Machine Learning
https://doi.org/10.1007/s10994-022-06256-y

# Meta-learning the invariant representation for domain generalization

**Chen Jia**[1] · **Yue Zhang**[2,3]

Received: 3 June 2022 / Revised: 7 August 2022 / Accepted: 19 September 2022


**Abstract**
Domain generalization studies how to generalize a machine learning model to unseen distributions. Learning invariant representation across different source distributions has been shown high effectiveness for domain generalization. However, the intrinsic possibility of overfitting in source domains can limit the generalization of invariance when faced with a target domain with large discrepancy to the source domains. To address this problem, we propose a meta-learning algorithm via bilevel optimization for domain generalization, where the inner-loop objective aims to minimize the discrepancy across different source domains while the outer-loop objective aims to minimize the discrepancy between source domains and a potential target domain. We show from a geometric perspective that the proposed algorithm can improve out-of-domain robustness for invariance learning. Empirically, we evaluate on five datasets and achieve the best results among a range of strong domain generalization baselines.

**Keywords** Domain generalization · Meta-learning · Invariance learning · Transfer learning

## 1 Introduction

Deep learning has achieved highly competitive performance on test data drawn from the same distribution as large training data. However, in practice, it is almost impossible to ensure that test data strictly follow source distributions. *Domain generalization* (DG) investigates how to generalize a hypothesis learned from source domains to unseen target domains (Blanchard et al., 2011; Muandet et al., 2013).

Editiors: Yu-Feng Li, Prateek Jain.

✉ Chen Jia
jiachen@westlake.edu.cn

Yue Zhang
zhangyue@westlake.edu.cn

[1] Fudan University, Shanghai, China

[2] School of Engineering, Westlake University, Hangzhou, China

[3] Institute of Advanced Technology, Westlake Institute for Advanced Study, Hangzhou, China

Published online: 28 October 2022

As a seminal method, the empirical risk minimization (ERM) algorithm (Vapnik, 1999) aims to learn a hypothesis that achieves the minimum empirical risk on all the source domains (Gulrajani & Lopez-Paz, 2020). Although the ERM algorithm has achieved promising results on DG (Gulrajani & Lopez-Paz, 2020), previous work have shown from both theoretical and empirical perspectives that the performance of ERM can be largely relayed on the number of source domains and the diversity of source samples (Li et al., 2022; Gulrajani & Lopez-Paz, 2020).

Recent DG work explores an invariance learning approach to alleviate the prediction gap that arises from the distributional diversity across different domains (Li et al., 2018a, b; Zhang et al., 2021). Such approach aims to obtain an invariant representation by training the feature embedding using discrepancy-based losses, which estimate discrepancy metrics on covariate shifts w.r.t. marginal feature distributions (Albuquerque et al., 2020) or conditional shifts w.r.t. conditional feature distributions (Zhang et al., 2021; Shui et al., 2022). Further, previous work has shown that an invariance of the excess risk across domains is equivalent to the invariance of representation (Zhang et al., 2021). Although the invariant feature learning can ensure prediction invariance across domains, the intrinsic distribution gap between the source and target domains and the possibility of overfitting in source domains can badly affect the generalization performance, as shown in Fig. 1b.

We improve the out-of-domain robustness for invariance learning via a bilevel meta-learning algorithm to learn more robust invariant representation across different domains. In particular, we follow the previous work to use an episodic training process (Li et al., 2018c), i.e., randomly extracting some meta-source domains for training and a meta-target domain for test from all the source domains as a meta-task to simulate domain shift.

## 1.1 Approach

We consider a learning algorithm for the feature embedding with meta-parameters, denoted as $A^f_{\phi}(\cdot)$. Then, a bilevel meta-learning algorithm (Finn et al., 2017) is used to learn the parameter initialization $\phi$, where the inner-loop objective aims to minimize the discrepancy across different meta-source domains while the outer-loop objective aims to minimize

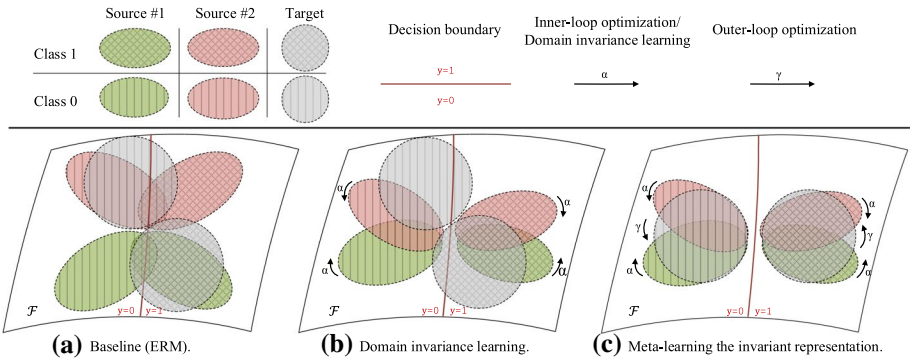



**Fig. 1** Illustration of our approach. Compared with the ERM baseline (**a**), domain invariance learning (**b**) reduces the discrepancy across source domains and performs well on source-domain classification, but it may still have big error on the target domain. Our approach (**c**) uses bilevel meta-learning to further reduce the discrepancy between the target domain and source domains, such that a hypothesis learned from the source domains can generalize to the target domain

the discrepancy between the meta-target and meta-source domains. Intuitively, the effectiveness of such bilevel meta-learning algorithm is shown in Fig. 1c.

### 1.2 Results

We formulate a geometric understanding for the bilevel meta-learning algorithm and show its effectiveness to minimize the intrinsic domain discrepancy, which is formulated as the $\mathcal{Y}$-discrepancy (Zhang et al., 2012) between the target domain and a convex hull of source domains. Empirically, we follow the training and evaluation protocol by Gulrajani and Lopez-Paz (2020) and conduct experiments on five datasets. Results show that our approach can effectively learn the domain invariance and achieve the best performance compared with a range of ERM, invariance learning and meta-learning algorithms. The code is released at https://github.com/jiachenwestlake/MLIR.

## 2 Related work

Domain generalization (DG) has become a popular field and achieved promising results in recent years. We review the most related DG work as follows.

### 2.1 Domain-invariance learning

Early DG work performs kernel-based approaches to learn an invariant feature mapping to the reproducing kernel Hilbert space (RKHS) (Muandet et al., 2013). Neural methods have achieved promising results in recent years, and invariant representation learning has become a strong approach for DG. Roughly speaking, such approach uses an additional loss w.r.t. a discrepancy measure across different source domains, which can employ maximum mean discrepancy (Li et al., 2018a), $\mathcal{H}$-divergence (Li et al., 2018b; Albuquerque et al., 2020), KL-divergence (Xiao et al., 2021), $\mathcal{Y}$-discrepancy (Zhang et al., 2021) and total variation distance (Shui et al., 2022). Furthermore, DMG (Chattopadhyay et al., 2020) learns a balance between invariant and specific representation; REG (Shui et al., 2022) uses regularization to improve the smoothness of representation. In contrast to these work, we aim to improve the robustness of invariant learning via meta-learning. Our work can be seen as an extension to the line of work (Zhang et al., 2021) with a meta-learning approach, which has shown the equivalence between transferability and $\mathcal{Y}$-discrepancy across different domains. Other invariance learning approaches such as IRM (Arjovsky et al., 2019) learns the labeling invariance across different domains, which is orthogonal to this work.

