VARIATIONAL INVARIANT LEARNING FOR BAYESIAN DOMAIN GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain generalization addresses the out-of-distribution problem, which is challenging due to the domain shift and the uncertainty caused by the inaccessibility to data from the target domains. In this paper, we propose *variational invariant learning*, a probabilistic inference framework that jointly models domain invariance and uncertainty. We introduce variational Bayesian approximation into both the feature representation and classifier layers to facilitate invariant learning for better generalization across domains. In the probabilistic modeling framework, we introduce a domain-invariant principle to explore invariance across domains in a unified way. We incorporate the principle into the variational Bayesian layers in neural networks, achieving domain-invariant representations and classifier. We empirically demonstrate the effectiveness of our proposal on four widely used cross-domain visual recognition benchmarks. Ablation studies demonstrate the benefits of our proposal and on all benchmarks our variational invariant learning consistently delivers state-of-the-art performance.

1 INTRODUCTION

Domain generalization (Muandet et al., 2013), as an out-of-distribution problem, aims to train a model on several source domains and have it generalize well to unseen target domains. The major challenge stems from the large distribution shift between the source and target domains, which is further complicated by the prediction uncertainty (Malinin & Gales, 2018) introduced by the inaccessibility to data from target domains during training. Previous approaches focus on learning domain-invariant features using novel loss functions (Muandet et al., 2013; Li et al., 2018a) or specific architectures (Li et al., 2017a; D'Innocente & Caputo, 2018). Meta-learning based methods were proposed to achieve similar goals by leveraging an episodic training strategy (Li et al., 2017b; Balaji et al., 2018; Du et al., 2020). Most of these methods are based on deep neural network backbones (Krizhevsky et al., 2012; He et al., 2016). However, while deep neural networks have achieved remarkable success in various vision tasks, their performance is known to degenerate considerably when the test samples are out of the training data distribution (Nguyen et al., 2015; Ilse et al., 2019), due to their poorly calibrated behavior (Guo et al., 2017; Kristiadi et al., 2020).

As an attractive solution, Bayesian learning naturally represents prediction uncertainty (Kristiadi et al., 2020; MacKay, 1992), possesses better generalizability to out-of-distribution examples (Louizos & Welling, 2017) and provides an elegant formulation to transfer knowledge across different datasets (Nguyen et al., 2018). Further, approximate Bayesian inference has been demonstrated to be able to improve prediction uncertainty (Blundell et al., 2015; Louizos & Welling, 2017; Atanov et al., 2019), even when only applied to the last network layer (Kristiadi et al., 2020). These properties make it appealing to introduce Bayesian learning into the challenging and unexplored scenario of domain generalization.

In this paper, we propose *variational invariant learning* (VIL), a Bayesian inference framework that jointly models domain invariance and uncertainty for domain generalization. We apply variational Bayesian approximation to the last two network layers for both the representations and classifier by placing prior distributions over their weights, which facilitates generalization. We adopt Bayesian neural networks to domain generalization, which enjoys the representational power of deep neural networks while facilitating better generalization. To further improve the robustness to domain shifts, we introduce the domain-invariant principle under the Bayesian inference framework, which enables us to explore domain invariance for both feature representations and the classifier in a unified way. We evaluate our method on four widely-used benchmarks for cross-domain visual object classification. Our ablation studies demonstrate the effectiveness of the variational Bayesian domain-invariant features and classifier for domain generalization. Results further show that our method achieves the best performance on all of the four benchmarks.

2 Methodology

We explore Bayesian inference for domain generalization. In this task, the samples from the target domains are never seen during training, and are usually out of the data distribution of the source domains. This leads to uncertainty when making predictions on the target domains. Bayesian inference offers a principled way to represent the predictive uncertainty in neural networks (MacKay, 1992; Kristiadi et al., 2020). We briefly introduce approximate Bayesian inference, under which we will introduce our variational invariant learning for domain generalization.

2.1 Approximate Bayesian inference

Given a dataset $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$ of N input-output pairs and a model parameterized by weights $\boldsymbol{\theta}$ with a prior distribution $p(\boldsymbol{\theta})$, Bayesian neural networks aim to infer the true posterior distribution $p(\boldsymbol{\theta}|\mathbf{x}, \mathbf{y})$. As the exact inference of the true posterior is computationally intractable, Hinton & Camp (1993) and Graves (2011) recommended learning a variational distribution $q(\boldsymbol{\theta})$ to approximate $p(\boldsymbol{\theta}|\mathbf{x}, \mathbf{y})$ by minimizing the Kullback-Leibler (KL) divergence between them:

$$\theta^{*} = \arg\min_{\theta} \mathbb{D}_{\mathrm{KL}}\big[q(\boldsymbol{\theta}) || p(\boldsymbol{\theta}|\mathbf{x}, \mathbf{y})\big]. \tag{1}$$

The above optimization is equivalent to minimizing the loss function:

$$\mathcal{L}_{\mathrm{Bayes}} = -\mathbb{E}_{q(\boldsymbol{\theta})}[\log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})] + \mathbb{D}_{\mathrm{KL}}[q(\boldsymbol{\theta}) || p(\boldsymbol{\theta})], \tag{2}$$

which is also known as the negative value of the evidence lower bound (ELBO) (Blei et al., 2017).

2.2 Variational domain-invariant learning

In domain generalization, let $\mathcal{D} = \{D_i\}_{i=1}^{|\mathcal{D}|} = \mathcal{S} \cup \mathcal{T}$ be a set of domains, where $\mathcal{S}$ and $\mathcal{T}$ denote source domains and target domains respectively. $\mathcal{S}$ and $\mathcal{T}$ do not have any overlap with each other but share the same label space. For each domain $D_i \in \mathcal{D}$, we can define a joint distribution $p(\mathbf{x}_i, \mathbf{y}_i)$ in the input space $\mathcal{X}$ and the output space $\mathcal{Y}$. We aim to learn a model $f : \mathcal{X} \to \mathcal{Y}$ in the source domains $\mathcal{S}$ that can generalize well to the target domains $\mathcal{T}$.

The fundamental problem in domain generalization is to achieve robustness to domain shift between source and target domains, that is, we aim to learn a model invariant to the distributional shift between the source and target domains. In this work, we mainly focus on the invariant property across domains instead of exploring general invariance properties (Nalisnick & Smyth, 2018). Therefore, we introduce a formal definition of domain invariance, which is easily incorporated as criteria into the Bayesian framework to achieve domain-invariant learning.

Provided that all domains in $\mathcal{D}$ are in the same domain space, then for any input sample $\mathbf{x}_s$ in domain D_s, we assume that there exists a domain-transform function $g_\zeta(\cdot)$ which is defined as a mapping function that is able to project $\mathbf{x}_s$ to other different domains $\mathcal{D}_\zeta$ with respect to the parameter ζ, where $\zeta \sim q(\zeta)$, and a different ζ lead to different post-transformation domains D_ζ. Usually the exact form of $g_\zeta(\cdot)$ is not necessarily known. Under this assumption, we introduce the definition of domain invariance, which we will incorporate into the Bayesian layers of neural networks for domain-invariant learning.

Definition 2.1 (Domain Invariance) *Let $\mathbf{x}_s$ be a given sample from domain $D_s \in \mathcal{D}$, and $\mathbf{x}_\zeta = g_\zeta(\mathbf{x}_s)$ be a transformation of $\mathbf{x}_s$ in another domain D_ζ, where $\zeta \sim q(\zeta)$. $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ denotes the output distribution of input $\mathbf{x}$ with model $\boldsymbol{\theta}$. The model $\boldsymbol{\theta}$ is domain-invariant if,*

$$p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s) = p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta), \qquad \forall \zeta \sim q(\zeta). \tag{3}$$

Here, we use $\mathbf{y}$ to represent the output from a neural layer with input $\mathbf{x}$, which can either be the prediction vector from the last layer or the feature vector from the convolutional layers.

