.1	98.9±0.1	98.9±0.1	99.0±0.0	98.9±0.0	**96.5±0.2**	98.0
Mixup	95.8±0.3	00+6.86	98.9±0.0	98.9±0.0	98.8±0.1	**96.5±0.3**	98.0
MLDG	95.8±0.1	98.9±0.1	99.0±0.0	98.9±0.1	99.0±0.0	95.8±0.3	97.9
CORAL	95.8±0.3	98.8±0.0	98.9±0.0	99.0±0.0	98.9±0.1	96.4±0.2	98.0
MMD	95.6±0.1	98.9±0.1	99.0±0.0	99.0±0.0	98.9±0.0	96.0±0.2	97.9
DANN	95.0±0.5	98.9±0.1	99.0±0.0	99.0±0.1	98.9±0.0	96.3±0.2	97.8
CDANN	95.7±0.2	98.8±0.0	98.9±0.1	98.9±0.1	98.9±0.2	96.1±0.3	97.9
MTL	95.6±0.1	99.0±0.1	99.0±0.0	98.9±0.1	99.0±0.1	95.8±0.2	97.9
SagNets	95.9±0.3	98.9±0.1	99.0±0.1	**99.1±0.0**	99.0±0.1	96.3±0.1	98.0
ARM	**96.7±0.2**	**99.1±0.0**	99.0±0.0	99.0±0.1	99.1±0.1	**96.5±0.4**	**98.2**
VREx	95.9±0.2	99.0±0.1	98.9±0.1	98.9±0.1	98.7±0.1	96.2±0.2	97.9
RSC	94.8±0.5	98.7±0.1	98.8±0.1	98.8±0.0	98.9±0.1	95.9±0.2	97.9
mDSDI	96.0±0.1	98.8±0.1	99.1±0.0	98.9±0.0	99.2±0.1	96.2±0.1	98.0
CJLS	95.9	99.0	**99.2**	99.0	**99.3**	96.2	98.1

the results with and without SWAD optimization, explicitly labeled as CJLS+ with SWAD and marked with "*." Significantly, when juxtaposed with method eschewing SWAD, our approach emerges as a frontrunner, showcasing a substantial performance boost, especially when compared with the baseline method (mDSDI). CJLS notches up the ACC score by nearly 3%. Even in comparison to method using SWAD, our approach exhibits a more noteworthy impact. This can be attributed to the intrinsic complexity of the DomainNet dataset, which surpasses others by hundreds of times in size and boasts categories tens of times more extensive. This complexity accentuates the intricacy of the entire dataset, diminishing the efficacy of analogous methods and underscoring the need for heightened model optimization. In this intricate scenario, model integration algorithms, such as SWAD, become pivotal. Compared with SWAD, CJLS boosts the ACC score from 46.5% to 48.4%.

The methods applied to the "quick" and "info" domains in DomainNet show significantly less effectiveness, primarily due to their substantial stylistic differences from the rest of the domains. Despite these challenges, our method consistently outperforms most other approaches, as demonstrated by the comparison in Table VIII. Our approach, grounded in a continuous function akin to a Gaussian model, extends its coverage to more distant regions. In domains linked to remote areas, the decline in feature sampling leads to lower confidence and a weaker influence on the source domain compared with closer ones, so there are potential areas for improvement. Overall, our results demonstrate substantial improvement, underscoring the effectiveness of our foundational assumption of a continuous feature space, facilitating more comprehensive spatial coverage and more compact feature arrangement.

TABLE IV
COMPARISON OF ACC (%) WITH THE SOTA METHODS ON VLCS

	VLCS				
Algorithm	C	L	S	V	Avg
ERM	97.7±0.4	64.3±0.9	73.4±0.5	74.6±1.3	77.5
IRM	98.6±0.1	64.9±0.9	73.4±0.6	77.3±0.9	78.5
GDRO	97.3±0.3	63.4±0.9	69.5±0.8	76.7±0.7	76.7
Mixup	98.3±0.6	64.8±1.0	72.1±0.5	74.3±0.8	77.4
MLDG	97.4±0.2	65.2±0.7	71.0±1.4	75.3±1.0	77.2
CORAL	98.3±0.1	66.1±1.2	73.4±0.3	77.5±1.2	78.8
MMD	97.7±0.1	64.0±1.1	72.8±0.2	75.3±3.3	77.5
DANN	99.0±0.3	65.1±1.4	73.1±0.3	77.2±0.6	78.6
CDANN	97.1±0.3	65.1±1.2	70.7±0.8	77.1±1.5	77.5
MTL	97.8±0.4	64.3±0.3	71.5±0.7	75.3±1.7	77.2
SagNets	97.9±0.4	64.5±0.5	71.4±1.3	77.5±0.5	77.8
ARM	98.7±0.2	63.6±0.7	71.3±1.2	76.7±0.6	77.6
VREx	98.4±0.3	64.4±1.4	74.1±0.4	76.2±1.3	78.3
RSC	97.9±0.1	62.5±0.7	72.3±1.2	75.6±0.8	77.1
SWAD	**98.8±0.1**	63.3±0.3	**75.3±0.5**	79.2±0.6	79.1
mDSDI	97.6±0.1	66.4±0.4	74.0±0.6	77.8±0.7	79.0
iDAG	98.1	62.7	69.9	77.1	76.9
GMDG	98.3	65.9	73.4	**79.3**	79.2
CJLS	98.4	**68.1**	75.2	78.4	**80.0**