### 2.2 Meta-learning

Meta-learning provides a framework to gain experience for future tasks over multiple training episodes, which has been introduced to address DG via simulating domain shift (Li et al., 2018c; Balaji et al., 2018; Dou et al., 2019). An early approach is MLDG (Li et al., 2018c), which uses bilevel meta-learning (Finn et al., 2017) to train a model on source

domains such that it generalizes to the target domain. MetaReg Balaji et al. (2018) learns a regularization on the classifier such that a classifier trained on source domains can generalize to target domain. These work have a common limitation that uses task objectives directly as the inner-loop and outer-loop objectives, which can be suboptimal, since it is highly abstracted from the feature representation. To address this problem, we focus on a meta-learning approach to reduce the discrepancy between the target domain and sources domains. In particular, we build a bilevel meta-learning procedure on the first-order MAML framework (Finn et al., 2017), which achieves highly computational efficiency while also preserving the accuracy. To our knowledge, we are the first to use meta-learning for invariance learning.

### 2.3 Convex domain combination

A closely related problem is in multiple-source domain adaptation, where the target domain is assumed to be a convex combination of source domains, but the weights can be unknown. Previous work (Mansour et al., 2008; Hoffman et al., 2018; Shao et al., 2021) assume that there exists pretrained hypothesis for each source domain and have well-studied how to combine the source hypotheses to derive a target hypothesis. Such work also indicate that simple linear combinations face difficulties due to the discrepancy across different source domains. In contrast to these work, DG often assumes that source-domain data are available for training, which can be used to learn an invariant representation to break the limitation of domain discrepancy for convex combination (Shao et al., 2021). Furthermore, we study a more general setting, where the target domain can be outside the convex hull of source domains. Accordingly, we propose a meta-learning approach to reduce the discrepancy between the target domain and the convex hull of source domains.

## 3 Preliminaries

### 3.1 Notations

Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ be the output space. Following previous work (Blanchard et al., 2011; Muandet et al., 2013), we define a domain as a joint distribution on Cartesian product of the input and output space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and let $\mathfrak{P}$ denote the set of all domains. We denote the set of $N$ source domains as $\mathcal{S} = \{\mathbb{S}^i\}_{1 \le i \le N}$. The corresponding set of training samples is denoted as $\hat{\mathcal{S}} = \{\hat{\mathbb{S}}^i\}_{1 \le i \le N}$, where the training sample for the $i$-th domain is denoted as $\hat{\mathbb{S}}^i = \{(x_k^i, y_k^i)\}_{1 \le k \le n_i}$ with cardinality $n_i$ and assuming that $(x_k^i, y_k^i) \overset{i.i.d.}{\sim} \mathbb{S}^i$. For brevity, we assume that all domains have the equal sample size, i.e., $n_1 = \ldots = n_N = n$.

A hypothesis $h \in \mathcal{H} : \mathcal{X} \to \mathcal{Y}$ is defined as a mapping from the input space to the output space. The associated error of a hypothesis $h$ at a data point $(x, y)$ is defined as $\ell(h(x), y)$. Given a domain $\mathbb{S}$ and its corresponding sample $\hat{\mathbb{S}} = \{(x_i, y_i)\}_{1 \le i \le n}$, the expected error and the empirical error are defined as $\epsilon_{\mathbb{S}}(h) = \mathbb{E}_{(x,y)\sim\mathbb{S}} \ell(h(x), y)$ and $\hat{\epsilon}_{\hat{\mathbb{S}}}(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i)$, respectively. In this work, we consider $h$ to be a neural network and decompose $h$ into a feature embedding $f_\psi \in \mathcal{F} : \mathcal{X} \to \mathbb{R}^d$, parameterized by $\psi$ (or $f$ for brevity) and a task

classifier $g_{\theta} \in \mathcal{G} : \mathbb{R}^d \to \mathcal{Y}$, parameterized by $\theta$ (or $g$ for brevity), i.e., $h = g_{\theta} \circ f_{\psi}$. Furthermore, this work is interested in a learning algorithm for the feature embedding $A^f_{\phi} : \bigcup_{N=1}^{\infty} \mathcal{Z}^{N \times n} \to \mathcal{F}$, with the meta-parameter $\phi \in \mathbf{\Phi}$, mapping from source-domain training samples to a feature embedding. Given source-domain training samples $\hat{\mathcal{S}}$, the hypothesis can therefore be represented as $g \circ A^f_{\phi}(\hat{\mathcal{S}})$.

### 3.2 Meta-learning for domain generalization

The main idea is to use a sequence of $M$ pairs of meta-training and meta-test samples $\{(\hat{\mathcal{D}}_i^{tr}, \hat{\mathcal{D}}_i^{te})\}_{1 \le i \le M}$ to improve the ability of an algorithm for tackling domain shift. To make connections with the standard meta-learning formulations (Baxter, 2000; Chen et al., 2020), each meta-sample $(\hat{\mathcal{D}}_i^{tr}, \hat{\mathcal{D}}_i^{te})$ can be seen as a pair of Qury/Support sets of a DG task, where for each $i \in [M]$, $\hat{\mathcal{D}}_i^{tr}$ denotes meta-training samples from a set of meta-source domains and $\hat{\mathcal{D}}_i^{te}$ denotes the meta-test sample from a meta-target domain which should not belong to any meta-source domain. In practice, an episodic training process is used to construct the meta-sample with training samples from $N$ source domains. In each training iteration, each domain can become the meta-target domain and the rest are served as the meta-source domains. Thus, the meta-sample $\{(\hat{\mathcal{D}}_i^{tr}, \hat{\mathcal{D}}_i^{te})\}_{1 \le i \le M}$ is defined as:

$$\left\{(\hat{\mathcal{D}}_i^{tr}, \hat{\mathcal{D}}_i^{te})\right\}_{1 \le i \le M} := \left\{\left(\{\hat{\mathbb{S}}^j\}_{j \ne i}, \hat{\mathbb{S}}^i\right) : 1 \le i \le N\right\} \tag{1}$$

### 3.3 Domain discrepancy

$\mathcal{Y}$-discrepancy has been used for domain invariance learning (Zhang et al., 2012, 2021). For convenience in presentation, we extend the hypothesis in the original definition (Zhang et al., 2012) to a learning algorithm for feature embedding.