To make the domain-invariant principle easier to implement, we then extend the Eq. (3) to an expectation form:

$$p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s) = \mathbb{E}_{q_\zeta}[p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)]. \tag{4}$$

Based on this definition, we use the Kullback-Leibler divergence between the two terms in Eq. (4), $\mathbb{D}_{\mathrm{KL}}\left[p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s)||\mathbb{E}_{q_\zeta}[p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)]\right]$, to quantify the domain invariance of the model, which will be zero when the model is domain invariant. As in most cases, there is no analytical form of the domain-transform function and only a few samples from D_ζ are available, which makes $\mathbb{E}_{q_\zeta}[p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)]$ intractable. Thus, we derive the following upper bound of the divergence:

$$\mathbb{D}_{\mathrm{KL}}\left[p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s)||\mathbb{E}_{q_\zeta}[p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)]\right] \leq \mathbb{E}_{q_\zeta}\Big[\mathbb{D}_{\mathrm{KL}}\left[p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s)||p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)\right]\Big], \tag{5}$$

which can be approximated by Monte Carlo sampling.

We define the complete objective function of our variational invariant learning by combining Eq. (5) with Eq. (2). However, in Bayesian inference, the likelihood is obtained by taking the expectation over the distribution of parameter $\boldsymbol{\theta}$, i.e., $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) = \mathbb{E}_{q(\boldsymbol{\theta})}[p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})]$, which is also intractable in Eq. (5). As the KL divergence is a convex function (Nalisnick & Smyth, 2018), we further extend Eq. (5) to an upper bound:

$$\mathbb{E}_{q_\zeta}\Big[\mathbb{D}_{\mathrm{KL}}\left[p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s)||p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)\right]\Big] \leq \mathbb{E}_{q_\zeta}\Big[\Big[\mathbb{E}_{q(\boldsymbol{\theta})}\mathbb{D}_{\mathrm{KL}}\left[p(\mathbf{y}_s|\mathbf{x}_s, \boldsymbol{\theta})||p(\mathbf{y}_\zeta|\mathbf{x}_\zeta, \boldsymbol{\theta})\right]\Big]\Big], \tag{6}$$

which is tractable with the unbiased Monte Carlo approximation. The complete derivations of Eq. (5) and Eq. (6) are provided in Appendix A.

In addition, it is worth noting that the domain-transformation distribution $q(\zeta)$ is implicit and inexpressible in reality and there are only a limited number of domains available in practice. This problem is exacerbated because the target domain is unseen during training, which further limits the number of available domains. Moreover, in most of the domain generalization databases, for a certain sample $\mathbf{x}_s$ from domain D_s, there is no transformation corresponding to $\mathbf{x}_s$ in other domains. This prevents the expectation with respect to q_ζ from being directly tractable in general.

Thus, we resort to use an empirically tractable implementation and adopt an episodic setting as in (Li et al., 2019). In each episode, we choose one domain from the source domains $\mathcal{S}$ as the meta-source domain D_s and the rest are used as the meta-target domains $\{D_t\}_{t=1}^T$. To achieve variational invariant learning in the Bayesian framework, we use samples from meta-target domains in the same category as $\mathbf{x}_s$ to approximate the samples of $g_\zeta(\mathbf{x}_s)$. Then we obtain a general loss function for domain-invariant learning:

$$\mathcal{L}_{\mathrm{I}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{q(\boldsymbol{\theta})}\Big[\mathbb{D}_{\mathrm{KL}}\left[p(\mathbf{y}_s|\mathbf{x}_s, \boldsymbol{\theta})||p(\mathbf{y}_t^i|\mathbf{x}_t^i, \boldsymbol{\theta})\right]\Big], \tag{7}$$

where $\{\mathbf{x}_t^i\}_{i=1}^N$ are from D_t, denoting the samples in the same category as $\mathbf{x}_s$. More details and an illustration of the domain-invariant loss function can be found in Appendix B.

With the aforementioned loss functions, we develop the loss function of variational invariant learning for domain generalization:

$$\mathcal{L}_{\mathrm{VIL}} = \mathcal{L}_{\mathrm{Bayes}} + \lambda\mathcal{L}_{\mathrm{I}}. \tag{8}$$

Our variational invariant learning combines the Bayesian framework, which is able to introduce uncertainty into the network and is beneficial for out-of-distribution problems (Daxberger & Hernández-Lobato, 2019), and a domain-invariant loss function $\mathcal{L}_{\mathrm{I}}$, which is designed based on predictive distributions to make the model generalize better to the unseen target domains. For Bayesian learning, it has been demonstrated that being just "a bit" Bayesian in the last layer of the neural network can well represent the uncertainty in predictions (Kristiadi et al., 2020). This indicates that applying the Bayesian treatment only to the last layer already brings sufficient benefits of Bayesian inference. Although adding Bayesian inference to more layers improves the performance, it also increases the computational cost. Further, from the perspective of domain invariance, making both the classifier and feature extractor more robust to the domain shifts also leads to better performance (Li et al., 2019). Thus, there is a trade-off between the benefits of variational Bayesian domain invariance and computational efficiency. Instead of applying the Bayesian principle to all the layers of the neural network, in this work we explore domain invariance by applying it to only the classifier layer $\boldsymbol{\psi}$ and the last feature extraction layer $\boldsymbol{\phi}$.

In this case, the $\mathcal{L}_{\text{Bayes}}$ in Eq. (2) becomes the ELBO with respect to $\boldsymbol{\psi}$ and $\boldsymbol{\phi}$ jointly. As they are independent, the $\mathcal{L}_{\text{Bayes}}$ is expressed as:

$$\mathcal{L}_{\text{Bayes}} = -\mathbb{E}_{q(\boldsymbol{\psi})}\mathbb{E}_{q(\boldsymbol{\phi})}[\log p(\mathbf{y}|\mathbf{x},\boldsymbol{\psi},\boldsymbol{\phi})] + \mathbb{D}_{\text{KL}}[q(\boldsymbol{\psi})||p(\boldsymbol{\psi})] + \mathbb{D}_{\text{KL}}[q(\boldsymbol{\phi})||p(\boldsymbol{\phi})]. \tag{9}$$

The above variational inference objective allows us to explore domain-invariant representations and classifier in a unified way. The detailed derivation of Eq. (9) is provided in Appendix A.

Domain-Invariant Classifier To establish the domain-invariant classifier, we directly incorporate the proposed domain-invariant principle into the last layer of the network, which gives rise to

$$\mathcal{L}_{\text{I}}(\boldsymbol{\psi}) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{q(\boldsymbol{\psi})}\Big[\mathbb{D}_{\text{KL}}\big[p(\mathbf{y}_s|\mathbf{z}_s,\boldsymbol{\psi})||p(\mathbf{y}_t^i|\mathbf{z}_t^i,\boldsymbol{\psi})\big]\Big], \tag{10}$$

where $\mathbf{z}$ denotes the feature representations of input $\mathbf{x}$, and the subscripts s and t indicate the meta-source domain and the meta-target domains as in Eq. (7). Since $p(\mathbf{y}|\mathbf{z},\boldsymbol{\psi})$ is a Bernoulli distribution, we can conveniently calculate the KL divergence in Eq. (10).