TABLE V
COMPARISON OF ACC (%) WITH THE SOTA METHODS ON PACS

Algorithm	A	C	P	S	Avg
ERM	84.7±0.4	80.8±0.6	97.2±0.3	79.3±1.0	85.5
IRM	84.8±1.3	76.4±1.1	96.7±0.6	76.1±1.0	83.5
GDRO	83.5±0.9	79.1±0.6	96.7±0.3	78.3±2.0	84.4
Mixup	86.1±0.5	78.9±0.8	97.6±0.1	75.8±1.8	84.6
MLDG	85.5±1.4	80.1±1.7	97.4±0.3	76.6±1.1	84.9
CORAL	88.3±0.2	80.0±0.5	97.5±0.3	78.8±1.3	86.2
MMD	86.1±1.4	79.4±0.9	96.6±0.2	76.5±0.5	84.6
DANN	86.4±0.8	77.4±0.8	97.3±0.4	73.5±2.3	83.6
CDANN	84.6±1.8	75.5±0.9	96.8±0.3	73.5±0.6	82.6
MTL	87.5±0.8	77.1±0.5	96.4±0.8	77.3±1.8	84.6
SagNets	87.4±1.0	80.7±0.6	97.1±0.1	80.0±0.4	86.3
ARM	86.8±0.6	76.8±0.5	97.4±0.3	79.3±1.2	85.1
VREx	86.0±1.6	79.1±0.6	96.9±0.5	77.7±1.7	84.9
RSC	85.4±0.8	79.7±1.8	97.6±0.3	78.2±1.2	85.2
SWAD	89.3±0.2	83.4±0.6	97.3±0.3	82.5±0.5	88.1
PCL	90.2	**83.9**	98.1	82.6	88.7
mDSDI	87.7±0.4	80.4±0.7	98.1±0.3	78.4±1.2	86.2
iDAG	**90.8**	83.7	98.0	**82.7**	**88.8**
GMDG	84.7	81.7	97.5	80.5	85.6
CJLS	90.5	82.8	**98.8**	82.6	88.7

TABLE VI
COMPARISON OF ACC (%) WITH THE SOTA METHODS ON OFFICE-HOME

Algorithm	A	C	P	R	Avg
ERM	61.3±0.7	52.4±0.3	75.8±0.1	76.6±0.3	66.5
IRM	58.9±2.3	52.2±1.6	72.1±2.9	74.0±2.5	64.3
GDRO	60.4±0.7	52.7±1.0	75.0±0.7	76.0±0.7	66.0
Mixup	62.4±0.8	54.8±0.6	76.9±0.3	78.3±0.2	68.1
MLDG	61.5±0.9	53.2±0.6	75.0±1.2	77.5±0.4	66.8
CORAL	65.3±0.4	54.4±0.5	76.5±0.1	78.4±0.5	68.7
MMD	60.4±0.2	53.3±0.3	74.3±0.1	77.4±0.6	66.3
DANN	59.9±1.3	53.0±0.3	73.6±0.7	76.9±0.5	65.9
CDANN	61.5±1.4	50.4±2.4	74.4±0.9	76.6±0.8	65.8
MTL	61.5±0.7	52.4±0.6	74.9±0.4	76.8±0.4	66.4
SagNets	63.4±0.2	54.8±0.4	75.8±0.4	78.3±0.3	68.1
ARM	58.9±0.8	51.0±0.5	74.1±0.1	75.2±0.3	64.8
VREx	60.7±0.9	53.0±0.9	75.3±0.1	76.6±0.5	66.4
RSC	60.7±1.4	51.4±0.3	74.8±1.1	75.1±1.3	65.5
SWAD	66.1±0.4	57.7±0.4	78.4±0.1	80.2±0.2	70.6
PCL	67.3	**59.9**	78.7	80.7	71.6
mDSDI	68.1±0.3	52.1±0.4	76.0±0.2	80.4±0.2	69.2
iDAG	68.2	57.9	79.7	81.4	71.8
GMDG	68.9	56.2	79.9	82.0	70.7
CJLS	**71.9**	58.4	**80.7**	**82.9**	**73.5**

