A Bit More Bayesian: Domain-Invariant Learning with Uncertainty

Zehao Xiao [1]   **Jiayi Shen** [1]   **Xiantong Zhen** [1][2]   **Ling Shao** [2]   **Cees G. M. Snoek** [1]

Abstract

Domain generalization is challenging due to the domain shift and the uncertainty caused by the inaccessibility of target domain data. In this paper, we address both challenges with a probabilistic framework based on variational Bayesian inference, by incorporating uncertainty into neural network weights. We couple domain invariance in a probabilistic formula with the variational Bayesian inference. This enables us to explore domain-invariant learning in a principled way. Specifically, we derive domain-invariant representations and classifiers, which are jointly established in a two-layer Bayesian neural network. We empirically demonstrate the effectiveness of our proposal on four widely used cross-domain visual recognition benchmarks. Ablation studies validate the synergistic benefits of our Bayesian treatment when jointly learning domain-invariant representations and classifiers for domain generalization. Further, our method consistently delivers state-of-the-art mean accuracy on all benchmarks.

1. Introduction

Learning to improve the generalization of deep neural networks to data out of their training distribution remains a fundamental yet challenging problem for machine learning (Wang et al., 2018; Bengio et al., 2019; Krueger et al., 2020). Domain generalization (Muandet et al., 2013) aims to train a model on several source domains and have it generalize well to unseen target domains. The main challenge stems from the large shift of distributions between the source and target domains, which is further complicated by the prediction uncertainty (Malinin & Gales, 2018) introduced by the inaccessibility to data from the target domains during training. Established approaches learn domain-invariant features by dedicated loss functions (Muandet et al., 2013;

Li et al., 2018a) or specific architectures (Li et al., 2017; D'Innocente & Caputo, 2018). The state-of-the-art relies on advanced deep neural network backbones, known to degenerate 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).

Domain generalization is susceptible to uncertainty as the domain shift from source to target domain is unknown a priori. Hence, uncertainty should be taken into account during domain-invariant learning. As deep neural networks are commonly trained by maximum likelihood estimation, they fail to effectively capture model uncertainty. This tends to make the models overconfident in their predictions, especially on out-of-distribution data (Daxberger & Hernández-Lobato, 2019). As a possible solution, approximate Bayesian inference offers a natural framework to represent prediction uncertainty (Kristiadi et al., 2020; MacKay, 1992). It 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). It even improves prediction uncertainty (Blundell et al., 2015; Atanov et al., 2019), 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, as of yet, unexplored scenario of domain generalization.

In this paper, we propose to address domain generalization under a probabilistic framework. To better explore domain-invariant learning, we introduce weight uncertainty to the model by leveraging variational Bayesian inference. To this end, we introduce the principle of domain invariance in a probabilistic formulation and incorporate it into the variational Bayesian inference framework. This enables us to explore domain invariance in a principled way to achieve domain-invariant feature representations and classifiers jointly. We evaluate our method on four widely-used benchmarks for cross-domain object classification. Our ablation studies demonstrate the benefit of domain-invariant learning in a probabilistic framework through its synergy with variational Bayesian inference, as well as the advantage of jointly learning domain-invariant feature extractors and classifiers for domain generalization. Our method achieves state-of-the-art mean accuracy on all four benchmarks.

---

[*]Equal contribution [1]AIM Lab, University of Amsterdam, The Netherlands [2]Inception Institute of Artificial Intelligence, UAE. Correspondence to: Z. Xiao <z.xiao@uva.nl>, X. Zhen <x.zhen@uva.l>, C. Snoek <C.G.M.Snoek@uva.nl>.

2. Methodology

2.1. Preliminaries

In domain generalization, we have $\mathcal{D}=\{D_i\}_{i=1}^{|\mathcal{D}|}=\mathcal{S}\cup\mathcal{T}$ as a set of domains, where $\mathcal{S}$ and $\mathcal{T}$ denote the source and target domains. $\mathcal{S}$ and $\mathcal{T}$ do not have any overlap besides sharing the same label space. Data from the target domains $\mathcal{T}$ is never seen during training. For each domain $D_i \in \mathcal{D}$, we define a joint distribution $p(\mathbf{x},\mathbf{y})$ in the input space $\mathcal{X}$ and the output space $\mathcal{Y}$. We aim to learn a model $f : \mathcal{X}\rightarrow\mathcal{Y}$ in the source domains $\mathcal{S}$ that generalizes well to the target domains $\mathcal{T}$.

We address domain generalization in a probabilistic framework of Bayesian inference by introducing weight uncertainty into neural networks. To be specific, we adopt variational Bayesian inference to learn a neural network that is assumed to be parameterized by weights $\boldsymbol{\theta}$ with a prior distribution $p(\boldsymbol{\theta})$ and the posterior distribution $p(\boldsymbol{\theta}|\mathbf{x},\mathbf{y})$, where $(\mathbf{x},\mathbf{y})$ are samples from the source domain $\mathcal{S}$. To learn the model, we find a variational distribution $q(\boldsymbol{\theta})$ to approximate $p(\boldsymbol{\theta}|\mathbf{x},\mathbf{y})$ by minimizing the Kullback-Leibler divergence $\mathbb{D}_{\mathrm{KL}}\big[q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathbf{x},\mathbf{y})\big]$ between them. This amounts to minimizing the following objective:

$$\mathcal{L}_{\mathrm{Bayes}} = -\mathbb{E}_{q(\boldsymbol{\theta})}[\log p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})] + \mathbb{D}_{\mathrm{KL}}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta})], \quad (1)$$

which is also known as the negative value of the evidence lower bound (ELBO) (Blei et al., 2017). We learn the model in the source domain $\mathcal{S}$ in the hope that it will generalize well to the target domain $\mathcal{T}$.