Domain-Invariant Representations To also make the representations domain invariant, we have

$$\mathcal{L}_{\text{I}}(\boldsymbol{\phi}) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{q(\boldsymbol{\phi})}\Big[\mathbb{D}_{\text{KL}}\big[p(\mathbf{z}_s|\mathbf{x}_s,\boldsymbol{\phi})||p(\mathbf{z}_t^i|\mathbf{x}_t^i,\boldsymbol{\phi})\big]\Big], \tag{11}$$

where $\boldsymbol{\phi}$ are the parameters of the feature extractor. Since the feature extractor is also a Bayesian layer, the distribution of $p(\mathbf{z}|\mathbf{x},\boldsymbol{\phi})$ will be a factorized Gaussian if the posterior of $\boldsymbol{\phi}$ is as well. We illustrate this as follows. Let $\mathbf{x}$ be the input feature of a Bayesian layer $\boldsymbol{\phi}$, which has a factorized Gaussian posterior, the posterior of the activation $\mathbf{z}$ of the Bayesian layer is also a factorized Gaussian (Kingma et al., 2015):

$$\begin{aligned} q(\phi_{i,j}) \sim &\mathcal{N}(\mu_{i,j},\sigma^2_{i,j}) \quad \forall \phi_{i,j} \in \boldsymbol{\phi} \Rightarrow p(z_j|\mathbf{x},\boldsymbol{\phi}) \sim \mathcal{N}(\gamma_j,\delta^2_j), \\ \gamma_j &= \sum_{i=1}^{N} x_i\mu_{i,j}, \quad \text{and} \quad \delta^2_j = \sum_{i=1}^{N} x_i^2\sigma^2_{i,j}, \end{aligned} \tag{12}$$

where z_j denotes the j-th element in $\mathbf{z}$, likewise for x_i, and $\phi_{i,j}$ denotes the element at the position (i,j) in $\boldsymbol{\phi}$. Based on this property of the Bayesian framework, we assume that the posterior of our variational invariant feature extractor has a factorized Gaussian distribution, which leads to an easier calculation of the KL divergence in Eq. (11). Note that with the domain-invariant representations, $\mathbf{z}$ in Eq. (10) corresponds to samples of the feature representation distributions: $\mathbf{z}_s \sim p(\mathbf{z}_s|\mathbf{x}_s,\boldsymbol{\phi})$ and $\mathbf{z}_t \sim p(\mathbf{z}_t|\mathbf{x}_t,\boldsymbol{\phi})$.

2.3 OBJECTIVE FUNCTION

The objective function of our variational invariant learning is defined as:

$$\mathcal{L}_{\text{VIL}} = \mathcal{L}_{\text{Bayes}} + \lambda_{\boldsymbol{\psi}}\mathcal{L}_{\text{I}}(\boldsymbol{\psi}) + \lambda_{\boldsymbol{\phi}}\mathcal{L}_{\text{I}}(\boldsymbol{\phi}), \tag{13}$$

where $\lambda_{\boldsymbol{\psi}}$ and $\lambda_{\boldsymbol{\phi}}$ are hyperparameters to control the domain-invariant terms. We adopt Monte Carlo sampling and obtain the empirical objective function for variational invariant learning as follows:

$$\begin{aligned} \mathcal{L}_{\text{VIL}} = \frac{1}{L}\sum_{\ell=1}^{L}\frac{1}{M}\sum_{m}^{M}\big[-\log p(\mathbf{y}_s|\mathbf{x}_s,\boldsymbol{\psi}^{(\ell)},\boldsymbol{\phi}^{(m)})\big] + \mathbb{D}_{\text{KL}}\big[q(\boldsymbol{\psi})||p(\boldsymbol{\psi})\big] + \mathbb{D}_{\text{KL}}\big[q(\boldsymbol{\phi})||p(\boldsymbol{\phi})\big] \\ + \lambda_{\boldsymbol{\psi}}\frac{1}{T}\sum_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N}\frac{1}{L}\sum_{\ell=1}^{L}\mathbb{D}_{\text{KL}}\big[p(\mathbf{y}_s|\mathbf{z}_s,\boldsymbol{\psi}^{(\ell)})||p(\mathbf{y}_t^i|\mathbf{z}_t^i,\boldsymbol{\psi}^{(\ell)})\big] \\ + \lambda_{\boldsymbol{\phi}}\frac{1}{T}\sum_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N}\frac{1}{M}\sum_{m=1}^{M}\mathbb{D}_{\text{KL}}\big[p(\mathbf{z}_s|\mathbf{x}_s,\boldsymbol{\phi}^{(m)})||p(\mathbf{z}_t^i|\mathbf{x}_t^i,\boldsymbol{\phi}^{(m)})\big], \end{aligned} \tag{14}$$

where $\mathbf{x}_s$ and $\mathbf{z}_s$ denote the input and its feature from D_s, respectively, and $\mathbf{x}_t^i$ and $\mathbf{z}_t^i$ are from D_t as in Eq. (7). The posteriors are set to factorized Gaussian distributions, i.e., $q(\boldsymbol{\psi}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\psi}}, \boldsymbol{\sigma}^2_{\boldsymbol{\psi}})$ and $q(\boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}, \boldsymbol{\sigma}^2_{\boldsymbol{\phi}})$. We adopt the reparameterization trick to draw Monte Carlo samples (Kingma & Welling, 2014) as $\boldsymbol{\psi}^{(\ell)} = \boldsymbol{\mu}_{\boldsymbol{\psi}} + \epsilon^{(\ell)} * \boldsymbol{\sigma}_{\boldsymbol{\psi}}$, where $\epsilon^{(\ell)} \sim \mathcal{N}(0, I)$. We draw the samples for $\boldsymbol{\phi}^{(m)}$ in a similar way. In the implementation of our variational invariant learning, to increase the flexibility of the prior distribution in our Bayesian layers, we choose to place a scale mixture of two Gaussian distributions as the priors $p(\boldsymbol{\psi})$ and $p(\boldsymbol{\phi})$ (Blundell et al., 2015):

$$\pi\mathcal{N}(0, \boldsymbol{\sigma}_1^2) + (1-\pi)\mathcal{N}(0, \boldsymbol{\sigma}_2^2), \tag{15}$$

where $\boldsymbol{\sigma}_1$, $\boldsymbol{\sigma}_2$ and π are hyperparameters chosen by cross-validation.

3 Related work

One solution for domain generalization is to generate more source domain data to increase the probability of covering the data in the target domains (Shankar et al., 2018; Volpi et al., 2018). Shankar et al. (2018) augmented the data by perturbing the input images with adversarial gradients generated by an auxiliary classifier. Qiao et al. (2020) proposed a more challenging scenario of domain generalization named single domain generalization, which only has one source domain, and they designed an adversarial domain augmentation method to create "fictitious" yet "challenging" data. Recently, Zhou et al. (2020) employed a generator to synthesize data from pseudo-novel domains to augment the source domains, maximizing the distance between source and pseudo-novel domains as measured by optimal transport (Peyré et al., 2019). Another solution for domain generalization is based on learning domain-invariant features (D'Innocente & Caputo, 2018; Li et al., 2018b; 2017a). Muandet et al. (2013) proposed domain-invariant component analysis to learn invariant transformantions by minimizing the dissimilarity across domains. Louizos et al. (2015) learned invariant representations by variational autoencoder (Kingma & Welling, 2014), which introduced Bayesian inference into invariant feature learning. Dou et al. (2019) and Seo et al. (2019) tried to achieve a similar goal by introducing two complementary losses, global class alignment loss and local sample clustering loss, and employing multiple normalization methods. Li et al. (2019) proposed an episodic training algorithm to obtain both domain-invariant feature extractor and classifier.

Recently, meta-learning based techniques have been considered to solve domain generalization problems. Li et al. (2018a) proposed a meta-learning domain generalization method which introduced the gradient-based method, i.e., model agnostic meta-learning (Finn et al., 2017), for domain generalization. Balaji et al. (2018) address the domain generalization problem by learning a regularization function in a meta-learning framework, making the model robust to domain shifts. Du et al. (2020) propose the meta variational information bottleneck to learn domain-invariant representations through episodic training.

To the best of our knowledge, Bayesian neural networks have not yet been explored in domain generalization. Our method introduces variational Bayesian approximation to both the feature extractor and classifier of the neural network in conjunction with the newly introduced domain-invariant principle for domain generalization. The resultant variational invariant learning combines the representational power of deep neural networks and variational Bayesian inference.