**Definition 1** *($\mathcal{Y}$-discrepancy):* Let $g \in \mathcal{G}$ be the classifier and $A^f_{\phi}(\hat{\mathcal{S}})$ be the feature embedding learned from source samples $\hat{\mathcal{S}}$, then the $\mathcal{Y}$-discrepancy $\mathrm{disc}(\mathbb{S}, \mathbb{T})$ between two domains $\mathbb{S}$ and $\mathbb{T}$ and its empirical version $\hat{\mathrm{disc}}(\hat{\mathbb{S}}, \hat{\mathbb{T}})$ w.r.t. the corresponding samples $\hat{\mathbb{S}}$ and $\hat{\mathbb{T}}$ are defined as:

$$\begin{aligned} \mathrm{disc}\left(\mathbb{S}, \mathbb{T}\right) &:= \sup_{g \in \mathcal{G}} \left| \epsilon_{\mathbb{S}}\left(g \circ A^f_{\phi}(\hat{\mathcal{S}})\right) - \epsilon_{\mathbb{T}}\left(g \circ A^f_{\phi}(\hat{\mathcal{S}})\right) \right|; \\ \hat{\mathrm{disc}}\left(\hat{\mathbb{S}}, \hat{\mathbb{T}}\right) &:= \sup_{g \in \mathcal{G}} \left| \hat{\epsilon}_{\hat{\mathbb{S}}}\left(g \circ A^f_{\phi}(\hat{\mathcal{S}})\right) - \hat{\epsilon}_{\hat{\mathbb{T}}}\left(g \circ A^f_{\phi}(\hat{\mathcal{S}})\right) \right|. \end{aligned} \tag{2}$$

It is clear that $\mathcal{Y}$-discrepancy defines a pseudo-distance between a pair of domains in that it satisfies symmetry and the triangle inequality but not satisfies identity of indiscernibility since $\mathrm{disc}(\mathbb{S}, \mathbb{T}) = 0 \not\Rightarrow \mathbb{S} = \mathbb{T}$. It can measure not only covariate shift between domains, but also conditional shift between domains (Zhang et al., 2012). Therefore, we choose $\mathcal{Y}$-discrepancy as a measurement for domain discrepancy for the proposed algorithm.

## 4 Approach

The goal of our algorithm is to reduce the $\mathcal{Y}$-discrepancy between the source domains and the target domain. We present a specific meta-learning algorithm.

### 4.1 Meta-learning via bilevel optimization

We focus on a bilevel meta-learning framework (Finn et al., 2017), which uses the meta-sample to learn a meta-parameter $\boldsymbol{\phi}^* \in \boldsymbol{\Phi}$ for a learning algorithm $A^f_{\boldsymbol{\phi}^*}(\cdot)$. Such learning algorithm can use the source samples $\hat{\mathcal{S}}$ for optimizing the feature embedding, represented as $A^f_{\boldsymbol{\phi}^*}(\cdot) : \hat{S} \mapsto f_{\boldsymbol{\psi}^*}$, where $f_{\boldsymbol{\psi}^*}$ denotes the feature embedding parameterized by the optimized parameter $\boldsymbol{\psi}^*$. For notation convenience, we will sometimes treat $f_{\boldsymbol{\psi}^*}$ and $\boldsymbol{\psi}^*$ equivalently to represent the learned feature embedding.

In this work, the meta-learner optimizes the meta-parameter $\boldsymbol{\phi}$ to minimize $\mathcal{Y}$-discrepancy between the meta-target domain and meta-source domains (will be defined in Eq. 4), such that the learned algorithm optimizes the parameter of feature embedding to minimize the $\mathcal{Y}$-discrepancy across different meta-source domains (will be defined in Eq. 5). We formally define the bilevel optimization problem as follows.

**Definition 2** *(Bilevel Optimization)* We denote the outer-loop and inner-loop objectives w.r.t. the feature embedding as $\hat{\mathcal{L}}_{out}$ and $\hat{\mathcal{L}}_{in}$, respectively. Let $A^f_{\boldsymbol{\phi}}(\cdot)$ be a learning algorithm parameterized by $\boldsymbol{\phi}$ for the inner-loop optimization. Given a meta-sample $\{(\hat{\mathcal{D}}^{tr}_i, \hat{\mathcal{D}}^{te}_i)\}_{1\le i\le M}$ defined in Eq. 1, the bilevel optimization problem is defined as:

$$\begin{aligned} &\text{Outer-loop:} \quad \boldsymbol{\phi}^* \in \arg\min_{\boldsymbol{\phi}\in\boldsymbol{\Phi}} \sum_{i\in[M]} \hat{\mathcal{L}}_{out}\big(A^f_{\boldsymbol{\phi}}(\hat{\mathcal{D}}^{tr}_i), (\hat{\mathcal{D}}^{tr}_i, \hat{\mathcal{D}}^{te}_i)\big); \\ &\text{Inner-loop:} \quad A^f_{\boldsymbol{\phi}}(\hat{\mathcal{D}}^{tr}_i) \in \arg\min_{\boldsymbol{\psi}\in\mathcal{C}(\boldsymbol{\phi})} \hat{\mathcal{L}}_{in}\big(\boldsymbol{\psi}; \hat{\mathcal{D}}^{tr}_i\big), \end{aligned} \tag{3}$$

where $\mathcal{C}(\boldsymbol{\phi})$ denotes the constrained parameter space of $\boldsymbol{\psi}$ by $\boldsymbol{\phi}$, which will be specified in the next section. Let $\boldsymbol{\psi}^*_i$ denote $\boldsymbol{\psi}^*_i := A^f_{\boldsymbol{\phi}}(\hat{\mathcal{D}}^{tr}_i)$, the empirical objectives in the outer-loop and inner-loop are defined as follows:

$$\hat{\mathcal{L}}_{out}\Big(\boldsymbol{\psi}^*_i; (\hat{\mathcal{D}}^{tr}_i, \hat{\mathcal{D}}^{te}_i)\Big) := \sum_{\hat{\mathbb{S}}^k_i \in \hat{\mathcal{D}}^{tr}_i} \hat{\mathrm{disc}}_{\mathcal{Y}}\Big(f_{\boldsymbol{\psi}^*_i}\Big(\hat{\mathcal{D}}^{te}_i, \hat{\mathbb{S}}^k_i\Big)\Big); \tag{4}$$

$$\hat{\mathcal{L}}_{in}\Big(\boldsymbol{\psi}; \hat{\mathcal{D}}^{tr}_i\Big) := \sum_{\hat{\mathbb{S}}^k_i, \hat{\mathbb{S}}^t_i \in \hat{\mathcal{D}}^{tr}_i} \hat{\mathrm{disc}}_{\mathcal{Y}}\big(f_{\boldsymbol{\psi}}\big(\hat{\mathbb{S}}^k_i, \hat{\mathbb{S}}^t_i\big)\big). \tag{5}$$

### 4.2 Gradient-based meta-learning algorithm

**Algorithm 1** Meta-training.

**Input data:** $N$ source-domain training samples, hyperparameters: $\eta$, $\alpha$, $\gamma$.
**Parameters:** feature embedding $f_{\boldsymbol{\psi}}$, adversarial classifiers $\mathcal{G} = \{g_{\boldsymbol{\theta}_{ij}}\}_{1\le i<j\le N}$, task classifier $g_{\boldsymbol{\theta}}$.
**Output:** meta-trained feature embedding $\boldsymbol{\psi}^{tr}$, meta-trained adversarial classifiers $\{\boldsymbol{\theta}_{ij}^{tr}\}_{1\le i<j\le N}$ and meta-trained task classier $\boldsymbol{\theta}^{tr}$