TABLE VII
COMPARISON OF ACC (%) WITH THE SOTA METHODS ON TERRA

Algorithm	L100	L38	L43	L46	Avg
ERM	49.8±4.4	42.1±1.4	56.9±1.8	35.7±3.9	46.1
IRM	54.6±1.3	39.8±1.9	56.2±1.8	39.6±0.8	47.6
GDRO	41.2±0.7	38.6±2.1	56.7±0.9	36.4±2.1	43.2
Mixup	59.6±2.0	42.2±1.4	55.9±0.8	33.9±1.4	47.9
MLDG	54.2±3.0	44.3±1.1	55.6±0.3	36.9±2.2	47.7
CORAL	51.6±2.4	42.2±1.0	57.0±1.0	39.8±2.9	47.6
MMD	41.9±3.0	34.8±1.0	57.0±1.9	35.2±1.8	42.2
DANN	51.1±3.5	40.6±0.6	57.4±0.5	37.7±1.8	46.7
CDANN	47.0±1.9	41.3±4.8	54.9±1.7	39.8±2.3	45.8
MTL	49.3±1.2	39.6±6.3	55.6±1.1	37.8±0.8	45.6
SagNets	53.0±2.9	43.0±2.5	57.9±0.6	40.4±1.3	48.6
ARM	49.3±0.7	38.3±2.4	55.8±0.8	38.7±1.3	45.5
VREx	48.2±4.3	41.7±1.3	56.8±0.8	38.7±3.1	46.4
RSC	50.2±2.2	39.2±1.4	56.3±1.4	40.8±0.6	46.6
SWAD	55.4±0.0	44.9±1.1	59.7±0.4	39.9±0.2	50.0
PCL	58.7	**46.3**	**60**	43.6	**52.1**
mDSDI	53.2±3.0	43.3±1.0	56.7±0.5	39.2±1.3	48.1
iDAG	58.7	35.1	57.5	33.0	46.1
GMDG	**59.8**	45.3	57.1	38.2	50.1
CJLS	58.7	45.5	57.6	**46.2**	52.0

C. Ablation Study and Analysis

In this section, we analyze the function of each component separately, check the reliability of sampling in the Monte Carlo process, and examine the impact of the hyperparameters.

1) Ablation Study on Core Components: In this ablation study, we assess the contributions of different components, consisting of model modules and their corresponding loss functions. The CFDD module uses adversarial learning, linked with $\mathcal{L}_D$ for feature decomposition. The IR module and CJS are associated with both $\mathcal{L}_J$ and $\mathcal{L}_R$ and are rooted in VAE principles. The Multi-CLS module, connected to $\mathcal{L}_C$, contributes to the learning of semantic categories. Table IX presents the performance of each component in enhancing the model's generalization compared with the baseline. The baseline is developed from mDSDI without meta-learning, and we progressively introduce the proposed modules. The CFDD module is first introduced, revealing that a simple joint discriminator yields comparable results without the need for extensive covariance matrix computations. Next, the CJS is introduced for feature extraction, showcasing results comparable to CJSL. However, the addition of the IR module significantly improves results on the Photo (P) dataset in CJSL compared with CJS. The introduction of IR addresses the information loss during disentanglement, enhancing the model's generalization ability. However, when using only the CJL + IR part, the model's effectiveness diminishes notably. This suggests the challenge of directly modeling all the features in continuous space, where semantics and domain-style information may be intricately mixed. Our analysis indicates that introducing CFDD for feature decomposition allows for better modeling of domain-specific information. Expanding