Similar to our proposal, CCSA by Motiian et al. (2017) also aligns representations across domains in the same class. Specifically, CCSA utilizes an L2 distance between deterministic features while we exploit Bayesian neural networks to learn domain-invariant representations by minimizing the distance between domain distributions. Theoretically, minimizing the distance between distributions incorporates larger inter-class variance than minimizing distance of deterministic features. Moreover, we apply our variational invariant learning to both the feature extractor and the classifier, while CCSA only considers an alignment loss on the feature representations.

4 Experiments

4.1 Datasets and settings

We conduct our experiments on four widely used benchmarks in domain generalization:

PACS (Li et al., 2017a) consists of 9,991 images of seven classes from four domains - *photo*, *art-painting*, *cartoon* and *sketch*. We follow the "leave-one-out" protocol in (Li et al., 2017a; 2018b; Carlucci et al., 2019), where the model is trained on any three of the four domains, which we call source domains, and tested on the last (target) domain.

Office-Home (Venkateswara et al., 2017) also has four domains: *art*, *clipart*, *product* and *real-world*. There are about 15,500 images of 65 categories for object recognition in office and home environments. We use the same experimental protocol as for PACS.

Rotated MNIST and Fashion-MNIST were introduced for evaluating domain generalization in (Piratla et al., 2020). For the fair comparison, we follow their recommended settings and randomly select a subset of 2,000 images from MNIST and 10,000 images from Fashion-MNIST, which is considered to have been rotated by $0°$. The subset of images is then rotated by $15°$ through $75°$ in intervals of $15°$, creating five source domains. The target domains are created by rotation angles of $0°$ and $90°$. We use these two datasets to demonstrate the generalizability by comparing the performance on in-distribution and out-of-distribution data.

For all four benchmarks, we use ResNet-18 (He et al., 2016) pretrained on ImageNet (Deng et al., 2009) as the CNN backbone. During training, we use Adam optimization (Kingma & Ba, 2014) with the learning rate set to 0.0001, and train for 10,000 iterations. In each iteration we choose one source domain as the meta-source domain. The batch size is 128. To fit memory footprint, we choose a maximum number of samples per category per target domain to implement the domain-invariant learning, i.e. sixteen for PACS, Rotated MNIST and Fashion-MNIST datasets, and four for the Office-Home dataset. We choose λ_ϕ and λ_ψ based on the performance on the validation set and their influence is summarized in the new Fig 3 in Appendix C. The optimal values of λ_ϕ and λ_ψ are 0.1 and 100 respectively. Parameters σ_1 and σ_2 in Eq. (15) are set to 0.1 and 1.5. The model with the highest validation set accuracy is employed for evaluation on the target domain. All code will be made publicly available.

4.2 Ablation Study

We conduct an ablation study to investigate the effectiveness of our variational invariant learning for domain generalization. The experiments are performed on the PACS dataset. Since the major contributions of this work are the Bayesian treatment and the domain-invariant principle, we evaluate their effect by individually incorporating them into the classifier - the last layer - ψ and the feature extractor - the penultimate layer - ϕ. The results are shown in Table 1. The "✓" and "×" in the "Bayesian" column denote whether the classifier ψ and feature extractor ϕ are Bayesian layers or deterministic layers. In the "Invariant" column they denote whether the domain-invariant loss is introduced into the classifier and the feature extractor. Note that the predictive distribution is a Bernoulli distribution, which also admits the domain-invariant loss, we therefore include this case for a comprehensive comparison.

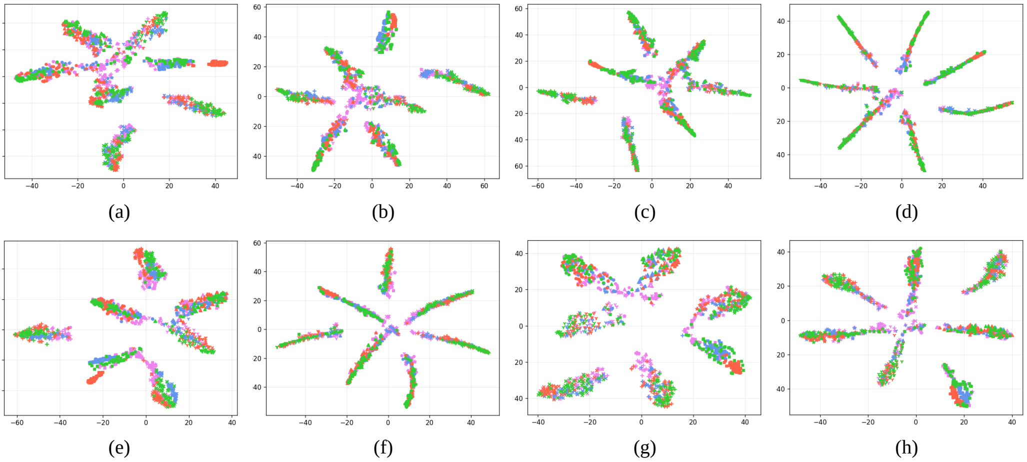
In Table 1, the first four rows demonstrate the benefit of the Bayesian treatment. The first row (a) serves as a baseline model, which is a vanilla deep convolutional network without any Bayesian treatment and domain-invariant loss. The backbone is also a ResNet-18 pretrained on ImageNet. It is clear the Bayesian treatment, either for the classifier (b) or the feature extractor (c), improves the performance, especially in the "Art-painting" and "Sketch" domains, and this is further demonstrated in (d) where we employ the Bayesian classifier and feature extractor simultaneously.

The benefit of the domain-invariant principle for classifiers is demonstrated by comparing (e) to (a) and (f) to (b). The settings with domain invariance consistently perform better than those without it. A similar trend is also observed when applying the domain-invariant principle to the feature extractor, as shown by comparing (g) to (c). Overall, our variational invariant learning (h) achieves the best performance compared to other variants, demonstrating its effectiveness for domain generalization. Note that the feature distributions $p(\mathbf{z}|\mathbf{x})$ are unknown without Bayesian formalism, leading to an intractable $\mathcal{L}_I(\phi)$. Therefore, we do not conduct the experiment with only the domain-invariant loss on both the classifier and the feature extractor.

To further demonstrate the domain-invariant property of our method, we visualize the features learned by different settings of our method in Table 1. We use t-SNE (Maaten & Hinton, 2008)

Table 1: Ablation study on PACS. All the individual components of our variational invariant learning benefit domain generalization performance. More comparisons can be found in Appendix D

	Bayesian		**Invariant**		**PACS**				
ID	ψ	ϕ	ψ	ϕ	Photo	Art-painting	Cartoon	Sketch	*Mean*
(a)	×	×	×	×	92.85	75.12	77.44	75.72	80.28
(b)	✓	×	×	×	93.89	77.88	78.20	77.75	81.93
(c)	×	✓	×	×	92.81	78.66	77.90	78.72	82.02
(d)	✓	✓	×	×	93.83	82.13	79.18	79.03	83.73
(e)	×	×	✓	×	93.95	80.03	78.03	77.83	82.46
(f)	✓	×	✓	×	95.21	81.25	80.67	79.31	84.11
(g)	×	✓	×	✓	95.15	80.96	79.57	79.15	83.71
(h)	✓	✓	✓	✓	**95.97**	**83.92**	**81.61**	**80.31**	**85.45**



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1: Visualization of feature representations. The eight sub-figures correspond to the eight settings in Table 1 (identified by ID). Colors denote domains, while shapes indicate classes. The target domain (violet) is "art-painting". The top row shows the Bayesian treatment enlarges the inter-class distance for all domains, considerably. The bottom row, compared with the top-row figures in the same column, shows the domain-invariant principle enlarges the inter-class distance in the target domain by reducing the intra-class distances between the source and target domains.

to reduce the feature dimension into a two-dimensional subspace, following Du et al. (2020). The visualization is shown in Fig. 1. More visualization results are provided in Appendix E.