**while** Stopping condition is not met **do**
  Sample $l \in [N]$ and minibatch of meta-test sample $\mathcal{B}^{te}$ from the $l$-th domain, minibatch of meta-training samples $\mathcal{B}^{tr}$ from the rest $N-1$ domains
  **for** $S^k, S^t \in \mathcal{B}^{tr}, k<t$ **do**
    $\hat{\mathcal{L}}_{in}^{kt} \leftarrow \left|\hat{\epsilon}_{S^k}(g_{\boldsymbol{\theta}_{kt}} \circ f_{\boldsymbol{\psi}}) - \hat{\epsilon}_{S^t}(g_{\boldsymbol{\theta}_{kt}} \circ f_{\boldsymbol{\psi}})\right| \quad g_{\boldsymbol{\theta}_{kt}} \in \mathcal{G}$ ▷ inner-loop objective Eq. 5
    $\boldsymbol{\theta}_{kt} \leftarrow \boldsymbol{\theta}_{kt} + \eta \nabla_{\boldsymbol{\theta}_{kt}} \hat{\mathcal{L}}_{in}^{kt}$ ▷ update adversarial classifiers
  **end for**
  $\boldsymbol{\psi}' \leftarrow \boldsymbol{\psi} - \alpha\eta \nabla_{\boldsymbol{\psi}} \sum_{k,t} \hat{\mathcal{L}}_{in}^{kt}$ ▷ update feature embedding
  **for** $S^k \in \mathcal{B}^{tr}$ **do**
    $\hat{\mathcal{L}}_{out}^{k} \leftarrow \left|\hat{\epsilon}_{S^k}(g_{\boldsymbol{\theta}_{kl}} \circ f_{\boldsymbol{\psi}'}) - \hat{\epsilon}_{\mathcal{B}^{te}}(g_{\boldsymbol{\theta}_{kl}} \circ f_{\boldsymbol{\psi}'})\right| \quad g_{\boldsymbol{\theta}_{kl}} \in \mathcal{G}$ ▷ outer-loop objective Eq. 4
    $\boldsymbol{\theta}_{kl} \leftarrow \boldsymbol{\theta}_{kl} + \eta \nabla_{\boldsymbol{\theta}_{kl}} \hat{\mathcal{L}}_{out}^{k}$ ▷ update adversarial classifiers
  **end for**
  $\boldsymbol{\psi} \leftarrow \boldsymbol{\psi} - \gamma\eta \nabla_{\boldsymbol{\psi}} \sum_{k} \hat{\mathcal{L}}_{out}^{k}$ ▷ update feature embedding
  $\hat{\mathcal{L}}_{task} \leftarrow \hat{\epsilon}_{\mathcal{B}^{tr}}(g_{\boldsymbol{\theta}} \circ f_{\boldsymbol{\psi}})$ ▷ task objective
  $[\boldsymbol{\theta}, \boldsymbol{\psi}] \leftarrow [\boldsymbol{\theta}, \boldsymbol{\psi}] - \eta \nabla_{\boldsymbol{\theta},\boldsymbol{\psi}} \hat{\mathcal{L}}_{task}$ ▷ update task classifier and feature embedding
**end while**

**Algorithm 2** Meta-test.

**Input data:** $N$ source-domain training samples, hyperparameters: $\eta$, $\alpha$
**Output:** feature embedding $F_{\boldsymbol{\psi}}$, task classifier $T_{\boldsymbol{\theta}}$

$\boldsymbol{\psi} \leftarrow \boldsymbol{\psi}^{tr}$, $\{\boldsymbol{\theta}_{ij}\}_{1\le i<j\le N} \leftarrow \{\boldsymbol{\theta}_{ij}^{tr}\}_{1\le i<j\le N}$, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{tr}$ ▷ initialization
**while** Stopping condition is not met **do**
  Sample minibatch of training samples $\mathcal{B}$ from all the $N$ source domains
  **for** $S^k, S^t \in \mathcal{B}, 1 \le k < t \le N$ **do**
    $\hat{\mathcal{L}}_{in}^{kt} \leftarrow \left|\hat{\epsilon}_{S^k}(g_{\boldsymbol{\theta}_{kt}} \circ f_{\boldsymbol{\psi}}) - \hat{\epsilon}_{S^t}(g_{\boldsymbol{\theta}_{kt}} \circ f_{\boldsymbol{\psi}})\right| \quad g_{\boldsymbol{\theta}_{kt}} \in \mathcal{G}$ ▷ inner-loop objective Eq. 5
    $\boldsymbol{\theta}_{kt} \leftarrow \boldsymbol{\theta}_{kt} + \eta \nabla_{\boldsymbol{\theta}_{kt}} \hat{\mathcal{L}}_{in}$ ▷ update adversarial classifiers
  **end for**
  $\boldsymbol{\psi} \leftarrow \boldsymbol{\psi} - \alpha\eta \nabla_{\boldsymbol{\psi}} \sum_{k,t} \hat{\mathcal{L}}_{in}^{kt}$ ▷ update feature embedding
  $\hat{\mathcal{L}}_{task} \leftarrow \hat{\epsilon}_{\mathcal{B}}(g_{\boldsymbol{\theta}} \circ f_{\boldsymbol{\psi}})$ ▷ task objective
  $[\boldsymbol{\theta}, \boldsymbol{\psi}] \leftarrow [\boldsymbol{\theta}, \boldsymbol{\psi}] - \eta \nabla_{\boldsymbol{\theta},\boldsymbol{\psi}} \hat{\mathcal{L}}_{task}$ ▷ update classifier and feature embedding
**end while**

In practice, we specify the previous bilevel meta-learning algorithm as the first-order MAML (Finn et al., 2017). In particular, the meta-parameter $\boldsymbol{\phi}$ is defined as the parameter initialization for the inner-loop learning algorithm $A_{\boldsymbol{\phi}}^{f}(\cdot)$, which corresponds to one or multiple steps of gradient descent for optimizing the inner-loop objective in the constrained parameter space $\mathcal{C}(\boldsymbol{\phi})$ by the initialization with $\boldsymbol{\phi}$. Given a batch of training samples $\mathcal{B} = \{(\mathcal{B}_i^{tr}, \mathcal{B}_i^{te})\}_{1\le i\le M}$ which contains $M$ pairs of meta-training and meta-test domains, the sample size of each meta-training domain and the meta-test domain in $\mathcal{B}$ are both $b$. The update in one iteration with $m$ inner-loop steps is computed as,

$$\boldsymbol{\phi} = \boldsymbol{\phi} - \gamma\eta \sum_{i=1}^{M} \nabla_{\boldsymbol{\psi}} \hat{\mathcal{L}}_{out}\left(\boldsymbol{\psi}; (\mathcal{B}_i^{tr}, \mathcal{B}_i^{te})\right)\Big|_{\boldsymbol{\psi}=A_{\boldsymbol{\phi}}^{f}(\mathcal{B}_i^{tr})} \tag{6}$$

$$s.t.\ \underbrace{\psi_i^{(0)} = \boldsymbol{\phi}; \boldsymbol{\psi}_i^{(m)} = \boldsymbol{\psi}_i^{(m-1)} - \alpha\eta\nabla_{\boldsymbol{\psi}_i^{(m-1)}}\hat{\mathcal{L}}_{in}\left(\boldsymbol{\psi}_i^{(m-1)}; \mathcal{B}_i^{tr}\right)}_{A_{\boldsymbol{\phi}}^f(\mathcal{B}_i^{tr})}, \tag{7}$$

where $\eta$ denotes the learning rate and $\alpha, \gamma$ denote the adversarial factors of the inner-loop and outer-loop, respectively. In addition to the first-order MAML framework, there exit other gradient-based meta-learning frameworks used in the prior work (Li et al., 2018c; Balaji et al., 2018). We analyze the differences to these work and propose two variants of our approach in Appendix 1.