TABLE VIII
COMPARISON OF ACC (%) WITH THE SOTA METHODS ON DOMAINNET

Algorithm	clip	info	paint	quick	real	sketch	Avg
ERM	58.1±0.3	18.8±0.3	46.7±0.3	12.2±0.4	59.6±0.1	49.8±0.4	40.9
IRM	48.5±2.8	15.0±1.5	38.3±4.3	10.9±0.5	48.2±5.2	42.3±3.1	33.9
GDRO	47.2±0.5	17.5±0.4	33.8±0.5	9.3±0.3	51.6±0.4	40.1±0.6	33.3
Mixup	55.7±0.3	18.5±0.5	44.3±0.5	12.5±0.4	55.8±0.3	48.2±0.5	39.2
MLDG	59.1±0.2	19.1±0.3	45.8±0.7	13.4±0.3	59.6±0.2	50.2±0.4	41.2
CORAL	59.2±0.1	19.7±0.2	46.6±0.3	13.4±0.4	59.8±0.2	50.1±0.6	41.5
MMD	32.1±13.3	11.0±4.6	26.8±11.3	8.7±2.1	32.7±13.8	28.9±11.9	23.4
DANN	53.1±0.2	18.3±0.1	44.2±0.7	11.8±0.1	55.5±0.4	46.8±0.6	38.3
CDANN	54.6±0.4	17.3±0.1	43.7±0.9	12.1±0.7	56.2±0.4	45.9±0.5	38.3
MTL	57.9±0.5	18.5±0.4	46.0±0.1	12.5±0.1	59.5±0.3	49.2±0.1	40.6
SagNets	57.7±0.3	19.0±0.2	45.3±0.3	12.7±0.5	58.1±0.5	48.8±0.2	40.3
ARM	49.7±0.3	16.3±0.5	40.9±1.1	9.4±0.1	53.4±0.4	43.5±0.4	35.5
VREx	47.3±3.5	16.0±1.5	35.8±4.6	10.9±0.3	49.6±4.9	42.0±3.0	33.6
RSC	55.0±1.2	18.3±0.5	44.4±0.6	12.2±0.2	55.7±0.7	47.8±0.9	38.9
mDSDI	62.1±0.3	19.1±0.4	49.4±0.4	12.8±0.7	62.9±0.3	50.4±0.4	42.8
GMDG	63.4	22.4	51.4	13.4	64.4	52.4	44.6
CJLS	63.2	22.5	52.0	14.0	66.6	55.2	45.6
SWAD	66.0±0.1	22.4±0.3	53.5±0.1	16.1±0.2	65.8±0.4	55.5±0.3	46.5
PCL	**67.9**	24.3	55.3	15.7	66.6	56.4	47.7
iDAG	**67.9**	24.2	55.0	**16.4**	66.1	56.9	47.7
CJLS+*	66.3	**25.8**	**57.0**	15.8	**68.1**	**57.3**	**48.4**

* CJLS+ denotes the improved version of CJLS by integrating the SWAD strategy. Note that PCL also integrates the SWAD strategy.

TABLE IX
COMPARISON OF ACC (%) FOR COMPONENTS ON PACS

Method	A	C	P	S	Avg
Baseline	87.3	78.2	96.9	77.8	85.1
Baseline w/ CFDD	86.9	79.5	97.1	78.2	85.4
Baseline w/ CJS + IR	81.6	51.6	85.5	66.2	71.2
Baseline w/ CFDD + CJS	89.8	82.3	97.9	81.9	88.0
CJSL	**90.5**	**82.8**	**98.8**	**82.6**	**88.7**

TABLE X
COMPARISON OF ACC (%) FOR COMPONENTS ON VLCS

Method	A	C	P	S	Avg
baseline	95.4	64.2	73.8	72.8	76.6
baseline w/ CFDD	96.6	67.7	74.1	74.9	78.3
baseline w/ CJS + IR	77.8	60.7	57.9	47.6	61.0
baseline w/ CFDD + CJS	97.6	68.0	**75.8**	75.5	79.2
Our method	**98.4**	**68.1**	75.2	**78.4**	**80.0**

our experimental validation to multiple datasets, we include results on the VLCS dataset in Table X. While PACS centers on distinct styles from each domain, VLCS concentrates on diverse acquisition sources within each domain. The inclusion of results from these two representative datasets enhances the comprehensiveness of our method's evaluation. The consistency of results across these datasets reinforces our method's effectiveness and robustness.

2) Ablation Study on Sampling: We use random sampling in the Monte Carlo procedure rather than sampling based on $\hat{z}$-estimation in our method. As shown in Table XI, if the z_S feature is not introduced, the result will drop slightly. And the sampling with $\hat{z}$ can perform better than the feature vectors that the direct model learns. $\hat{z}$ denotes the feature directly obtained by the specific feature extractor, then $\hat{z}$-estimation is to use $\hat{z}$ to estimate the distribution of the target domain to construct the

TABLE XI
COMPARISON OF ACC (%) FOR SAMPLING ON PACS

Method	A	C	P	S	Avg
Baseline	87.3	78.2	96.9	77.8	85.1
Baseline w/o z_S	87.1	77.8	97.1	77.6	84.9
Sampling w/ $\hat{z}$	89.8	82.6	98.2	81.9	88.1
Multi Random sampling	**90.5**	82.8	**98.8**	**82.6**	**88.7**
Random sampling (M=1)	89.7	**83.2**	98.1	76.4	86.9
Random sampling (M=2)	90.4	83.1	98.0	77.8	87.3
Random sampling (M=4)	89.8	82.1	98.3	82.1	88.1

required specific feature. It can further demonstrate that this continuous space-sampling-based method does indeed produce more accurate specific feature vectors. However, it does not outperform the random sampling method. This intriguing and counterintuitive phenomenon reflects the main tenet of our strategy. The potential reasons are twofold. First, estimating the target domain distribution based on $\hat{z}$ may be unreliable and biased. As a result, this distribution cannot be used to accurately sample an accurate specific feature vector. Second, assuming that the target domain distribution can also be accurately obtained, the sampling feature vectors will be highly clustered in one region, which is also biased for multiple sampling. In addition, not all the points in our space have the same density since it is made up of a variety of different domain distributions. Thus, when we map the target domain to the source domain classification space, this aggregation may result in inadequate confidence. Consequently, our theory is supported by the blend of the two types of factors mentioned above and experimental results. Furthermore, we present the outcomes of different sampling numbers, uncovering distinct strengths and weaknesses in each domain for individual samplings. Surprisingly, some domains yield superior results with less sampling compared with multiple samplings. But in general, the more the sampling times, the better the effect. It reveals that each sampling exhibits its own strengths and weaknesses within each domain, contributing to some overall variability. Consequently, we emphasize the importance of bolstering model stability and robustness through an averaging strategy across multiple samples.