Figs. 1 (a), (b), (c) and (d) show *the baseline, Bayesian classifier, Bayesian representations* and *both Bayesian classifier and representations*, demonstrating the benefits of Bayesian inference for learning domain-invariant features. The Bayesian treatment on both the classifier and the feature extractor enlarges the inter-class distance in all domains, which benefits the classification performance on the target domain, as shown in Fig. 1 (d). Figs. 1 (e), (f), (g), (h) are visualizations of feature representations after introducing the domain-invariant principle, compared to the upper row. Comparing the two figures in each column indicates that our domain-invariant principle imposed on either the representation or the classifier further enlarges the inter-class distances. At the same time, it reduces the distance between samples of the same class from different domains. This is even more apparent in the intra-class distance between samples from source and target domains. As a result, the inter-class distances in the target domain become larger, therefore improving performance. It is worth noting that the domain-invariant principle on the classifier in Fig. 1 (f) and on the feature extractor in Fig. 1 (g)) both improve the domain-invariant features. Our variational invariant learning in Fig. 1 (h) therefore has better performance by combining their benefits.

Table 2: Ablation with two variational invariant layers in the feature extractor on PACS. Bayesian ϕ' and Invariant ϕ' denote whether the additional variational invariant layer in the feature extractor has a Bayesian property and domain-invariant property. More Bayesian layers benefits the performance while excessive domain-invariant learning harms it.

Bayesian ϕ'	Invariant ϕ'	Photo	Art-painting	Cartoon	Sketch	*Mean*
×	×	**95.97**	**83.92**	81.61	80.31	85.45
✓	×	95.69	83.28	**82.06**	**81.00**	**85.51**
✓	✓	95.72	82.33	81.10	80.67	84.96

We also experiments with more layers in the feature extractor, see Table 2. "Bayesian ϕ'" and "Invariant ϕ'" denote whether the additional feature extraction layer ϕ' has the Bayesian property and domain-invariant property. The classifiers have both properties in all cases in Table 2. The first row is the setting with only one variational invariant layer in the feature extractor. When introducing another Bayesian learning layer ϕ' without the domain-invariant property into the model, as shown in the second row in Table 2, the average performance improves slightly. If we introduce both the Bayesian learning and domain-invariant learning into ϕ', as shown in the third row, the overall performance declines a bit. One reason might be the information loss in feature representations caused by the excessive use of domain-invariant learning. In addition, due to the Bayesian inference and Monte-Carlo sampling, more variational-invariant layers leads to higher memory usage and more computations, which is also one reason for us to apply the variational invariant learning only to the last feature extraction layer and the classifier.

Table 3: Comparison on PACS. Our method achieves the best performance on the "Cartoon" domain, is competitive on the other three domains and obtains the best overall mean accuracy.

	Photo	Art-painting	Cartoon	Sketch	*Mean*
Baseline	92.85	75.12	77.44	75.72	80.28
JiGen (Carlucci et al., 2019)	96.03	79.42	75.25	71.35	80.51
Epi-FCR (Li et al., 2019)	93.90	82.10	77.00	73.00	81.50
MetaReg (Balaji et al., 2018)	95.50	83.70	77.20	70.30	81.68
MASF (Dou et al., 2019)	94.99	80.29	77.17	71.69	81.04
CSD (Piratla et al., 2020)	94.10	78.90	75.80	76.70	81.38
DMG (Chattopadhyay et al., 2020)	93.35	76.90	80.38	75.21	81.46
L2A-OT (Zhou et al., 2020)	**96.20**	83.30	78.20	73.60	82.83
DSON (Seo et al., 2019)	95.87	**84.67**	77.65	**82.23**	85.11
RSC (Huang et al., 2020)	95.99	83.43	80.31	80.85	85.15
VIL (*This paper*)	95.97	83.92	**81.61**	80.31	**85.45**

4.3 State-of-the-art comparison

In this section, we compare our method with several state-of-the-art methods on four datasets. The results are reported in Tables 3-5. The baseline on PACS (Table 3), Office-Home (Table 4), and rotated MNIST and Fashion-MNIST (Table 5) are all based on the same vanilla deep convolutional ResNet-18 network, without any Bayesian treatment, the same as row (a) in Table 1

On PACS, as shown in Table 3, our variational invariant learning method achieves the best overall performance. On each domain, our performance is competitive with the state-of-the-art and we exceed all other methods on the "Cartoon" domain. On Office-Home, as shown in Table 4, we again achieve the best recognition accuracy. It is worth mentioning that on the most challenging "Art" and "Clipart" domains, our variational invariant learning also delivers the highest performance, with a good improvement over previous methods.

L2A-OT and DSON outperform the proposed model on some domains of PACS and Office-Home. L2A-OT learns a generator to synthesize data from pseudo-novel domains to augment the source domains. The pseudo-novel domains often have similar characteristics with the source data. Thus, when the target data also have similar characteristics with the source domains this pays off as the

Table 4: Comparison on Office-Home. Our variational invariant learning achieves the best performance on the "Art" and "Clipart" domains, while being competitive on the "Product" and "Real" domains. Again we report the best overall mean accuracy.

	Art	Clipart	Product	Real	*Mean*
Baseline	54.84	49.85	72.40	73.14	62.55
JiGen (Carlucci et al., 2019)	53.04	47.51	71.47	72.79	61.20
CCSA (Motiian et al., 2017)	59.90	49.90	74.10	75.70	64.90
MMD-AAE (Li et al., 2018b)	56.50	47.30	72.10	74.80	62.68
CrossGrad (Shankar et al., 2018)	58.40	49.40	73.90	75.80	64.38
DSON (Seo et al., 2019)	59.37	45.70	71.84	74.68	62.90
RSC (Huang et al., 2020)	58.42	47.90	71.63	74.54	63.12
L2A-OT (Zhou et al., 2020)	60.60	50.10	**74.80**	**77.00**	65.63
VIL (*This paper*)	**61.81**	**53.27**	74.27	76.31	**66.42**

Table 5: Comparison on Rotated MNIST and Fashion-MNIST. In-distribution performance is evaluated on the test sets of MNIST and Fashion-MNIST with rotation angles of $15°$, $30°$, $45°$, $60°$ and $75°$, while the out-of-distribution performance is evaluated on test sets with angles of $0°$ and $90°$. Our VIL achieves the best performance on both the in-distribution and out-of-distribution test sets.

	MNIST		**Fashion-MNIST**	
	In-distribution	Out-of-distribution	In-distribution	Out-of-distribution
Baseline	98.4	93.5	89.6	76.9
MASF (Dou et al., 2019)	98.2	93.2	86.9	72.4
CSD (Piratla et al., 2020)	98.4	94.7	89.7	78.0
VIL (*This paper*)	**99.0**	**96.5**	**91.5**	**83.5**

pseudo domains are more likely to cover the target domain, such as "Product" and "Real World" in Office-Home and "Photo" in PACS. When the test domain is different from all of the training domains the performance suffers, e.g., "Clipart" in Office-Home and "Sketch" in PACS. Our method generates domain-invariant representations and classifiers, resulting in competitive results across all domains and overall. DSON mixtures batch and instance normalization for domain generalization. This tactic is effective on PACS, but less competitive on Office-Home. We attribute this to the larger number of categories on Office-Home, where instance normalization is known to make features less discriminative with respect to object categories (Seo et al., 2019). Our domain-invariant network makes feature distributions and predictive distributions similar across domains, resulting in good performance on both PACS and Office-Home.

On the Rotated MNIST and Fashion-MNIST datasets, following the experimental settings in (Piratla et al., 2020), we evaluate our method on the in-distribution and out-of-distribution sets. As shown in Table 5, our VIL achieves the best performance on both sets of the two datasets, surpassing other methods. Moreover, our method especially improves the classification performance on the out-of-distribution sets, demonstrating its strong generalizability to unseen domains, which is also consistent with the findings in Fig. 1.