We use an adversarial training strategy (Goodfellow et al., 2014; Zhang et al., 2021) to optimize the inner-loop and outer-loop objectives ($\hat{\mathcal{L}}_{in}$, $\hat{\mathcal{L}}_{out}$ in Def. 2). Following the previous work (Zhang et al., 2021), the $\mathcal{Y}$-discrepancy is estimated by the trained classifier using gradient ascent updates, while the minimizing of $\mathcal{Y}$-discrepancy is performed via gradient descent w.r.t the parameters of feature embedding. The whole meta-learning procedure is shown in Algorithms 1 & 2 and described as follows.

### 4.3 Meta-training

As shown in Algorithm 1, lines 3–7 show an adversarial training process to optimize the inner-loop objective $\hat{\mathcal{L}}_{in}$, which can be seen as a two-player minimax game between adversarial classifiers and the feature embedding. Lines 8–12 show a similar way to optimize the outer-loop objective $\hat{\mathcal{L}}_{out}$ via adversarial training. In addition, lines 13–14 show the training process of the classification task w.r.t. the task classifier and feature embedding with the source samples.

### 4.4 Meta-test

As shown in Algorithm 2, the learned feature embedding is further trained on all the $N$ source domains with the inner-loop objective in lines 4–8 and simultaneously, the classification task w.r.t. the task classifier and feature embedding is also trained with the source samples in lines 9–10.

### 4.5 Computational complexity

Following the convergence analysis on bilevel meta-learning by Ji et al. (2022), we assume that $\nabla\hat{\mathcal{L}}_{in}(\cdot)$ and $\nabla\hat{\mathcal{L}}_{out}(\cdot)$ are Lipschitz continuous, $\nabla\hat{\mathcal{L}}_{out}(\cdot)$ has a bounded variance and the batch size is large enough. Then, to achieve $\mathbb{E}[\|\nabla\hat{\mathcal{L}}_{out}(\boldsymbol{\phi})\|] \leq \varepsilon$, we need $\mathcal{O}(\varepsilon^{-2})$ iterations. Therefore, by the computational cost of each iteration analyzed in Appendix 1, we need a total number $\mathcal{O}(mbN^3\varepsilon^{-2})$ of gradient computations.

## 5 Theoretical analysis

We analyze the learned feature distribution from a geometric perspective. For convenience in presentation, we regard the feature embedding $A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})$ as a mapping from a domain $\mathbb{D}$ on $\mathcal{X}\times\mathcal{Y}$ to a domain $\mathbb{D}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}$ on Cartesian product of the feature space and the output space $\mathbb{R}^d\times\mathcal{Y}$. To show such definition is reasonable, we can regard the feature embedding as a random transformation $\Phi(x'|x)$, where $x\in\mathcal{X}$ and $x'\in\mathbb{R}^d$. In particular, the deterministic representation function is a special case such that $\Phi(x'|x)$ is the Dirac delta function $\delta_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})(x)}$. Therefore, we can define the domain on $\mathbb{R}^d\times\mathcal{Y}$ as $\mathbb{D}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}(x',y)=\int\Phi(x'|x)\mathbb{D}(x,y)dx$, for any $y\in\mathcal{Y}$. We denote the set of all domains on $\mathbb{R}^d\times\mathcal{Y}$ induced by $A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})$ as $\mathfrak{P}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}$. The associated $\mathcal{Y}$-discrepancy, equivalent to Def. 1, is defined as follows.

**Definition 3** Let $g\in\mathcal{G}$ be the classifier and $A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})$ be the feature embedding, then, the $\mathcal{Y}$ -discrepancy between two domains $\mathbb{S}$ and $\mathbb{T}$ is defined as:

$$\mathrm{disc}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}(\mathbb{S},\mathbb{T}) := \sup_{g\in\mathcal{G}}\left|\epsilon_{\mathbb{T}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}}(g)-\epsilon_{\mathbb{S}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}}(g)\right|. \tag{8}$$

**Definition 4** *(Intrinsic domain discrepancy)* Given a feature embedding $A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})$, We define the intrinsic domain discrepancy as the $\mathcal{Y}$-discrepancy between the target domain $\mathbb{T}$ and the convex hull of source domains $\mathrm{conv}(\mathcal{S})$:

$$\mathrm{disc}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}(\mathbb{T},\mathrm{conv}(\mathcal{S}))=\mathrm{disc}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}\left(\overline{\mathbb{T}}^*,\mathbb{T}\right), \tag{9}$$

where $\overline{\mathbb{T}}^*$ denotes the nearest point to the target domain in $\mathrm{conv}(\mathcal{S})$,

$$\overline{\mathbb{T}}^* := \operatorname*{arg\,min}_{\overline{\mathbb{T}}\in\mathrm{conv}(\mathcal{S})}\mathrm{disc}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}(\overline{\mathbb{T}},\mathbb{T}). \tag{10}$$

**Proposition 1** **(Geometric understanding)** *Given a feature embedding $A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})$, we consider a pseudo-metric space $\left(\mathcal{M}(\mathfrak{P}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}),\mathrm{disc}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}(\cdot,\cdot)\right)$, defined as the space of all domains $\mathfrak{P}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}$ equipped with a pseudo-metric $\mathrm{disc}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}(\cdot,\cdot)$. Let $\widetilde{\mathcal{S}}$ denote the average of source domains $\widetilde{\mathcal{S}}=\frac{1}{N}\sum_{i\in[N]}\mathbb{S}^i$ and $\overline{\mathbb{T}}^*$ be defined as Def. 4, by triangle inequality w.r.t. the pseudo-metric, we first have*:

$$\mathrm{disc}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}(\mathbb{T},\mathrm{conv}(\mathcal{S}))\le\mathrm{disc}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}\left(\mathbb{T},\widetilde{\mathcal{S}}\right)+\mathrm{disc}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}\left(\overline{\mathbb{T}}^*,\widetilde{\mathcal{S}}\right). \tag{11}$$

*Then, we assume that there exits a meta-distribution over the set of all domains, represented as $\mathscr{P}$. We also assume that the classifier class $\mathcal{G}$ has a finite VC-dimension $d$. Given the training set of $N$ source domains $\hat{\mathcal{S}}$ and the associated meta-sample $\{(\hat{\mathcal{D}}_i^{tr},\hat{\mathcal{D}}_i^{te})\}_{1\le i\le M}$ defined in Eq. 1, we have for any $\delta>0$, with probability at least $1-5\delta$,*

$$\begin{aligned}
&\mathbb{E}_{\mathscr{P}}\left[\mathrm{disc}_{A^f_{\boldsymbol{\phi}}(\hat{\mathcal{S}})}\left(\mathbb{T}, \mathrm{conv}(\mathcal{S})\right)\right] \\
&\lessapprox \underbrace{\frac{1}{M}\sum_{i=1}^{M}\frac{1}{|\hat{\mathcal{D}}_i^{tr}|}\sum_{\hat{\mathbb{S}}_i^j\in\hat{\mathcal{D}}_i^{tr}}\hat{\mathrm{disc}}_{A^f_{\boldsymbol{\phi}}(\hat{\mathcal{D}}_i^{tr})}\left(\hat{\mathcal{D}}_i^{te},\hat{\mathbb{S}}_i^j\right)}_{\text{meta-training objective}} + \underbrace{\frac{2}{N}\sum_{i<j}^{N}\hat{\mathrm{disc}}_{A^f_{\boldsymbol{\phi}}(\hat{\mathcal{S}})}\left(\hat{\mathbb{S}}^i,\hat{\mathbb{S}}^j\right)}_{\text{meta-test objective}} \\
&+4\sqrt{\frac{8d\log(2en/d)+8\log(4/\delta)}{n}}+\sqrt{\frac{\log\delta^{-1}}{2N}}.
\end{aligned} \tag{12}$$

***Proof*** In Appendix 1. □

***Remark 1*** Proposition 1 shows that the expectation of intrinsic domain discrepancy can be approximately upper-bounded by (i) the empirical objective of meta-training and (ii) the empirical objective of meta-test. Thus, the meta-training procedure directly optimizes the first empirical term (i), where the optimized meta-parameter is denoted as $\boldsymbol{\phi}^*$. Then, the second empirical term (ii) can also be minimized, since $A^f_{\boldsymbol{\phi}^*}(\hat{\mathcal{S}})$ is defined as an algorithm for optimizing the discrepancy across source domains. Therefore, the proposed meta-learning approach can approximately minimize the upper bound of intrinsic domain discrepancy. An intuitive illustration of the meta-learning procedure is shown in Fig. 2.