3) Ablation Study on the Impact of Loss Functions: We evaluate the impact of the proposed loss functions. First, we concentrate on the entropy loss L_u in the Monte Carlo process. The comparison results are reported in Table XII, from which we can see that CJSL improves the score of the version without L_u from 88.3% to 88.7%, indicating the importance of introducing L_u. As shown in Table XIII, we compare the relationship of loss while exploiting the hyperparameters α, β, and γ. From the results, we can see that our method can achieve high ACC scores under different values for α, β, and γ, indicating that the performance of CJSL is not sensitive to hyperparameters.

4) Impact of the Number of Domains: Table XIV shows comparison of ACC for different numbers of domains on the Rotated 0 MNIST dataset. The results demonstrate the performance of the method under consideration as the number of domains varies. This suggests that the method maintains

TABLE XII
COMPARISON OF ACC (%) ABOUT IMPACT OF L_u ON PACS

Method	A	C	P	S	Avg
CJSL w/o L_u	90.1	81.9	**98.8**	82.3	88.3
CJSL	**90.5**	**82.8**	98.8	**82.6**	**88.7**

TABLE XIII
COMPARISON OF ACC (%) FOR HYPERPARAMETERS ON PACS

α	β	γ	A	C	P	S	Avg
0.1	1	1	89.3	78.9	98.3	79.1	86.4
1	0.1	1	88.7	82.7	98.0	80.4	87.4
1	1	0.1	87.0	79.8	98.2	83.1	87.0
1	1	1	88.4	81.4	98.4	80.5	87.2
5	1	1	90.5	82.8	98.8	82.6	**88.7**

TABLE XIV
COMPARISON OF ACC (%) FOR DIFFERENT NUMBER OF DOMAINS ON ROTATED 0 MNIST

# Domains	2	3	4	5
ACC	95.50	95.53	95.64	95.87

high ACC across different numbers of domains, with a gradual performance increase as the involving more domains grows. From the perspective of different datasets, Table V shows the results on PACS, while Table VIII shows the results on domainNet. It is clear to see that as the number of domains increases, our method has a significant improvement over the percentage of effect improvement. To a certain extent, it can be shown that our method is scalable to the increase in the number of domains.

5) Compute-Efficiency Analysis: Table XV provides a comparison of the efficiency of various methods applied to the PACS dataset. The methods evaluated include the baseline model, the baseline model enhanced with the CFDD, the baseline model supplemented with both CFDD and CJS, and our proposed continuous joint space learning method. The efficiency metrics assessed are the training time cost (in seconds) for each iteration, the inference time for each image, and the GPU memory usage (in MiB) for training with a batch size of 16 images. The results show that CJSL incurs only a slight overhead compared with the baseline in terms of the inference time cost, increasing from 0.534 to 0.551 s per image, and the memory usage for training, increasing from 18 132 to 20 942 MiB. Note that all the images are resized to 224 × 224 pixels in our experiment. The main difference in resource requirements between methods is the training time, as different methods converge after varying numbers of iterations on different datasets. The results for the other two metrics will remain consistent across different datasets if the image size and batch size are kept the same. These results indicate that our CJSL method improves performance without significantly increasing resource demands, thus maintaining practicality and avoiding excessive complexity.

TABLE XV
EFFICIENCY COMPARISON: TRAINING TIME COST PER ITERATION (TTC), INFERENCE TIME COST PER IMAGE (ITC), AND MEMORY USAGE (MEM.) FOR TRAINING ON PACS