5 Conclusion

In this work, we propose *variational invariant learning* (VIL), a variational Bayesian learning framework for domain generalization. We introduce Bayesian neural networks into the model, which is able to better represent uncertainty and enhance the generalization to out-of-distribution data. To handle the domain shift between source and target domains, we propose a domain-invariant principle under the variational inference framework, which is incorporated by establishing a domain-invariant feature extractor and classifier. Our variational invariant learning combines the representational power of deep neural networks and uncertainty modeling ability of Bayesian learning, showing great effectiveness for domain generalization. Extensive ablation studies demonstrate the benefits of the Bayesian inference and domain-invariant principle for domain generalization. Our variational invariant learning sets a new state-of-the-art on four domain generalization benchmarks.

References

Andrei Atanov, Arsenii Ashukha, Kirill Struminsky, Dmitriy Vetrov, and Max Welling. The deep weight prior. In *International Conference on Learning Representations*, 2019. 1

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems*, pp. 998–1008, 2018. 1, 5, 8

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. 2

Charles Blundell, Julien Cornebise, K. Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *ArXiv*, abs/1505.05424, 2015. 1, 5, 13

Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2229–2238, 2019. 6, 8, 9

Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. *arXiv preprint arXiv:2008.12839*, 2020. 8

Erik Daxberger and José Miguel Hernández-Lobato. Bayesian variational autoencoders for unsupervised out-of-distribution detection. *arXiv preprint arXiv:1912.05651*, 2019. 3

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. 6

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, pp. 6450–6461, 2019. 5, 8, 9

Yingjun Du, Jun Xu, Huan Xiong, Qiang Qiu, Xiantong Zhen, Cees GM Snoek, and Ling Shao. Learning to learn with variational information bottleneck for domain generalization. *arXiv preprint arXiv:2007.07645*, 2020. 1, 5, 7

Antonio D'Innocente and Barbara Caputo. Domain generalization with domain-specific aggregation modules. In *German Conference on Pattern Recognition*, pp. 187–198. Springer, 2018. 1, 5

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135, 2017. 5

A. Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, 2011. 2

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330, 2017. 1

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. 1, 6

Geoffrey E. Hinton and D. Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *COLT '93*, 1993. 2

Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. *arXiv preprint arXiv:2007.02454*, 2020. 8, 9

Maximilian Ilse, Jakub M Tomczak, Christos Louizos, and Max Welling. Diva: Domain invariant variational autoencoders. *arXiv preprint arXiv:1905.10427*, 2019. 1

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. 5

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in neural information processing systems*, pp. 2575–2583, 2015. 4

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. *arXiv preprint arXiv:2002.10118*, 2020. 1, 2, 3

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012. 1

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5542–5550, 2017a. 1, 5, 6

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. *arXiv preprint arXiv:1710.03463*, 2017b. 1

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI Conference on Artificial Intelligence*, 2018a. 1, 5

Da Li, J. Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M. Hospedales. Episodic training for domain generalization. *IEEE/CVF International Conference on Computer Vision*, pp. 1446–1455, 2019. 3, 4, 5, 8

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5400–5409, 2018b. 5, 6, 9

Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. In *International Conference on Machine Learning*, pp. 2218–2227, 2017. 1

Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*, 2015. 5

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 6

David JC MacKay. The evidence framework applied to classification networks. *Neural computation*, 4(5):720–736, 1992. 1, 2

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, pp. 7047–7058, 2018. 1

Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *IEEE International Conference on Computer Vision*, pp. 5715–5725, 2017. 5, 9

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pp. 10–18, 2013. 1, 5

Eric Nalisnick and Padhraic Smyth. Learning priors for invariance. In *International Conference on Artificial Intelligence and Statistics*, pp. 366–375, 2018. 2, 3

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015. 1

Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018. 1

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. 5

Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. *arXiv preprint arXiv:2003.12815*, 2020. 6, 8, 9

Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12556–12565, 2020. 5

Seonguk Seo, Yumin Suh, Dongwan Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. *arXiv preprint arXiv:1907.04275*, 2019. 5, 8, 9

Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018. 5, 9

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5018–5027, 2017. 6

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Advances in neural information processing systems*, pp. 5334–5344, 2018. 5

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. *arXiv preprint arXiv:2007.03304*, 2020. 5, 8, 9

A DERIVATION

A.1 DERIVATION OF THE UPPER BOUNDS OF DOMAIN-INVARIANT LEARNING

As in most cases the $\mathbb{E}_{q_\zeta}[p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)]$ is intractable, we derive the upper bound in Eq. (5), which is achieved via Jensen's inequality:

$$\begin{aligned}\mathbb{D}_{\mathrm{KL}}\left[p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s)||\mathbb{E}_{q_\zeta}[p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)]\right] &= \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s)}[\log p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s)] - \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s)}\left[\log \mathbb{E}_{q_\zeta}[p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)]\right] \\ &\leq \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s)}[\log p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s)] - \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s)}\mathbb{E}_{q_\zeta}[\log p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)] \\ &= \mathbb{E}_{q_\zeta}\Big[\mathbb{D}_{\mathrm{KL}}\left[p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s)||p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)\right]\Big].\end{aligned} \tag{16}$$

In Bayesian inference, computing the likelihood $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) = \int_{\boldsymbol{\theta}} p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} = \mathbb{E}_{q(\boldsymbol{\theta})}[p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})]$ is notoriously difficult. Thus, as the fact that KL divergence is a convex function, we obtain the upper bound in Eq. (6) achieved via Jensen's inequality similar to Eq. (16):

$$\begin{aligned}\mathbb{E}_{q_\zeta}\Big[\mathbb{D}_{\mathrm{KL}}\left[p_{\boldsymbol{\theta}}(\mathbf{y}_s|\mathbf{x}_s)||p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)\right]\Big] &= \mathbb{E}_{q_\zeta}\Big[\mathbb{D}_{\mathrm{KL}}\left[\mathbb{E}_{q(\boldsymbol{\theta})}[p(\mathbf{y}_s|\mathbf{x}_s,\boldsymbol{\theta})]||\mathbb{E}_{q(\boldsymbol{\theta})}[p(\mathbf{y}_\zeta|\mathbf{x}_\zeta,\boldsymbol{\theta})]\right]\Big] \\ &\leq \mathbb{E}_{q_\zeta}\Big[\Big[\mathbb{E}_{q(\boldsymbol{\theta})}\mathbb{D}_{\mathrm{KL}}\left[p(\mathbf{y}_s|\mathbf{x}_s,\boldsymbol{\theta})||p(\mathbf{y}_\zeta|\mathbf{x}_\zeta,\boldsymbol{\theta})\right]\Big]\Big].\end{aligned} \tag{17}$$

A.2 DERIVATION OF VARIATIONAL BAYESIAN APPROXIMATION FOR REPRESENTATION ($\boldsymbol{\phi}$) AND CLASSIFIER (ψ) LAYERS.

We consider the model with two Bayesian layers $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ as the last layer of feature extractor and the classifier respectively. The prior distribution of the model is $p(\boldsymbol{\phi},\boldsymbol{\psi})$, and the true posterior distribution is $p(\boldsymbol{\phi},\boldsymbol{\psi}|\mathbf{x},\mathbf{y})$. Following the settings in Section 2.1, we need to learn a variational distribution $q(\boldsymbol{\phi},\boldsymbol{\psi})$ to approximate the true posterior by minimizing the KL divergence from $q(\boldsymbol{\phi},\boldsymbol{\psi})$ to $p(\boldsymbol{\phi},\boldsymbol{\psi}|\mathbf{x},\mathbf{y})$:

$$\boldsymbol{\phi}^*,\boldsymbol{\psi}^* = \underset{\boldsymbol{\phi},\boldsymbol{\psi}}{\arg\min}\,\mathbb{D}_{\mathrm{KL}}\left[q(\boldsymbol{\phi},\boldsymbol{\psi})||p(\boldsymbol{\phi},\boldsymbol{\psi}|\mathbf{x},\mathbf{y})\right]. \tag{18}$$