To show the effectiveness of optimizing the intrinsic domain discrepancy for DG, we give a generalization bound as follows.

**Proposition 2 (Upper bound)** *Albuquerque et al.* (2020). *Let* $h = g \circ A^f_{\boldsymbol{\phi}}(\hat{\mathcal{S}})$ *be the hypothesis. We assume that there exists a meta-distribution* $\mathscr{P}$ *over the set of domains. Then,*

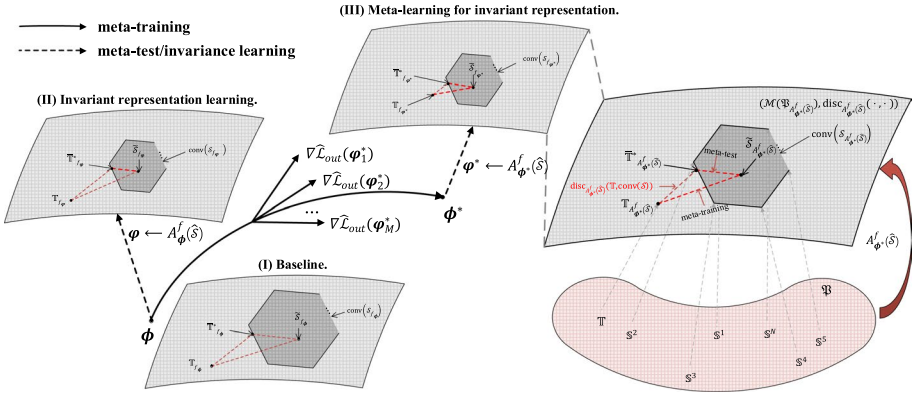



**Fig. 2** Geometric understanding. Independently performing the inner-loop optimization based on the initial parameter $\boldsymbol{\phi}$ can only reduce the discrepancy across sources, as shown by (**I**)→(**II**). The bilevel meta-training approach optimizes the meta-parameter $\boldsymbol{\phi}^*$ such that performing the inner-loop optimization in meta-test can reduce not only the discrepancy across sources but also the discrepancy between the target and sources, resulting in an optimization on the intrinsic domain discrepancy (Def. 4), as shown by (**III**)

$$\mathbb{E}_{\mathscr{P}}\left[\epsilon_{\mathbb{T}}(h)\right] \leq \mathbb{E}_{\mathscr{P}}\left[\mathrm{disc}_{A^f_{\phi}(\hat{\mathcal{S}})}(\mathbb{T}, \mathrm{conv}(\mathcal{S}))\right] + \frac{1}{N}\sum_{i=1}^{N}\epsilon_{\mathbb{S}^i}(h) + \frac{2}{N}\sum_{i<j}^{N}\mathrm{disc}_{A^f_{\phi}(\hat{\mathcal{S}})}\left(\mathbb{S}^i, \mathbb{S}^j\right).$$

***Proof*** The proof largely follows Albuquerque et al. (2020) with only slight modification of replacing the $\mathcal{H}$-divergence with $\mathcal{Y}$-discrepancy and taking expectation on the target domain. □

***Remark 2*** Proposition 2 gives an upper bound for DG, which consists of (i) the intrinsic domain discrepancy, (ii) the weighted average of source-domain errors and (iii) the discrepancy across domains. Compared with the invariance learning approach, which can be seen as only performing the inner-loop or the outer-loop optimization of our approach, the proposed bilevel meta-learning algorithm can further minimize the intrinsic domain discrepancy while also optimizing the discrepancy across source domains by meta-test.

## 6 Experiments

### 6.1 Experimental settings

#### 6.1.1 Datasets and evaluation metrics

Following (Gulrajani & Lopez-Paz, 2020), we evaluate the proposed algorithm on five real-world datasets, including `PACS` (Li et al., 2017) (9,991 images, 7 classes and 4 domains), `VLCS` (Fang et al., 2013) (10,729 images, 5 classes and 4 domains), `OfficeHome` (Venkateswara et al., 2017) (15,588 images, 65 classes, 4 domains),

**Table 1** Hyperparameter, the default value and distribution for random search

| Hyperparameters | Default value | Random distribution |
|---|---|---|
| Batch size (`DomainNet`) | 32 | $2^{\mathcal{U}(3,5)}$ |
| Batch size (Other datasets) | 32 | $2^{\mathcal{U}(3,5.5)}$ |
| Dopout | 0 | Random select from $\{0, 0.1, 0.5\}$ |
| Larning rate $\eta$ | $5e^{-5}$ | $10^{\mathcal{U}(-5,-3.5)}$ |
| Generator learning rate | $5e^{-5}$ | $10^{\mathcal{U}(-5,-3.5)}$ |
| Classifier learning rate | $5e^{-5}$ | $10^{\mathcal{U}(-5,-3.5)}$ |
| Weight decay | 0 | $10^{\mathcal{U}(-6,-2)}$ |
| Generator weight decay | 0 | $10^{\mathcal{U}(-6,-2)}$ |
| Classifier weight decay | 0 | $10^{\mathcal{U}(-6,-2)}$ |
| Adam $\beta_1$ | 0.5 | Random select from $\{0, 0.5\}$ |
| Inner-loop gradient steps (meta-training) $m$ | 5 | Random select from $\{5, 10, 15\}$ |
| Inner-loop gradient steps (meta-test) $m$ | 15 | Random select from $\{5, 10, 15\}$ |
| Adversarial factor (inner) $\alpha$ | 1 | $10^{\mathcal{U}(-1,1)}$ |
| Adversarial factor (outer) $\gamma$ | 1 | $10^{\mathcal{U}(-1,1)}$ |

$\mathcal{U}(a, b)$ denotes a random variable sampled according to the uniform distribution on $[a, b]$

TerraIncognita (Beery et al., 2018), (24,788 images, 10 classes and 4 domains) and DomainNet (Peng et al., 2019) (586,575 images, 345 classes, 6 domains).

We report the out-of-domain accuracy for each dataset and their average, i.e., we use the training set of each source domain to train a model and use the validation sets aggregated by source domains for model selection. Each reported result is the average of three independent repetitions with different hyperparameters, initialization and dataset splits.