Kristiadi et al. (2020) show that applying Bayesian approximation to the last layer of a neural network effectively captures model uncertainty. This is also appealing when dealing with uncertainty in domain generalization. In this paper, we incorporate variational Bayesian approximation into the last two network layers to jointly establish domain-invariant feature representations and classifiers. To achieve better domain invariance, we introduce a domain-invariant principle in a probabilistic form under the Bayesian framework. We use ϕ and ψ to denote the parameters of the feature extractor and classifier. We incorporate the domain-invariant principle into the inference of posteriors over ψ and ϕ, which results in our two-layer Bayesian network. Next we detail the Bayesian domain-invariant learning of the network and the objective for optimization.

2.2. Bayesian Domain-Invariant Learning

Existing domain generalizations try to achieve domain invariance by minimizing the distance between intra-class samples from different domains in the hope of reducing the domain gaps (Motiian et al., 2017). However, individual samples cannot be assumed to be representative of the distributions of samples from a certain domain. Therefore, it is
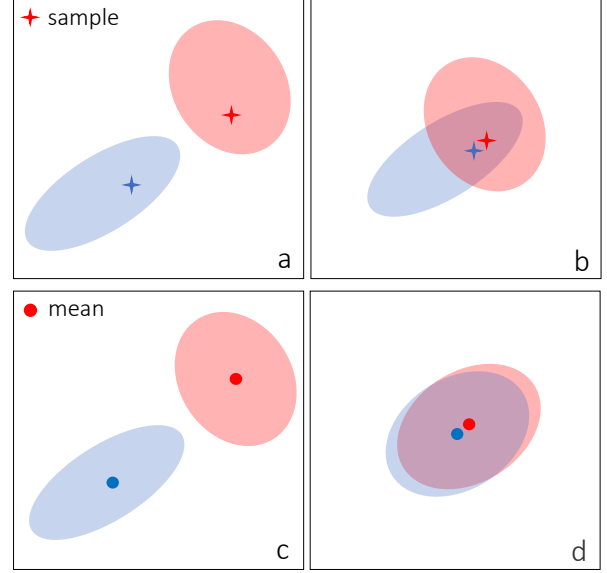

Figure 1. Illustrative contrast between deterministic invariance (top) and probabilistic invariance (bottom), where colors indicate domains. The deterministic invariance tends to minimize the distance between two deterministic samples ($a \rightarrow b$). The samples come from their respective distributions, but with limited distributional awareness. This means that the samples cannot represent the complete distributions. Thus, the deterministic invariance can lead to small distances between samples, but large gaps between distributions, as shown in figure (b). In contrast, the probabilistic invariance directly minimizes the distance between different distributions ($c \rightarrow d$), which yields a better domain invariance as most of the samples in the distributions are taken into account.

preferable to minimize the distributional distance of intra-class samples from different domains, which can directly narrow the domain gap. A contrastive illustration is provided in Figure 1. Motivated by this, we propose to address domain generalization under the probabilistic modeling by incorporating weight uncertainty into invariant learning in the variational Bayesian inference framework.

We first introduce the definition of domain invariance in a probabilistic form, which we adopt to learn domain-invariant feature representations and classifiers in a unified way. We define a continuous domain space $\mathfrak{D}$ containing a set of domains $\{D_i\}_{i=1}^{|\mathcal{D}|}$ in $\mathcal{D}$ and a domain-transform function $g_\zeta(\cdot)$ with parameters ζ in the domain space. $\{D_i\}_{i=1}^{|\mathcal{D}|}$ are discrete samples in $\mathfrak{D}$. Function $g_\zeta(\cdot)$ transforms samples $\mathbf{x}$ from a reference domain to a different domain D_ζ with respect to ζ, where $\zeta\sim q(\zeta)$, and different samples ζ lead to different sample domains $D_\zeta\sim\mathfrak{D}$.

As a concrete example, consider the rotation domains in Rotated MNIST (Ghifary et al., 2015). $\mathfrak{D}$ is the rotation domain space, containing images rotated by continuous angles from 0 to 2π. A sample $\mathbf{x}$ from any domain in $\mathfrak{D}$ can

be transferred into a different domain D_ζ by:

$$g_\zeta(\mathbf{x}) = \begin{bmatrix} cos(\zeta) & -sin(\zeta) \\ sin(\zeta) & -cos(\zeta) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \qquad (2)$$

where $\zeta \sim \text{Uniform}(0, 2\pi)$. ζ is the parameter of $g_\zeta(\cdot)$ and different ζ lead to different rotation angles of the original sample $\mathbf{x}$, which form another domain D_ζ in $\mathfrak{D}$. In practice, transformations between domains are more complicated and the exact forms of $g_\zeta(\cdot)$ and $q(\zeta)$ are not explicitly known.

Based on the above assumptions about $\mathfrak{D}$, $g_\zeta(\cdot)$ and $q(\zeta)$, we introduce our definition of the probabilistic form of domain invariance as follows.

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

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

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 last convolutional layer of a deep neural network.

To adopt the domain-invariant principle in a variational Bayesian inference framework, we reformulate (3) to an expectation form with respect to q_ζ, following Nalisnick & Smyth (2018):

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

According to Definition 2.1, we use the Kullback-Leibler divergence between the two terms in (4), $\mathbb{D}_{\text{KL}}[p_{\boldsymbol{\theta}}(\mathbf{y}_i|\mathbf{x}_i)||\mathbb{E}_{q_\zeta}[p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)]]$, to quantify the domain invariance of the model, which will be zero when the model is domain invariant in the domain space $\mathfrak{D}$. To further facilitate the computation, we derive the upper bound of the KL divergence:

$$\mathbb{D}_{\text{KL}}[p_{\boldsymbol{\theta}}(\mathbf{y}_i|\mathbf{x}_i)||\mathbb{E}_{q_\zeta}[p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)]] \\ \leq \mathbb{E}_{q_\zeta}\Big[\mathbb{D}_{\text{KL}}[p_{\boldsymbol{\theta}}(\mathbf{y}_i|\mathbf{x}_i)||p_{\boldsymbol{\theta}}(\mathbf{y}_\zeta|\mathbf{x}_\zeta)]\Big], \qquad (5)$$

which can be approximated by Monte Carlo sampling as we usually have access to samples from different domains. The complete derivation of (5) is provided in the supplementary materials.