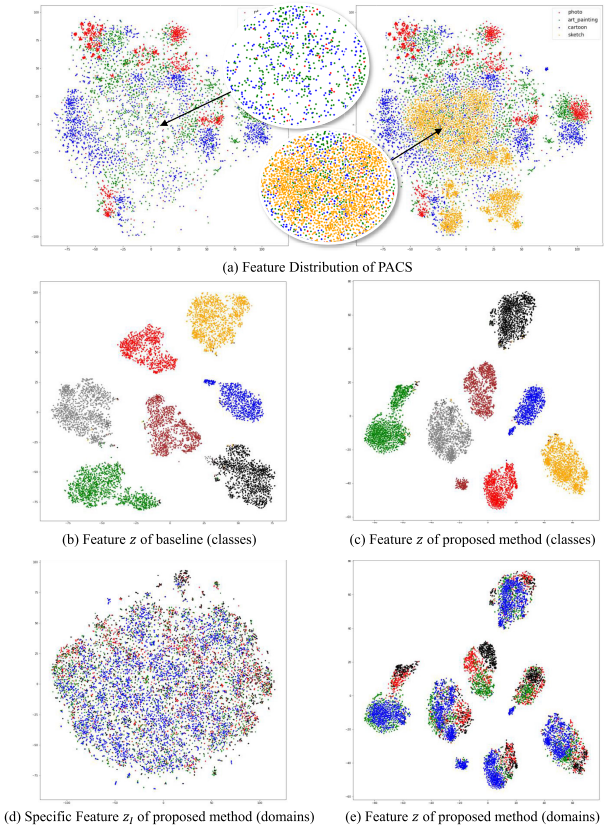
Method	TTC (sec)	ITC (sec)	Mem. (MiB)
mDSDI	2.431	0.541	18,650
Baseline	2.420	0.534	18,132
Baseline w/ CFDD	2.414	0.533	19,692
Baseline w/ CFDD + CJS	2.438	0.542	20,014
CJSL	**2.453**	**0.551**	**20,942**

D. Visualization and Discussion

We use t-distributed stochastic neighbor embedding (t-SNE) [69] to visualize the learned feature representations and analyze the feature space to further understand the effectiveness of our method. Fig. 5 illustrates the results on PACS. The visual representation in Fig. 5(a) illustrates the challenges associated with extracting features directly from diverse domains using ResNet-50. The uncovered areas signify regions within the acquired feature subspace that were part of the source domain training but were inadequately learned by the model. In addition, the overall distribution lacks distinct semantically relevant aggregation clusters, indicating that the features are decomposed before effectively expressing semantic information. From Fig. 5(c), we can see that the embedding results of feature representations obtained by our CJLS are approximately grouped into different clusters according to their semantics. While some of the clusters in Fig. 5(b) obtained by mDSDI cannot be well-distinguished but also far exceed the effect of Fig. 5(b). In other words, in Fig. 5(c), each category aggregation region is more compact, which improves the classification performance since it is easier to differentiate between different categories. Furthermore, Fig. 5(d) shows the embedding of specific feature vectors obtained by CJLS. We can see that the samples from different domains are mixed together. The specific feature vectors all land in the GMM and the feature vectors from different domains tend to be in different positions. For instance, the domains depicted in black have a bias to the upper right, whereas the domains depicted in blue have a bias to the lower left. The concatenation of these domain-specific feature vectors and the domain-invariant feature vectors is distributed in several small clusters, as shown in Fig. 5(e). In each cluster, samples from all the domains are included. It indicates that all the sampled domain-specific feature vectors will be within the GMM instead of landing in some uncover area as mentioned in Fig. 1. Within the cluster, it should be able to exhibit some discriminatory ability between domains. The differentiability between blocks should be very high for categories, resulting in an overall distribution similar to Fig. 5(b). This aligns with our foundational assumption of using a continuous space to represent domain-specific features. The continuity of the distribution ensures the absence of uncovered areas within the feature space.

V. DISCUSSION

Although our method surpasses the current SOTA methods on most benchmarks, it has two limitations that must be taken into account. First, the training process takes longer due



(a) Feature Distribution of PACS

(b) Feature z of baseline (classes)

(c) Feature z of proposed method (classes)

(d) Specific Feature z_l of proposed method (domains)

(e) Feature z of proposed method (domains)

Fig. 5. Visualization of embedding of high-level feature vectors z using the t-SNE algorithm [69]. (a) Visualization of features extracted from diverse domains using a ResNet-50. In (b) and (c), different colors denote samples from different classes. In (d) and (e), different colors denote samples from different domains.

to the intricacy of constructing a CJS from various source domains and training Monte Carlo dropout-based classifiers. This complexity results in increased iteration consumption. Second, our method relies more on larger amounts of data. The amount of data and the distance between domains significantly affect the construction of the continuous space. Therefore, as the gap between domains widens, more data are required. Finally, we have not addressed the challenge of model updates upon encountering a new domain. We plan to incorporate principles of continual learning. In this context, our approach necessitates further refinement with additional constraints to mitigate the risk of catastrophic forgetting. The introduction of new domain data may lead to memory loss in the model, hindering accurate embedding of the new domain into the existing latent space. Furthermore, as our CFDD adapts and the number of discriminated domains increases, retraining the fully connected layer becomes essential. In future, we will pay our efforts to tackle these key challenges

VI. Conclusion

In this article, we presented a novel method for learning a continuous disentangled joint space for domain-specific representations in feature disentanglement, with the aim of improving DG. The proposed method formulates the CJS explicitly and samples feature vectors over its distribution instead of learning directly through the model. To stabilize predictions, a Monte Carlo sampling-based Bayesian strategy is developed, which incorporates findings from multiple source domains and numerous samplings. Empirical results verified the relationship between domain specificity and stylization and demonstrated the effectiveness of the proposed method, which outperforms 19 peer methods on seven popular public benchmarks. Future work will involve exploring DG in distributed learning systems, with the goal of leveraging the power of multiple source domains while preserving participant privacy.