**Optimization protocol** For a fair comparison, we follow training and evaluation protocol by Gulrajani and Lopez-Paz (2020) for our method and other baselines. In particular, we use an ImageNet pretrained ResNet-50 (Gulrajani & Lopez-Paz, 2020) as the feature embedding and Adam as the optimizer in all experiments. For hyperparameter search, each hyperparameter is assigned with a default value as well as a range near the default value, all hyperparameters are tuned jointly via random search (Gulrajani & Lopez-Paz, 2020) according to their search distributions with a maximum number of 20 trials. The settings of hyperparameter search for our method and other baselines are the same, except for some hyperparameters specific to ours, which are detailed listed in Table 1.

## 6.2 Results

Table 2 shows the main results and Tables 3 & 4 show the ablation study.

### 6.2.1 Methods

We make comparisons with several related methods in Table 2. The compared approaches include ERM (Vapnik, 1999), domain-invariance learning (Chattopadhyay et al., 2020; Ganin et al., 2016; Sun & Saenko, 2016; Li et al., 2018a, b; Nam et al., 2019; Arjovsky

**Table 2** Accuracy (%) on five DG datasets using pretrained ResNet-50 backbone. † denotes results of the baseline are reproduced under the same training and evaluation protocol (by Gulrajani and Lopez-Paz (2020)) as ours. Results of the other three baselines are from the original literature Dou et al. (2019); Chattopadhyay et al. (2020); Xiao et al. (2021)

| Algorithm | PACS | VLCS | OfficeHome | TerraInc | DomainNet | Avg. |
|---|---|---|---|---|---|---|
| MASF Dou et al. (2019) | 82.7 | – | – | – | – | – |
| DMG Chattopadhyay et al. (2020) | 83.4 | – | – | – | 43.6 | – |
| DILU Xiao et al. (2021) | 85.5 | – | 66.4 | – | – | – |
| ERM† Vapnik (1999) | 85.5 | 77.5 | 66.5 | 46.1 | 40.9 | 63.3 |
| IRM† Arjovsky et al. (2019) | 83.5 | 78.6 | 64.3 | 47.6 | 33.9 | 61.6 |
| DANN† Ganin et al. (2016) | 83.6 | 78.6 | 65.9 | 46.7 | 38.3 | 62.6 |
| CDANN† Li et al. (2018b) | 82.6 | 77.5 | 65.7 | 45.8 | 38.3 | 60.2 |
| CORAL† Sun and Saenko (2016) | 86.2 | 78.8 | 68.7 | 47.7 | 41.5 | 64.5 |
| MMD† Li et al. (2018a) | 84.7 | 77.5 | 66.4 | 42.2 | 23.4 | 58.8 |
| MLDG† Li et al. (2018c) | 84.9 | 77.2 | 66.8 | 47.8 | 41.2 | 63.6 |
| SagNet† Nam et al. (2019) | 86.3 | 77.8 | 68.1 | 48.6 | 40.3 | 64.2 |
| MetaReg† Balaji et al. (2018) | 84.2 | 76.7 | 67.6 | 48.2 | 43.4 | 64.0 |
| **Ours** | **86.8** | **80.7** | **69.8** | **51.0** | **44.2** | **66.5** |

**Table 3** Ablation study on inner-loop and outer-loop objectives

| $\hat{\mathcal{L}}_{in}$ | $\hat{\mathcal{L}}_{out}$ | PACS | DomainNet | Avg. |
|---|---|---|---|---|
| $\mathcal{Y}$-Disc (*single-loop*) | | 84.7 | 41.2 | 63.0 |
| Task | $\mathcal{Y}$-Disc | 85.5 | 41.7 | 63.6 |
| $\mathcal{Y}$-Disc+Task | Task | 85.9 | 40.8 | 63.4 |
| $\mathcal{Y}$-Disc | $\mathcal{Y}$-Disc | **86.8** | **44.2** | **65.5** |

**Table 4** Ablation study on bilevel meta-learning

| Method | PACS | DomainNet | Avg. |
|---|---|---|---|
| MLDG Li et al. (2018c) | 84.9 | 41.2 | 63.1 |
| MetaReg Balaji et al. (2018) | 84.2 | 43.4 | 63.8 |
| Ours-MLDG (Eq. 13) | 86.1 | 42.8 | 64.5 |
| Ours-MetaReg (Eq. 14) | 85.5 | 43.7 | 64.6 |
| Ours | **86.8** | **44.2** | **65.5** |

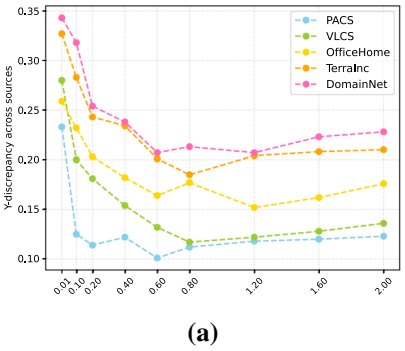
et al., 2019; Xiao et al., 2021) and meta-learning (Li et al., 2018c; Balaji et al., 2018; Dou et al., 2019). Compared with these baselines, our algorithm achieves the best results on all the five datasets, which shows the effectiveness of the proposed bilevel optimization algorithm for DG.

### 6.2.2 Ablation study on inner-loop and outer-loop objectives

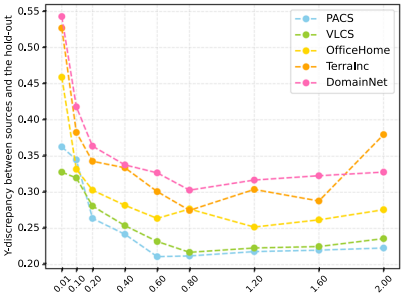
As shown in Table 3, we compare a range of variations of choosing the inner-loop or outer-loop objectives between task objective and $\mathcal{Y}$-discrepancy. The first line is similar to the invariance learning approach (Zhang et al., 2021), which optimizes the $\mathcal{Y}$-discrepancy across different source domains. Compared with this baseline, our approach (bottom line) achieves better results on both datasets, which shows that the proposed bilevel optimization algorithm can improve invariant representation learning for DG. In addition, compared with other meta-learning approaches, the proposed algorithm achieves the best results, which shows the potential of optimizing domain discrepancy to reduce domains shift for DG.

### 6.2.3 Ablation study on bilevel meta-learning

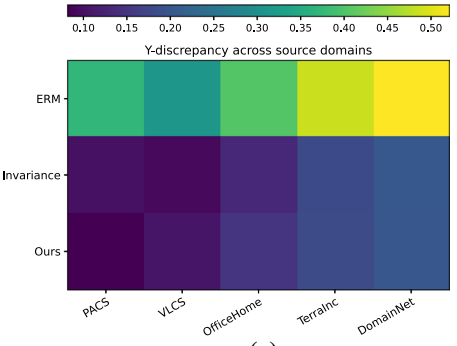
As shown in Table 4, we compare with two prior meta-learning algorithms (Li et al., 2018c; Balaji et al., 2018). We further make connection to these methods by unifying the empirical inner-loop and outer-loop objectives as our approach, and present two baselines Ours-MLDG and Ours-MetaReg to compare the frameworks of bilevel meta-learning. Results show that our approach is more effective than other variants of meta-learning framework. Besides, Ours-MLDG and Ours-MetaReg outperform the original MLDG (Li et al., 2018c) and MetaReg (Balaji et al., 2018), respectively. This shows the effectiveness of meta-learning the invariant representation for DG.