We now adopt the probabilistic domain invariance principle in the variational Bayesian approximation of the last two layers with parameters of ϕ and ψ. The introduced weight uncertainty results in distributional representations $p_\phi(\mathbf{z}|\mathbf{x})$ and predictions $p_\psi(\mathbf{y}|\mathbf{z})$, based on which we derive domain-invariant learning.

Invariant Classifier. For the classifier with parameters ψ and input features $\mathbf{z}$, the probabilistic domain-invariant property in (5) is represented as $\mathbb{E}_{q_\zeta}[\mathbb{D}_{\text{KL}}[p_\psi(\mathbf{y}_i|\mathbf{z}_i)||p_\psi(\mathbf{y}_\zeta|\mathbf{z}_\zeta)]]$. Under the Bayesian framework, the predictive distribution $p_\psi(\mathbf{y}|\mathbf{z})$ is obtained by taking the expectation over the distribution of parameter $\boldsymbol{\theta}$, i.e., $p_\psi(\mathbf{y}|\mathbf{z}) = \mathbb{E}_{q(\psi)}[p(\mathbf{y}|\mathbf{z}, \psi)]$. As the KL divergence is a convex function (Nalisnick & Smyth, 2018), we further extend it to the upper bound:

$$\mathbb{E}_{q_\zeta}\Big[\mathbb{D}_{\text{KL}}[p_\psi(\mathbf{y}_i|\mathbf{z}_i)||p_\psi(\mathbf{y}_\zeta|\mathbf{z}_\zeta)]\Big] \\ = \mathbb{E}_{q_\zeta}\Big[\mathbb{D}_{\text{KL}}[\mathbb{E}_{q(\psi)}[p(\mathbf{y}_i|\mathbf{z}_i, \psi)]||\mathbb{E}_{q(\psi)}[p(\mathbf{y}_\zeta|\mathbf{z}_\zeta, \psi)]]\Big] \\ \leq \mathbb{E}_{q_\zeta}\Big[\mathbb{E}_{q(\psi)}\mathbb{D}_{\text{KL}}[p(\mathbf{y}_i|\mathbf{z}_i, \psi)||p(\mathbf{y}_\zeta|\mathbf{z}_\zeta, \psi)]\Big], \qquad (6)$$

which is tractable with the unbiased Monte Carlo approximation.

In practice, we estimate the expectation over $q(\zeta)$ in an empirical way. Specifically, in each iteration, we choose one domain from the source domains $\mathcal{S}$ as the meta-target domain D_t and the rest are used as the meta-source domains $\{D_s\}_{s=1}^{S}$, where $S = |\mathcal{S}| - 1$. Then we use a batch of samples $\mathbf{x}_s$ from each meta-source domain in the same category as $\mathbf{x}_t$ to approximate $\mathbf{x}_\zeta = g_\zeta(\mathbf{x}_t)$.

Thereby, the domain-invariant classifier is established by minimizing:

$$\frac{1}{SN} \sum_{s=1}^{S} \sum_{i=1}^{N} \mathbb{E}_{q(\psi)}\Big[\mathbb{D}_{\text{KL}}[p(\mathbf{y}_t|\mathbf{z}_t, \psi)||p(\mathbf{y}_s^i|\mathbf{z}_s^i, \psi)]\Big], \qquad (7)$$

where $\{\mathbf{z}_s^i\}_{i=1}^{N}$ are representations of samples $\mathbf{x}_s$ from D_s, which are in the same category as $\mathbf{x}_t$.

The variational Bayesian inference enables us to flexibly incorporate our domain invariant principle into different layers of the neural network. To enhance the domain invariance, we obtain domain-invariant representations by adopting the principle in the penultimate layer.

Invariant Representations. To obtain domain-invariant representations $p(\mathbf{z}|\mathbf{x})$ with feature extractor ϕ, we extend the quantification of our probabilistic domain invariance in (5) to $\mathbb{E}_{q_\zeta}[\mathbb{D}_{\text{KL}}[p_\phi(\mathbf{z}_i|\mathbf{x}_i)||p_\phi(\mathbf{z}_\zeta|\mathbf{x}_\zeta)]]$. Based on the Bayesian layer, the probabilistic distribution of features $p_\phi(\mathbf{z}|\mathbf{x})$ will be a factorized Gaussian distribution if the posterior of ϕ is as well. We illustrate this as follows. Let ϕ be the last Bayesian layer in the feature extractor with a factorized Gaussian posterior and $\mathbf{x}$ be the input feature of ϕ. The posterior of the activation $\mathbf{z}$ is also a factorized

Gaussian (Kingma et al., 2015):

$$q(\phi_{i,j}) \sim \mathcal{N}(\mu_{i,j}, \sigma_{i,j}^2) \quad \forall \phi_{i,j} \in \boldsymbol{\phi}$$
$$\Rightarrow p(z_j|\mathbf{x}, \boldsymbol{\phi}) \sim \mathcal{N}(\gamma_j, \delta_j^2),$$
$$\gamma_j = \sum_{i=1}^N x_i \mu_{i,j}, \quad \text{and} \quad \delta_j^2 = \sum_{i=1}^N x_i^2 \sigma_{i,j}^2, \tag{8}$$

where z_j denotes the j-th element in $\mathbf{z}$, likewise for x_i, and $\phi_{i,j}$ denotes the element at position (i,j) in $\boldsymbol{\phi}$. Thus, we assume the posterior of the last Bayesian layer in the feature extractor has a factorized Gaussian distribution. Then, it is easy to obtain the Bayesian domain invariance of the feature extractor.