References

[1] Y. Wang et al., "Adversarial multimodal fusion with attention mechanism for skin lesion classification using clinical and dermoscopic images," *Med. Image Anal.*, vol. 81, Oct. 2022, Art. no. 102535.

[2] J. Zhang, L. Qi, Y. Shi, and Y. Gao, "MVDG: A unified multi-view framework for domain generalization," 2021, *arXiv:2112.12329*.

[3] J. Li, S. Shang, and L. Chen, "Domain generalization for named entity boundary detection via metalearning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 9, pp. 3819–3830, Sep. 2021.

[4] J. Kang, S. Lee, N. Kim, and S. Kwak, "Style neophile: Constantly seeking novel styles for domain generalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7120–7130.

[5] Z. Wang, X. Shu, Y. Wang, Y. Feng, L. Zhang, and Z. Yi, "A feature space-restricted attention attack on medical deep learning systems," *IEEE Trans. Cybern.*, vol. 53, no. 8, pp. 5323–5335, Sep. 2023.

[6] J. Lin, Y. Tang, J. Wang, and W. Zhang, "Constrained maximum cross-domain likelihood for domain generalization," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 13, 2024, doi: 10.1109/TNNLS.2023.3292242.

[7] C. Biffi et al., "Explainable anatomical shape analysis through deep hierarchical generative models," *IEEE Trans. Med. Imag.*, vol. 39, no. 6, pp. 2088–2099, Jun. 2020.

[8] 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. (ICCV)*, Oct. 2019, pp. 2100–2110.

[9] Y. Feng et al., "Contrastive domain adaptation with consistency match for automated pneumonia diagnosis," *Med. Image Anal.*, vol. 83, Jan. 2023, Art. no. 102664.

[10] J. Zhang, L. Qi, Y. Shi, and Y. Gao, "Generalizable model-agnostic semantic segmentation via target-specific normalization," *Pattern Recognit.*, vol. 122, Feb. 2022, Art. no. 108292.

[11] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4396–4415, Apr. 2023.

[12] X. Zhang et al., "Towards domain generalization in object detection," 2022, *arXiv:2203.14387*.

[13] W. Lu, X. Jia, W. Xie, L. Shen, Y. Zhou, and J. Duan, "Geometry constrained weakly supervised object localization," in *Proc. Eur. Conf. Comput. Vis.*, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., Cham, Switzerland: Springer, 2020, pp. 481–496.

[14] D. Li, J. Yang, K. Kreis, A. Torralba, and S. Fidler, "Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8300–8311.

[15] S. Lee, H. Seong, S. Lee, and E. Kim, "WildNet: Learning domain generalized semantic segmentation from the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9926–9936.

[16] Z. Wang, X. Shu, C. Chen, Y. Teng, L. Zhang, and J. Xu, "A semi-symmetric domain adaptation network based on multi-level adversarial features for meningioma segmentation," *Knowl.-Based Syst.*, vol. 228, Sep. 2021, Art. no. 107245.

[17] C. Ouyang, K. Kamnitsas, C. Biffi, J. Duan, and D. Rueckert, "Data efficient unsupervised domain adaptation for cross-modality image segmentation," in *Proc. Int. Conf. Med. Image Comput. Assist. Intervent.*, vol. 11765, D. Shen et al., Eds., Cham, Switzerland: Springer, 2019, pp. 669–677.

[18] P. Chattopadhyay, Y. Balaji, and J. Hoffman, "Learning to balance specificity and invariance for in and out of domain generalization," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 301–318.

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

[20] F. D. Johansson, D. A. Sontag, and R. Ranganath, "Support and invertibility in domain-invariant representations," in *Proc. Int. Conf. Artif. Intell. Statist.*, vol. 89, 2019, pp. 527–536.

[21] M. Bui, T. Tran, A. Tran, and D. Q. Phung, "Exploiting domain-specific features to enhance domain generalization," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021, pp. 21189–21201.

[22] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, Y. Bengio and Y. LeCun, Eds., 2014, pp. 1–14.

[23] L. Li et al., "Progressive domain expansion network for single domain generalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 224–233.

[24] Z. Xu, D. Liu, J. Yang, C. Raffel, and M. Niethammer, "Robust and generalizable visual representation learning via random convolutions," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–19.

[25] R. Gong, W. Li, Y. Chen, and L. Van Gool, "DLOW: Domain flow for adaptation and generalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2477–2486.

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

[27] S. Jeon, K. Hong, P. Lee, J. Lee, and H. Byun, "Feature stylization and domain-aware contrastive learning for domain generalization," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 22–31.