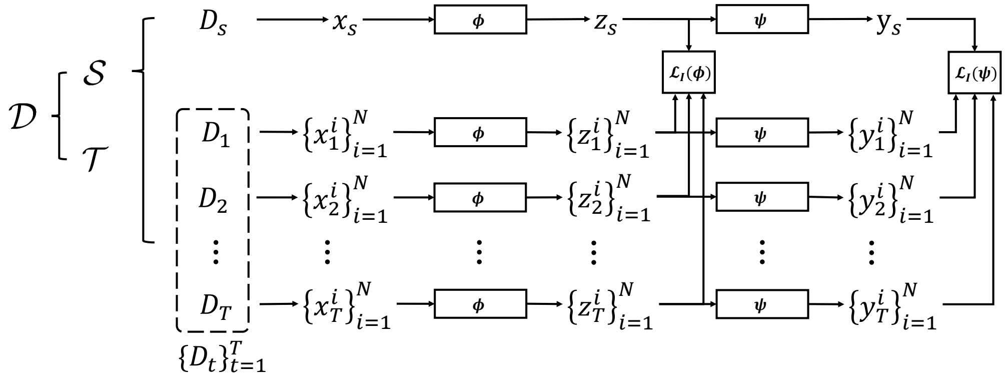


Figure 2: Illustration of the domain-invariant loss in the training phase of VIL. $\mathcal{S}$ denotes the source domains, $\mathcal{T}$ denotes the target domains, and $\mathcal{D} = \mathcal{S} \cup \mathcal{D}$. $\mathbf{x}$, $\mathbf{z}$ and $\mathbf{y}$ denote inputs, features and outputs of samples in each domain. $\mathcal{L}_I(\phi)$ and $\mathcal{L}_I(\psi)$ denote the domain-invariant loss functions for representations and the classifier.

By applying the Bayesian rule $p(\phi, \psi|\mathbf{x}, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}, \phi, \psi)p(\phi, \psi)$, the optimization is equivalent to minimizing:

$$\begin{aligned}\mathcal{L}_{\text{Bayes}} &= \int q(\phi, \psi) \log \frac{q(\phi, \psi)}{p(\phi, \psi)p(\mathbf{y}|\mathbf{x}, \phi, \psi)} \mathrm{d}\phi \mathrm{d}\psi \\ &= \mathbb{D}_{\text{KL}}\big[q(\phi, \psi)||p(\phi, \psi)\big] - \mathbb{E}_{q(\phi,\psi)}\big[\log p(\mathbf{y}|\mathbf{x}, \phi, \psi)\big].\end{aligned} \tag{19}$$

With ϕ and ψ are independent,

$$\mathcal{L}_{\text{Bayes}} = -\mathbb{E}_{q(\psi)}\mathbb{E}_{q(\phi)}[\log p(\mathbf{y}|\mathbf{x}, \psi, \phi)] + \mathbb{D}_{\text{KL}}[q(\psi)||p(\psi)] + \mathbb{D}_{\text{KL}}[q(\phi)||p(\phi)]. \tag{20}$$

B Details of domain-invariant loss in VIL training

We split the training phase of VIL into several episodes. In each episode, as shown in Fig. 2, we randomly choose a source domain as the meta-source domain D_s, and the rest of the source domains $\{D_t\}_{t=1}^T$ are treated as the meta-target domains. From D_s, we randomly select a batch of samples $\mathbf{x}_s$. For each $\mathbf{x}_s$, we then select N samples $\{\mathbf{x}_t^i\}_{i=1}^N$, which are in the same category as $\mathbf{x}_s$, from each of the meta-target domains D_t. All of these samples are sent to the variational invariant feature extractor ϕ to get the representations $\mathbf{z}_s$ and $\{\mathbf{z}_t^i\}_{i=1}^N$, which are then sent to the variational invariant classifier ψ to obtain the predictions $\mathbf{y}_s$ and $\{\mathbf{y}_t^i\}_{i=1}^N$. We obtain the domain-invariant loss for feature extractor $\mathcal{L}_I(\phi)$ by calculating the mean of the KL divergence of $\mathbf{z}_s$ and each $\mathbf{z}_t^i$ as Eq.(11). The domain-invariant loss for feature classifier $\mathcal{L}_I(\psi)$ is calculated in a similar way on $\mathbf{y}_s$ and $\{\mathbf{y}_t^i\}_{i=1}^N$ as Eq.(10).

C Ablation study for hyperparamters

We also ablate the hyperparameters λ_ϕ, λ_ψ and π on PACS with cartoon as the target domain. Results are shown in Fig 3 (a), (b) and (c). We obtain Fig 3 (a) by fixing λ_ψ as 100 and adjusting λ_ϕ, Fig 3 (b) by fixing λ_ϕ as 1 and adjusting λ_ψ and Fig 3 (c) by adjusting π while fixing other settings as in Section 4.1. λ_ϕ and λ_ψ balance the influence of the Bayesian learning and domain-invariant learning, and their optimal values are 1 and 100. If the values are too small, the model tends to overfit to source domains as the performance on target data drops more obviously than on validation data. In contrast, too large values of them harm the overall performance of the model as there are obvious decrease of accuracy on both validation data and target data. Moreover, π balances the two components of the scale mixture prior of our Bayesian model. According to Blundell et al. (2015), the two components cause a prior density with heavier tail while many weights tightly concentrate around zero. Both of them are important. The performance is the best when π is 0.5 according to Fig 3 (c), which demonstrates the effectiveness of the two components in the scale mixture prior.

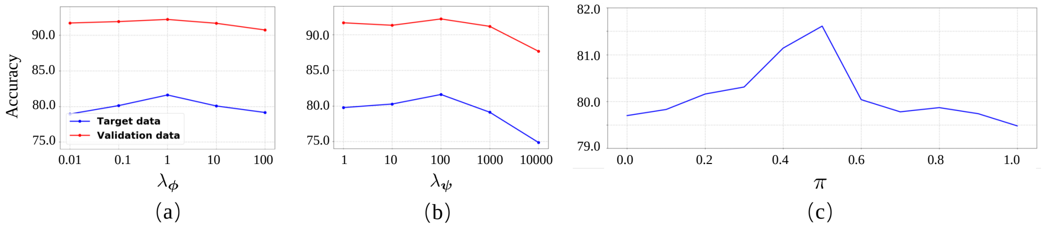


Figure 3: Performance on “Cartoon” domain in PACS with different hyperparameters λ_ϕ, λ_ψ and π. The red line denotes the accuracy on validation data while the blue line denotes accuracy on target data. The optimal value of λ_ϕ, λ_ψ and π are 1, 100 and 0.5 respectively.

Table 6: More detailed ablation study on PACS. Compared to Table 1 we add three more settings with IDs (i), (j) and (k). All the individual components of our variational invariant learning benefit domain generalization performance.

	Bayesian		**Invariant**		**PACS**				
ID	ψ	ϕ	ψ	ϕ	Photo	Art painting	Cartoon	Sketch	*Mean*
(a)	×	×	×	×	92.85	75.12	77.44	75.72	80.28
(b)	✓	×	×	×	93.89	77.88	78.20	77.75	81.93
(c)	×	✓	×	×	92.81	78.66	77.90	78.72	82.02
(d)	✓	✓	×	×	93.83	82.13	79.18	79.03	83.73
(e)	×	×	✓	×	93.95	80.03	78.03	77.83	82.46
(f)	✓	×	✓	×	95.21	81.25	80.67	79.31	84.11
(g)	×	✓	×	✓	95.15	80.96	79.57	79.15	83.71
(i)	✓	✓	✓	×	95.33	81.20	80.97	80.07	84.39
(j)	✓	✓	×	✓	95.87	81.15	79.39	80.15	84.14
(k)	×	✓	✓	✓	95.39	82.32	80.27	79.61	84.40
(h)	✓	✓	✓	✓	**95.97**	**83.92**	**81.61**	**80.31**	**85.45**