In a similar way to (7), we estimate the expectation over $q(\zeta)$ empirically. Therefore, the domain-invariant representations are established by minimizing:

$$\frac{1}{SN} \sum_{s=1}^S \sum_{i=1}^N \left[\mathbb{D}_{\text{KL}} \left[p(\mathbf{z}_t|\mathbf{x}_t, \boldsymbol{\phi}) || p(\mathbf{z}_s^i|\mathbf{x}_s^i, \boldsymbol{\phi}) \right] \right], \tag{9}$$

where $\left\{ \mathbf{x}_s^i \right\}_{i=1}^N$ are from D_s, and denote the samples in the same category as $\mathbf{x}_t$. More details and an illustration of the Bayesian domain-invariant learning are in the supplementary materials.

2.3. Objective Function

Having the Bayesian treatment for the last two layers with respect to parameters $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$, the $\mathcal{L}_{\text{Bayes}}$ in (1) is instantiated as the objective with respect to $\boldsymbol{\psi}$ and $\boldsymbol{\phi}$ as follows:

$$\mathcal{L}_{\text{Bayes}} = -\mathbb{E}_{q(\boldsymbol{\psi})} \left[\mathbb{E}_{q(\boldsymbol{\phi})} [\log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\psi}, \boldsymbol{\phi})] \right]$$
$$+ \mathbb{D}_{\text{KL}}[q(\boldsymbol{\psi})||p(\boldsymbol{\psi})] + \mathbb{D}_{\text{KL}}[q(\boldsymbol{\phi})||p(\boldsymbol{\phi})]. \tag{10}$$

The detailed derivation of (10) is provided in the supplementary materials.

By integrating (7) and (9) into (10), we obtain the Bayesian domain-invariant learning objective as follows:

$$\mathcal{L}_{\text{BIL}} = \frac{1}{L} \sum_{\ell=1}^L \left[\frac{1}{M} \sum_m^M [-\log p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\psi}^{(\ell)}, \boldsymbol{\phi}^{(m)})] \right.$$
$$+ \frac{1}{SN} \sum_{s=1}^S \sum_{i=1}^N \left[\lambda_{\boldsymbol{\psi}} \mathbb{D}_{\text{KL}}[p(\mathbf{y}_t|\mathbf{z}_t, \boldsymbol{\psi}^{(\ell)})||p(\mathbf{y}_s^i|\mathbf{z}_s^i, \boldsymbol{\psi}^{(\ell)})] \right.$$
$$+ \lambda_{\boldsymbol{\phi}} \mathbb{D}_{\text{KL}}[p(\mathbf{z}_t|\mathbf{x}_t, \boldsymbol{\phi})||p(\mathbf{z}_s^i|\mathbf{x}_s^i, \boldsymbol{\phi})]]$$
$$\left. + \mathbb{D}_{\text{KL}}[q(\boldsymbol{\psi})||p(\boldsymbol{\psi})] + \mathbb{D}_{\text{KL}}[q(\boldsymbol{\phi})||p(\boldsymbol{\phi})] \right], \tag{11}$$

where $\lambda_{\boldsymbol{\psi}}$ and $\lambda_{\boldsymbol{\phi}}$ are hyperparameters that control the domain-invariant terms. $\mathbf{x}_t$ and $\mathbf{z}_t$ denote the input and its feature from D_t, and $\mathbf{x}_s^i$ and $\mathbf{z}_s^i$ are from D_s. The

posteriors are set to factorized Gaussian distributions, i.e., $q(\boldsymbol{\psi}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\psi}}, \boldsymbol{\sigma}_{\boldsymbol{\psi}}^2)$ and $q(\boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}, \boldsymbol{\sigma}_{\boldsymbol{\phi}}^2)$. 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 $\epsilon^{(\ell)} \sim \mathcal{N}(0, I)$. Likewise, we draw the Monte Carlo samples $\boldsymbol{\phi}^{(m)}$ for $q(\boldsymbol{\phi})$.

When implementing our Bayesian invariant learning, to increase the flexibility of the prior distribution in our Bayesian layers, we place a scale mixture of two Gaussian distributions as the priors $p(\boldsymbol{\psi})$ and $p(\boldsymbol{\phi})$ (Blundell et al., 2015):

$$\pi \mathcal{N}(0, \boldsymbol{\sigma}_1^2) + (1 - \pi) \mathcal{N}(0, \boldsymbol{\sigma}_2^2), \tag{12}$$

where $\boldsymbol{\sigma}_1$, $\boldsymbol{\sigma}_2$ are set according to (Blundell et al., 2015) and π is chosen by cross-validation. More experiments and analyses on the hyperparameters $\lambda_{\boldsymbol{\psi}}$, $\lambda_{\boldsymbol{\phi}}$ and π are provided in the supplementary materials.

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) augment the data by perturbing the input images with adversarial gradients generated by an auxiliary classifier. Qiao et al. (2020) propose an even more challenging scenario of domain generalization named single domain generalization, which only has one source domain, and introduce an adversarial domain augmentation method that creates "fictitious" yet "challenging" data. Recently, Zhou et al. (2020) employ a generator to synthesize data from pseudo-novel domains to augment the source domains, maximizing the distance between the source and pseudo-novel domains as measured by optimal transport (Peyré et al., 2019).