[28] O. Nuriel, S. Benaim, and L. Wolf, "Permuted AdaIN: Reducing the bias towards global statistics in image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9477–9486.

[29] J. Wang, C. Lan, C. Liu, Y. Ouyang, and T. Qin, "Generalizing to unseen domains: A survey on domain generalization," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 4627–4635.

[30] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. Wilson, "Averaging weights leads to wider optima and better generalization," in *Proc. 34th Conf. Uncertainty Artif. Intell.*, vol. 2, 2018, pp. 876–885.

[31] H. Wang, Z. He, Z. C. Lipton, and E. P. Xing, "Learning robust representations by projecting superficial statistics out," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–16.

[32] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain adaptive ensemble learning," *IEEE Trans. Image Process.*, vol. 30, pp. 8008–8018, 2021.

[33] J. Cha et al., "SWAD: Domain generalization by seeking flat minima," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021, pp. 22405–22418.

[34] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi, "Domain generalization by solving jigsaw puzzles," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2229–2238.

[35] V. Piratla, P. Netrapalli, and S. Sarawagi, "Efficient domain generalization via common-specific low-rank decomposition," in *Proc. Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 7728–7738.

[36] 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.

[37] H. Nam, H. Lee, J. Park, W. Yoon, and D. Yoo, "Reducing domain gap by reducing style bias," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8690–8699.

[38] Y. Li et al., "Deep domain generalization via conditional invariant adversarial networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 647–663.

[39] M. M. Rahman, C. Fookes, M. Baktashmotlagh, and S. Sridharan, "Correlation-aware adversarial domain adaptation and generalization," *Pattern Recognit.*, vol. 100, Apr. 2020, Art. no. 107124.

[40] L. Zhang, Z. Liu, W. Zhang, and D. Zhang, "Style uncertainty based self-paced meta learning for generalizable person re-identification," *IEEE Trans. Image Process.*, vol. 32, pp. 2107–2119, 2023.

[41] M. Xu, L. Qin, W. Chen, S. Pu, and L. Zhang, "Multi-view adversarial discriminator: Mine the non-causal factors for object detection in unseen domains," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 8103–8112.

[42] R. Meng et al., "Attention diversification for domain generalization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2022, pp. 322–340.

[43] M. Kim and H. Byun, "Learning texture invariant representation for domain adaptation of semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12972–12981.

[44] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *32nd Int. Conf. Mach. Learn. (ICML)*, vol. 2, Jul. 2015, pp. 1180–1189.

[45] K. Ahuja, K. Shanmugam, K. R. Varshney, and A. Dhurandhar, "Invariant risk minimization games," in *Proc. Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 145–155.

[46] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2551–2559.

[47] 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.*, Dec. 2013, pp. 1657–1664.

[48] D. Li, Y. Yang, Y. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5543–5551.

[49] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5385–5394.

[50] S. Beery, G. V. Horn, and P. Perona, "Recognition in Terra incognita," in *Proc. Eur. Conf. Comput. Vis.*, vol. 11220, 2018, pp. 472–489.

[51] 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. (ICCV)*, Oct. 2019, pp. 1406–1415.

[52] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–29.

[53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[55] X. Yao et al., "PCL: Proxy-based contrastive learning for domain generalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7087–7097.

[56] V. Vapnik, *Statistical Learning Theory*. Hoboken, NJ, USA: Wiley, 1998.

[57] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, "Distributionally robust neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–19.

[58] G. Blanchard, A. A. Deshmukh, Ü. Dogan, G. Lee, and C. Scott, "Domain generalization by marginal transfer learning," *J. Mach. Learn. Res.*, vol. 22, pp. 2:1–2:55, Jan. 2021.

[59] M. Zhang, H. Marklund, N. Dhawan, A. Gupta, S. Levine, and C. Finn, "Adaptive risk minimization: Learning to adapt to domain shift," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 23664–23678.

[60] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3490–3497.

[61] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 443–450.

[62] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, pp. 59:1–59:35, May 2016.

[63] Y. Li, M. Gong, X. Tian, T. Liu, and D. Tao, "Domain generalization via conditional invariant representations," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 3579–3587.

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

[65] Y. Wang, H. Li, and A. C. Kot, "Heterogeneous domain generalization via domain mixup," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 3622–3626.

[66] Z. Huang, H. Wang, E. P. Xing, and D. Huang, "Self-challenging improves cross-domain generalization," in *Proc. Eur. Conf. Comput. Vis.*, vol. 12347, 2020, pp. 124–140.

[67] Z. Huang, H. Wang, J. Zhao, and N. Zheng, "IDAG: Invariant DAG searching for domain generalization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 19112–19122.

[68] Z. Tan, X. Yang, and K. Huang, "Rethinking multi-domain generalization with a general learning objective," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2024, pp. 23512–23522.

[69] V. der Maaten, Laurens, and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.