D Detailed ablation study on PACS

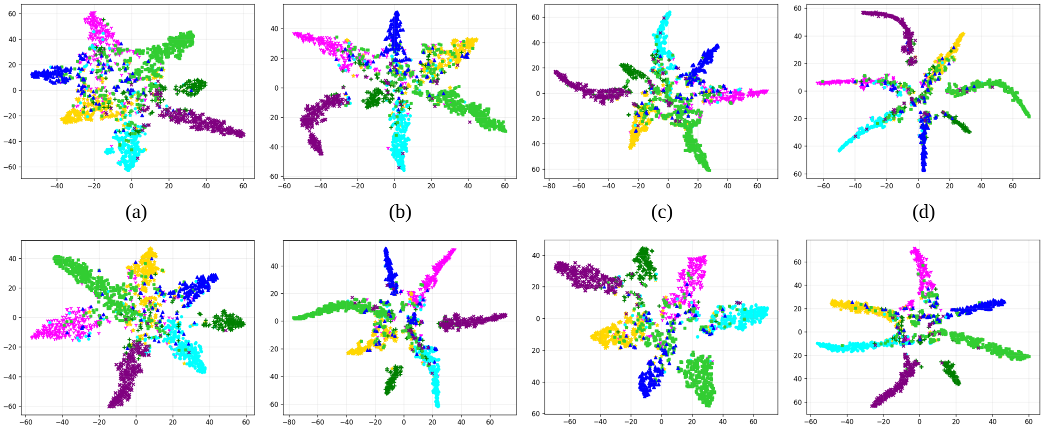
In addition to the aforementioned experiments, we conduct some supplementary experiments with other settings on PACS to further demonstrate the effectiveness of the Bayesian inference and domain-invariant loss as shown in Table 6. The evaluated components are the same as in Table 1. For better comparison, we show the contents of Table 1 again in Table 6, and add three other settings with IDs (i), (j) and (k). Note that as the distribution of features $\mathbf{z}$ is unknown without a Bayesian feature extractor ϕ, the settings with $\mathcal{L}_I(\phi)$ and a non-Bayesian feature extractor is intractable.

Comparing (i) with (d), we find that employing Bayesian inference to the last layer of the feature extractor improves the overall performance and the classification accuracy on three of the four domains. Moreover, comparing (j) and (k) to (g) shows the benefits of introducing variational Bayes and the domain-invariant loss to the classifier on most of the domains and the average of them.

E Extra visualization results

To further observe and analyze the benefits of the individual components of VIL for domain-invariant learning, we visualize the features of all categories from the target domain only in Fig. 4, and features of only one category from all domains in Fig. 5. The same as Fig. 1, the visualization is conducted on the PACS dataset and the target domain is “art-painting”. The chosen category in Fig. 5 is “horse”.

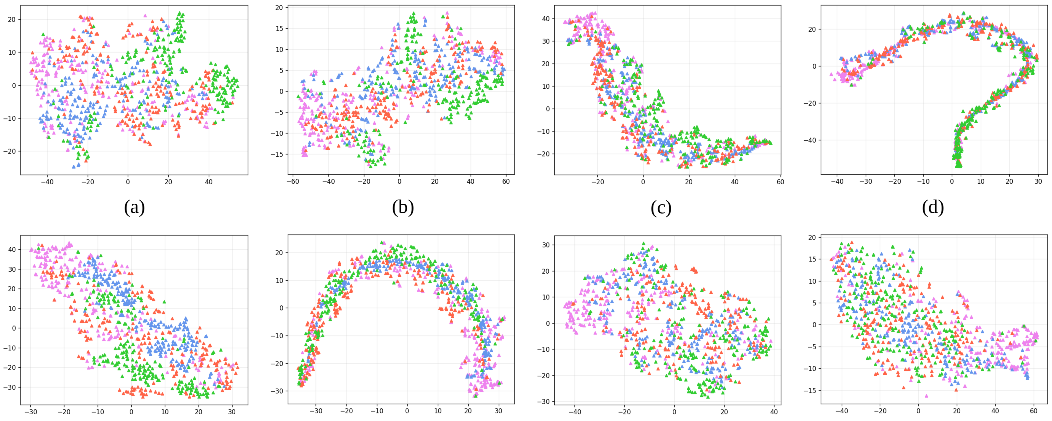
Fig. 4 provides a more intuitive observation of the benefits of the Bayesian framework and domain-invariant learning in our method for enlarging the inter-class distance in the target domain. The conclusion is similar as in Fig. 1. From the figures in the first row, it is clear that the Bayesian framework whether in the classifier ((b)) or the feature extractor ((c)) increases the inter-class distance compared with the baseline method ((a)). With both of them ((d)), the performance becomes



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4: Visualization of feature representations of the target domain. Different colors denote different categories. The sub-figures have the same experimental settings as the experiments in Table 1 and Fig. 1. Visualizing only the feature representations of the target domain shows the benefits of the individual components to the target domain recognition more intuitively. The target domain is "art-painting", as in Fig. 1. We obtain a similar conclusion to Section 4.2, where the Bayesian inference enlarges the inter-class distance for all domains and the domain-invariant principle reduces the intra-class distance of the source and target domains.



(a) (b) (c) (d)

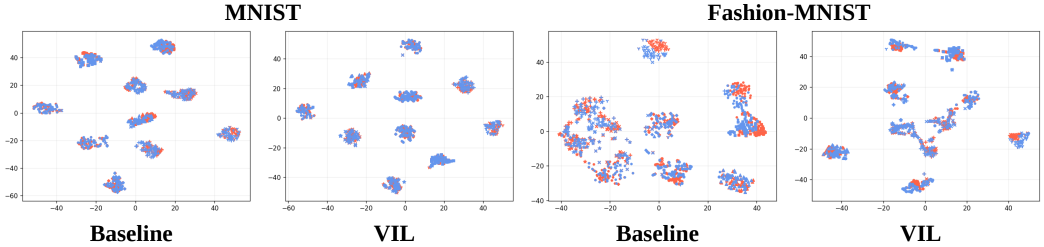
(e) (f) (g) (h)

Figure 5: Visualization of feature representations of one category. All samples are from the "horse" category with colors denoting different domains. The target domain is "art-painting" (violet). The top row shows Bayesian inference benefits domain generalization by gathering features from different domains to the same manifold. The figures in each column indicate domain-invariant learning reduces the intra-class distance between domains, resulting in better target domain performance.

better. Further, comparing two figures in each column, the inter-class distance is also enlarged by introducing the domain-invariant principle into the setting of each figure in the first row. VIL ((h)) achieves the best performance by combining the benefits of both the Bayesian framework and the domain-invariant principle in both feature extractor and classifier.

Fig. 5 provides a deeper insight into the intra-class feature distributions of the same category from different domains. By introducing the Bayesian inference into the model, the features demonstrate the manifold of the category as shown in the first row ((b), (c) and (d)). This makes recognition easier. Indeed, the visualization of features from multiple categories has similar properties as shown in Fig. 1. As shown in each column, introducing the domain-invariant learning into the model leads to a better mixture of features from different domains. The resultant domain-invariant representation makes the model more generalizable to unseen domains.

We also visualize the features in rotated MNIST and Fashion MNIST datasets, as shown in Fig. 6. Different shapes denote different categories. Red samples denote features from the in-distribution set and blue samples denote features from the out-of-distribution set. Compared with the baseline, our method reduces the intra-class distance between samples from the in-distribution set and the out-of-distribution set and clusters the out-of-distribution samples of the same categories better, especially in the rotated Fashion-MNIST dataset.


MNIST

Fashion-MNIST

Baseline

VIL

Baseline

VIL

Figure 6: Visualization of feature representations in rotated MNIST and rotated Fashion-MNIST datasets. Samples from the in-distribution and out-of-distribution sets are in red and blue, respectively. Different shapes denote different categories. Compared to other methods, our VIL achieves better performance on both the in-distribution and out-of-distribution sets in each dataset, and especially on the out-of-distribution set from the Fashion-MNIST benchmark.