Another solution for domain generalization involves learning domain-invariant features (D'Innocente & Caputo, 2018; Li et al., 2018b; 2017). Muandet et al. (2013) propose domain-invariant component analysis to learn invariant transformations by minimizing the dissimilarity across domains. Louizos et al. (2015) learn invariant representations by a variational auto-encoder (Kingma & Welling, 2014), introducing Bayesian inference into invariant feature learning. Both Dou et al. (2019) and Seo et al. (2019) achieve a similar goal by introducing two complementary losses and employing multiple normalizations. Li et al. (2019) propose an episodic training algorithm to obtain both a domain-invariant feature extractor and classifier. Zhao et al. (2020) propose entropy-regularization to learn domain-invariant features. Seo et al. (2019) and Zhou et al. (2021) design normalizations to combine different feature statistics and embed the samples into a domain-invariant feature space.

Meta-learning has also been considered for domain general-

ization. Li et al. (2018a) introduce a gradient-based method, i.e., model agnostic meta-learning (Finn et al., 2017), for domain generalization. Balaji et al. (2018) meta-learn a regularization function, making their model robust to domain shifts. Du et al. (2020) propose the meta-variational information bottleneck to learn domain-invariant representations through episodic training.

Gulrajani & Lopez-Paz (2020) find that with careful implementation, empirical risk minimization methods outperform many state-of-the-art models in domain generalization. They claim that model selection is non-trivial for domain generalization and algorithms for this task should specify their own model selection criteria, which is important for the completeness and comparability of the method. In this paper, we implement our method based on the pretrained ResNet-18, as done for our baseline and most methods we compare against in Section 4.

Motiian et al. (2017) previously considered representation alignment across domains in the same class. They propose a classification and contrastive semantic alignment loss based on the L2 distance between deterministic features. Different from them, we exploit Bayesian neural networks to learn domain-invariant representations by minimizing the distance between probabilistic distributions.

Bayesian neural networks have not yet been explored for 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 Bayesian domain-invariant learning combines the representational power of deep neural networks and variational Bayesian inference.

4. Experiments

4.1. Datasets and Settings

We conduct our experiments on four widely used benchmarks for domain generalization and report the average classification accuracy on target domains.

PACS[1] (Li et al., 2017) consists of 9,991 images of seven classes from four domains – *photo*, *art-painting*, *cartoon* and *sketch*. We follow the "leave-one-out" protocol from (Li et al., 2017; 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[2] (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 of-

fice and home environments. We use the same experimental protocol as for PACS.

Rotated MNIST[3] and **Fashion-MNIST**[4] are introduced in (Piratla et al., 2020) for evaluating domain generalization. For 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 are 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 rotations of $0°$ and $90°$. These datasets allow us to demonstrate the generalizability of our model by comparing its performance on in-distribution and out-of-distribution data.

Settings. For all four benchmarks, we employ a ResNet-18 (He et al., 2016) pretrained on ImageNet (Deng et al., 2009) as the backbone. During training we use Adam optimization (Kingma & Ba, 2014) with a learning rate of 0.0001, and train for 10,000 iterations. In each iteration we choose one source domain as the meta-target domain. The batch size is 128. To fit the memory footprint, we choose a maximum number of samples per category and target domain to implement the domain-invariant learning, i.e. sixteen for PACS, Rotated MNIST and Fashion-MNIST, and four for Office-Home. We select λ_ϕ and λ_ψ based on validation set performance and summarize their influence in the supplementary materials. The optimal values of λ_ϕ and λ_ψ are 0.1 and 100. Parameters σ_1 and σ_2 in (12) 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 available.

4.2. Results

We first conduct an ablation study on PACS to investigate the effectiveness of our Bayesian invariant layer for domain generalization. Since the major contributions of this work are the Bayesian treatment and the probabilistic 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, with a corresponding t-SNE visualization of the features learned by various settings in Figure 2, following Du et al. (2020). We also provide a comparison with the state-of-the-art methods on four widely used benchmarks. More results and visualizations are provided in the supplementary materials.

Benefits of Bayesian Invariant Classifier. In Table 1, rows (a) to (d) demonstrate the benefits of the Bayesian invariant classifier. Row (a) serves as our baseline model, which is a vanilla deep convolutional network without any Bayesian

---

[1]https://domaingeneralization.github.io/
[2]https://www.hemanthdv.org/officeHomeDataset.html

[3]http://yann.lecun.com/exdb/mnist/
[4]https://www.kaggle.com/zalando-research/fashionmnist

Table 1. Ablation study on PACS. The "✓" and "×" in the "Bayesian" column indicate whether the classifier ψ and feature extractor ϕ are Bayesian layers or deterministic layers. In the "Invariant" column, they indicate whether the domain-invariant learning is introduced into the classifier and the feature extractor. Note that with "✓" in the Bayesian column, the invariant column denotes Bayesian domain-invariant learning. Otherwise it denotes the deterministic one. The results show that both Bayesian and domain-invariant learning benefit domain generalization, but our Bayesian domain-invariant learning is better. We obtain the best performance with Bayesian domain-invariant learning in both the classifier and feature extractor. We also provide the visualizations of the features for each of the settings in Figure 2.

| ID | Classifier ψ | | Feature extractor ϕ | | PACS | | | | |
	Bayesian	Invariant	Bayesian	Invariant	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)	×	✓	×	×	93.95	80.03	78.03	77.83	82.46
(d)	✓	✓	×	×	95.21	81.25	80.67	79.31	84.11
(e)	×	×	✓	×	92.81	78.66	77.90	78.72	82.02
(f)	×	×	×	✓	94.17	79.75	79.51	78.31	82.94
(g)	×	×	✓	✓	95.15	80.96	79.57	79.15	83.71
(h)	✓	×	✓	×	93.83	82.13	79.18	79.03	83.54
(i)	×	✓	×	✓	94.12	80.52	80.39	78.53	83.39
(j)	✓	✓	✓	✓	**95.97**	**83.92**	**81.61**	**80.31**	**85.45**

treatment or domain-invariant loss. The backbone is also a ResNet-18 pretrained on ImageNet. Rows (b), (c) and (d) show the performance with a Bayesian classifier, a deterministic domain-invariant classifier and our Bayesian-invariant classifier. By comparing (b) and (a), it is clear that the Bayesian treatment for the classifier improves the performance, especially in the "art-painting" and "sketch" domains. The deterministic invariant property, as shown in row (c), also benefits the performance. Nevertheless, our Bayesian invariant learning based on the probabilistic framework performs better. The results demonstrate that the Bayesian invariant learning enhances the robustness of Bayesian neural networks on out-of-distribution data and better leverages domain-invariance than the deterministic invariant model.

The three subfigures in the first row of Figure 2 also demonstrate the benefits of the Bayesian invariant classifier. The Bayesian treatment enlarges the inter-class distance in all domains, as shown in Figure 2 (b). The Bayesian layer incorporates weight uncertainty into the predictions and improves their diversity, which enhances the classification performance on the source domains as well as the generalization to the target domain. The deterministic invariant classifier tends to minimize the distance between samples with the same label. However, the effect is mainly on the source domain, without obvious impact on the target domain (pink samples), as shown in Figure 2 (c). By introducing uncertainty into both the classification and domain-invariant procedure, our Bayesian domain-invariant classifier further enlarges the inter-class distance in all domains and better generalizes the domain-invariant property to the target domain, as comparing Figure 2 (d) to (b) and (c).

Benefits of Bayesian Invariant Feature Extractor. The benefit of our Bayesian domain-invariant principle for the feature extractor is demonstrated in rows (e), (f), and (g) of Table 1. Similar to the classifier, introducing Bayesian inference into the feature extractor (comparing row (e) with row (a)) brings a good accuracy improvement. By comparing (g) to (e) and (f), we observe that our Bayesian invariant layer achieves consistently better accuracy than the deterministic invariant model and regular Bayesian layer.

The subfigures in the second row of Figure 2 further demonstrate the effectiveness of the Bayesian domain-invariant feature extractor. Similar to the classifier, introducing the Bayesian framework into the feature extractor also enlarges the inter-class distance of all domains by introducing model uncertainty, as shown in (e). The deterministic invariant feature extractor can also minimize the intra-class distance between different domains. However, while the effect is more obvious than the deterministic domain-invariant classifier, it is still poor on the target domain (Figure 2 (f)). Compared to (e) and (f), the Bayesian invariant feature extractor in (g) tends to further maximize the inter-class distance while minimizing the intra-class distance between different domains, which obviously enlarges the inter-class distance between samples from the target domain. Since the Bayesian invariant feature extractor achieves domain invariance in a probabilistic way, more uncertainties are taken into account, enabling the model to achieve better generalization on the target domain. Note that although the Bayesian invariant classifier in (d) and feature extractor in (g) achieve domain-invariant learning from different directions of the feature space, they both improve the performance on the target domain and do not conflict with each other.
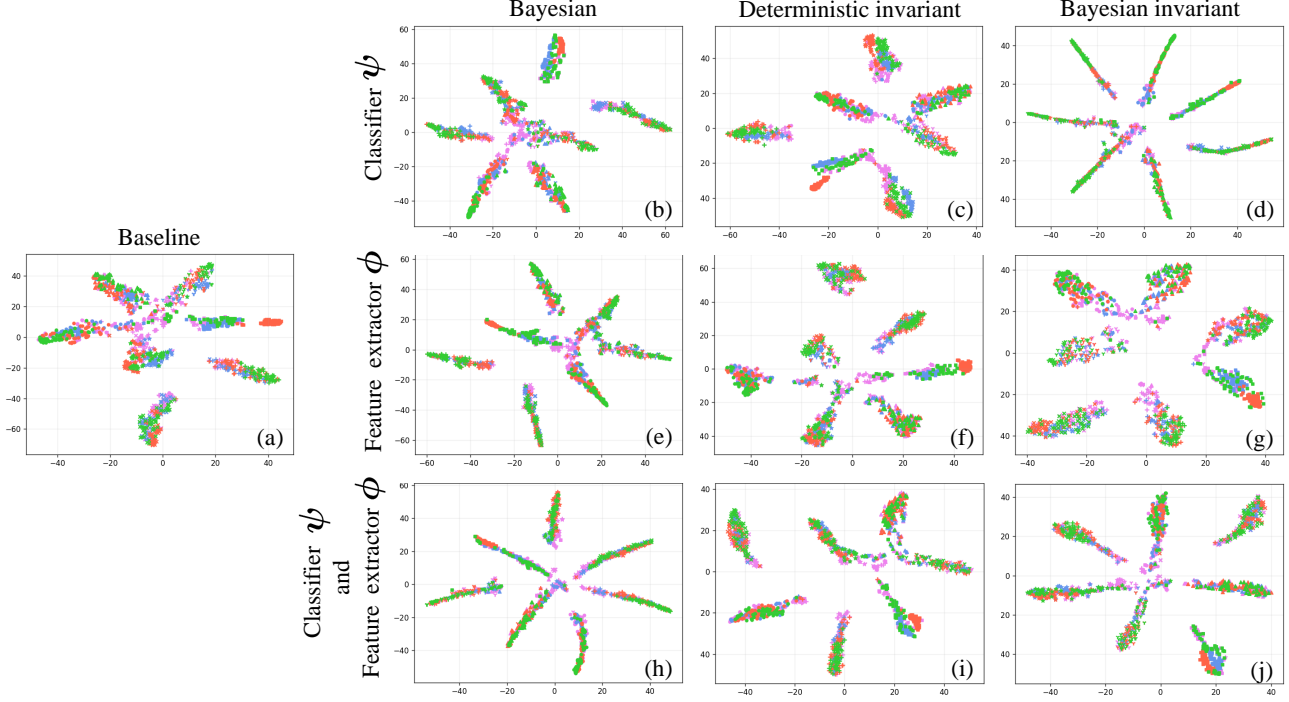
Figure 2. Illustration of the benefit of Bayesian domain-invariant learning by visualization of the feature representations. Colors denote domains, where the target domain "art-painting" is in violet, and shapes indicate classes. Our Bayesian domain-invariant learning (shown in the right column), especially when employed in both the classifier and feature extractor ((j)), achieves better results than the other cases shown in the ten subfigures, which correspond to the ten settings in Table 1 (identified by ID).

Benefits of Synergistic Bayesian Invariant Learning.
The last three rows of Table 1 show the performance when introducing Bayesian *and* invariant learning into both the classifier and feature extractor. Both the Bayesian learning (row (h)) and deterministic invariant learning (row (i)) in two layers perform better than introducing the corresponding properties into only one layer of the model (compared with (b), (c) and (e), (f)) on most domains, as well as in terms of mean performance. Overall, employing our Bayesian invariant layer in both the classifier and feature extractor achieves the best performance, as shown in row (j).

The last row in Figure 2 further demonstrates the benefits of synergistic Bayesian invariant learning. Incorporating the Bayesian classifier and the Bayesian feature extractor further maximizes the inter-class distance of all domains, as comparing (h) to (b) and (e). Introducing deterministic invariance into both the classifier and feature extractor also improves the domain-invariant property of features and the generalization to the target domain to some extent, as shown in subfigure (i). Moreover, by utilizing both the Bayesian domain-invariant classifier and Bayesian domain-invariant feature extractor, our method combines their benefits shown in (d) and (g) and achieves the best performance, as shown in (j). This indicates the benefits of the synergy between Bayesian inference and probabilistic domain-invariant learning for domain generalization.

Table 2. Effect of more Bayesian layers on PACS. The "Bayesian" and "Invariant" columns indicate whether the penultimate layer ϕ' in the feature extractor has a Bayesian and/or domain-invariant property. More Bayesian layers benefit the performance while excessive domain-invariant learning is harmful.

ϕ'		PACS				
Bayesian	Invariant	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

Effect of More Bayesian Layers. We also experiment with more Bayesian layers in the feature extractor, as shown in Table 2. The settings of the model in the first row are the same as row (j) in Table 1. When introducing another Bayesian layer ϕ' without domain invariance into the model, as shown in the second row, the average performance improves slightly. However, if we introduce the Bayesian domain-invariant learning into ϕ' (third row), the overall performance deteriorates slightly. This may due to the loss of information in the features caused by the excessive use of domain-invariant learning. In addition, due to the Bayesian inference and Monte Carlo sampling, more Bayesian layers leads to higher memory usage and more computations (more detailed discussion in supplementary materials). As such, we prefer to apply the Bayesian invariant learning only in 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
Carlucci et al. (2019)	96.03	79.42	75.25	71.35	80.51
Dou et al. (2019)	94.99	80.29	77.17	71.69	81.04
Zhao et al. (2020)	**96.65**	80.70	76.40	71.77	81.38
Piratla et al. (2020)	94.10	78.90	75.80	76.70	81.40
Chattopadhyay et al. (2020)	93.35	76.90	80.38	75.21	81.46
Li et al. (2019)	93.90	82.10	77.00	73.00	81.50
Balaji et al. (2018)	95.50	83.70	77.20	70.30	81.68
Zhou et al. (2020)	96.20	83.30	78.20	73.60	82.83
Seo et al. (2019)	95.87	**84.67**	77.65	**82.23**	85.11
Huang et al. (2020)	95.99	83.43	80.31	80.85	85.15
This paper	95.97	83.92	**81.61**	80.31	**85.45**

State-of-the-Art Comparisons. We compare our method with the state-of-the-art on four datasets. For comprehensive comparison, we also include a vanilla deep convolutional ResNet-18 network as baseline, without any Bayesian treatment, as done in row (a) of Table 1. The results for each dataset are reported in Tables 3, 4 and 5.

On PACS, in Table 3, our method achieves the best mean accuracy. For each individual domain, we are competitive with the state-of-the-art and even exceed all other methods on the "cartoon" domain. On Office-Home, in Table 4, we again achieve the best mean accuracy. It is worth mentioning that on the most challenging "art" and "clipart" domains, we also deliver the highest accuracy, with a good improvement over previous methods. However, Zhou et al. (2020) and Seo et al. (2019) outperform the proposed model on some domains of PACS and Office-Home. In (Zhou et al., 2020), the source domains are augmented by a generator that synthesizes data from pseudo-novel domains, which often have similar characteristics with the source data. This pays off when the target data also has similar characteristics to the source domains, as the pseudo domains are more likely to cover the target domain, as can be seen for "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. We highlight that our method generates domain-invariant representations and classifiers, resulting in competitive results across all domains and overall. In addition, Seo et al. (2019) combine batch and instance normalization for domain generalization. This tactic is effective on PACS, but less so on Office-Home. We attribute this to the larger number of categories in Office-Home, where instance normalization is known to make features less discriminative with respect to object categories (Seo et al., 2019). In contrast, our Bayesian domain-invariant learning learns domain-invariant features and predictions in a probabilistic form by introducing uncertainty into the model, resulting in good performance on both PACS and Office-Home.

Table 4. Comparison on Office-Home. Our method 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
Carlucci et al. (2019)	53.04	47.51	71.47	72.79	61.20
Li et al. (2018b)	56.50	47.30	72.10	74.80	62.68
Seo et al. (2019)	59.37	45.70	71.84	74.68	62.90
Huang et al. (2020)	58.42	47.90	71.63	74.54	63.12
Zhou et al. (2020)	60.60	50.10	**74.80**	**77.00**	65.63
This paper	**61.81**	**53.27**	74.27	76.31	**66.42**

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

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

On Rotated MNIST and Fashion-MNIST, 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 method achieves the best performance on both sets of the two datasets. In particular, our method 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 Figure 2.

5. Conclusion

In this work, we propose a variational Bayesian learning framework for domain generalization. We introduce Bayesian neural networks into the model, which are 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 method combines the representational power of deep neural networks and uncertainty modeling ability of Bayesian learning, demonstrating effectiveness for domain generalization. Ablation studies further validate these benefits. Our Bayesian invariant learning sets a new state-of-the-art on four domain generalization benchmarks.

References

Atanov, A., Ashukha, A., Struminsky, K., Vetrov, D., and Welling, M. The deep weight prior. In *International Conference on Learning Representations*, 2019.

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

Bengio, Y., Deleu, T., Rahaman, N., Ke, N. R., Lachapelle, S., Bilaniuk, O., Goyal, A., and Pal, C. A meta-transfer objective for learning to disentangle causal mechanisms. In *International Conference on Learning Representations*, 2019.

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

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. *ArXiv*, abs/1505.05424, 2015.

Carlucci, F. M., D'Innocente, A., Bucci, S., Caputo, B., and Tommasi, T. Domain generalization by solving jigsaw puzzles. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2229–2238, 2019.

Chattopadhyay, P., Balaji, Y., and Hoffman, J. Learning to balance specificity and invariance for in and out of domain generalization. *arXiv preprint arXiv:2008.12839*, 2020.

Daxberger, E. and Hernández-Lobato, J. M. Bayesian variational autoencoders for unsupervised out-of-distribution detection. *arXiv preprint arXiv:1912.05651*, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Dou, Q., de Castro, D. C., Kamnitsas, K., and Glocker, B. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, pp. 6450–6461, 2019.

Du, Y., Xu, J., Xiong, H., Qiu, Q., Zhen, X., Snoek, C. G., and Shao, L. Learning to learn with variational information bottleneck for domain generalization. *arXiv preprint arXiv:2007.07645*, 2020.

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

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

Ghifary, M., Bastiaan Kleijn, W., Zhang, M., and Balduzzi, D. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pp. 2551–2559, 2015.

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Huang, Z., Wang, H., Xing, E. P., and Huang, D. Self-challenging improves cross-domain generalization. *arXiv preprint arXiv:2007.02454*, 2020.

Ilse, M., Tomczak, J. M., Louizos, C., and Welling, M. Diva: Domain invariant variational autoencoders. *arXiv preprint arXiv:1905.10427*, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Kingma, D. P., Salimans, T., and Welling, M. Variational dropout and the local reparameterization trick. In *Advances in neural information processing systems*, pp. 2575–2583, 2015.

Kristiadi, A., Hein, M., and Hennig, P. Being bayesian, even just a bit, fixes overconfidence in relu networks. *arXiv preprint arXiv:2002.10118*, 2020.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Priol, R. L., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex). *arXiv preprint arXiv:2003.00688*, 2020.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5542–5550, 2017.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Learning to generalize: Meta-learning for domain generalization. In *AAAI Conference on Artificial Intelligence*, 2018a.

Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.-Z., and Hospedales, T. M. Episodic training for domain generalization. *IEEE/CVF International Conference on Computer Vision*, pp. 1446–1455, 2019.

Li, H., Jialin Pan, S., Wang, S., and Kot, A. C. Domain generalization with adversarial feature learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5400–5409, 2018b.

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

Louizos, C., Swersky, K., Li, Y., Welling, M., and Zemel, R. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*, 2015.

MacKay, D. J. The evidence framework applied to classification networks. *Neural computation*, 4(5):720–736, 1992.

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

Motiian, S., Piccirilli, M., Adjeroh, D. A., and Doretto, G. Unified deep supervised domain adaptation and generalization. In *IEEE International Conference on Computer Vision*, pp. 5715–5725, 2017.

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pp. 10–18, 2013.

Nalisnick, E. and Smyth, P. Learning priors for invariance. In *International Conference on Artificial Intelligence and Statistics*, pp. 366–375, 2018.

Nguyen, A., Yosinski, J., and Clune, J. 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.

Nguyen, C. V., Li, Y., Bui, T. D., and Turner, R. E. Variational continual learning. In *International Conference on Learning Representations*, 2018.

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

Piratla, V., Netrapalli, P., and Sarawagi, S. Efficient domain generalization via common-specific low-rank decomposition. *arXiv preprint arXiv:2003.12815*, 2020.

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

Seo, S., Suh, Y., Kim, D., Han, J., and Han, B. Learning to optimize domain specific normalization for domain generalization. *arXiv preprint arXiv:1907.04275*, 2019.

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

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5018–5027, 2017.

Volpi, R., Namkoong, H., Sener, O., Duchi, J. C., Murino, V., and Savarese, S. Generalizing to unseen domains via adversarial data augmentation. In *Advances in neural information processing systems*, pp. 5334–5344, 2018.

Wang, H., He, Z., Lipton, Z. C., and Xing, E. P. Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations*, 2018.

Zhao, S., Gong, M., Liu, T., Fu, H., and Tao, D. Domain generalization via entropy regularization. *Advances in Neural Information Processing Systems*, 33, 2020.

Zhou, K., Yang, Y., Hospedales, T., and Xiang, T. Learning to generate novel domains for domain generalization. *arXiv preprint arXiv:2007.03304*, 2020.

Zhou, K., Yang, Y., Qiao, Y., and Xiang, T. Domain generalization with mixstyle. In *International Conference on Learning Representations*, 2021.