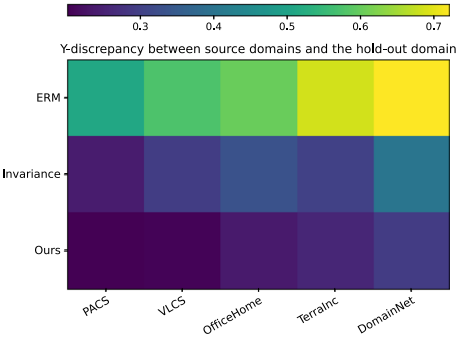

**(a)**




**(b)** Sensitivity of adversarial factor $\alpha$ or $\gamma$ for optimizing the $\mathcal{Y}$-discrepancy.




**(c)**




**(d)** Comparison between different algorithms for reducing the $\mathcal{Y}$-discrepancy.

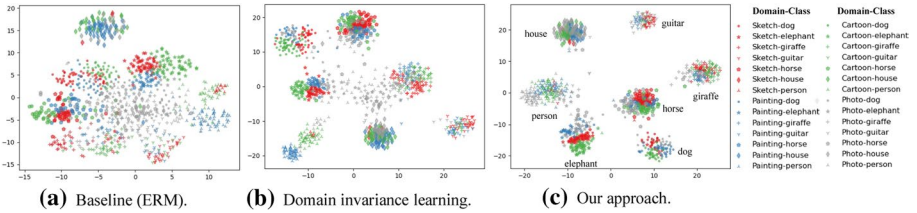
**Fig. 3** The effectiveness of reducing $\mathcal{Y}$-discrepancy by the bilevel optimization algorithm

## 6.3 Analysis

### 6.3.1 Domain discrepancy

In Fig. 3b, we show the effectiveness of adversarial training strategy against the factor $\alpha$ and $\gamma$ for minimizing the $\mathcal{Y}$-discrepancy across different source domains (top left), and the $\mathcal{Y}$-discrepancy between the hold-out domain and source domains (bottom left), respectively. We can find that with the adversarial factors increasing from 0.01 to 2.00, both the $\mathcal{Y}$ -discrepancy across different source domains and the $\mathcal{Y}$-discrepancy between the hold-out domain and source domains first decrease with only some small fluctuations and then come to a plateau or tend to slightly increase. This shows the sensitivity of adversarial factors for minimizing the $\mathcal{Y}$-discrepancy in both inner-loop optimization and outer-loop optimization.

As shown in Fig. 3d, we compare $\mathcal{Y}$-discrepancy (Zhang et al., 2012) with the ERM algorithm and an invariant representation learning algorithm (the same as the first line of Table 3) on five datasets. The top right picture shows that both our approach and invariance learning can better reduce the $\mathcal{Y}$-discrepancy between source domains compared with the ERM algorithm. This is because these two approaches have a training objective to reduce $\mathcal{Y}$-discrepancy across different source domains. In addition, the bottom right picture shows




**(a)** Baseline (ERM). **(b)** Domain invariance learning. **(c)** Our approach.

**Fig. 4** t-SNE visualization of feature representation on PACS when the target domain is *photo*. Each class is represented by a specific marker and each domain is represented in a specific colors where the target domain is in gray

that the $\mathcal{Y}$-discrepancy between the hold-out domain and source domains of our approach is lower than both the ERM algorithm and the invariance learning algorithm, which shows the effectiveness of meta-learning to achieve more robust domain invariance.

### 6.3.2 Visualization

We visualize the learned feature representation in Fig. 1. We randomly select 250 test examples from each domain. As shown in Fig. 4, compared with ERM, both domain-invariant learning and our method can match the feature distributions of source domains; Compared with the domain-invariant learning, our method can also well match the feature distributions of the target and source domains, which benefits from the outer-loop objective in bilevel optimization to improve the robustness to domain shift.

## 7 Conclusion

We investigated a meta-learning approach for invariant representation learning to improve domain generalization. In particular, we learn a more robust domain invariance via a bilevel optimization algorithm, where the inner-loop aims to minimize the $\mathcal{Y}$-discrepancy across source domains while the outer-loop aims to minimize the $\mathcal{Y}$-discrepancy between the target and source domains. Theoretically, we show from a geometric perspective that the meta-learning approach minimizes the $\mathcal{Y}$-discrepancy between the target domain and a convex hull of source domains. Empirically, our approach achieves the best results on five domain generalization datasets among a range of strong baselines.

## Appendix 1: Connections to MLDG and MetaReg

Despite the most significant difference between our approach and these meta-learning algorithms is the inner-loop and outer-loop optimization objectives, we also analyze the differences w.r.t. bilevel meta-learning framework and make connections to these approaches by replacing their original inner-loop and outer-loop objectives with $\hat{\mathcal{L}}_{in}$ and $\hat{\mathcal{L}}_{out}$ in Def. 2.

**MLDG** Li et al. (2018c) can be regarded as adding the inner-loop objectives to the outer-loop (Eq. 3). We revise our meta-learning objective accordingly to connect with MLDG as:

$$\min_{\boldsymbol{\phi}\in\boldsymbol{\Phi}} \sum_{i\in[M]} \left[\hat{\mathcal{L}}_{out}\left(A^f_{\boldsymbol{\phi}}(\hat{\mathcal{D}}^{tr}_i)\right) + \hat{\mathcal{L}}_{in}\left(\boldsymbol{\phi};\hat{\mathcal{D}}^{tr}_i\right)\right] s.t.\ A^f_{\boldsymbol{\phi}}(\hat{\mathcal{D}}^{tr}_i) \in \arg\min_{\boldsymbol{\psi}\in\mathcal{C}(\boldsymbol{\phi})} \hat{\mathcal{L}}_{in}\left(\boldsymbol{\psi};\hat{\mathcal{D}}^{tr}_i\right). \quad (13)$$

It can be viewed as integrating our meta-test procedure into the meta-training procedure. Thus, it can increase the computational cost. We denote this variant of our approach as Ours-MLDG.

**MetaReg** Balaji et al. (2018) can be regarded as meta-learning the regularization instead of the parameter initialization in our approach. We revise our meta-learning objective to connect with MetaReg as:

$$\min_{\boldsymbol{\phi}\in\boldsymbol{\Phi}} \sum_{i\in[M]} \hat{\mathcal{L}}_{out}\left(A^f_{\boldsymbol{\phi}}(\hat{\mathcal{D}}^{tr}_i)\right) s.t.\ A^f_{\boldsymbol{\phi}}(\hat{\mathcal{D}}^{tr}_i) \in \arg\min_{\boldsymbol{\psi}} \hat{\mathcal{L}}_{in}\left(\boldsymbol{\psi};\hat{\mathcal{D}}^{tr}_i\right) + \|\boldsymbol{\psi}-\boldsymbol{\phi}\|_1. \quad (14)$$

Such approach is similar to iMAML (Rajeswaran et al., 2019), where optimizing the Hessian-vector products can be much more costly than our approach, which neglects the second-order gradients as analyzed in Appendix 1. We denote this variant of our approach as Ours-MetaReg.

## Appendix 2: Computational complexity in one iteration

By the gradient updating rule in Eq. 6, we have:

$$\boldsymbol{\phi} = \boldsymbol{\phi} - \gamma\eta \sum_{i=1}^{M}\prod_{n=0}^{m-1}\left(\mathbf{I} - \alpha\eta\nabla^2_{\boldsymbol{\psi}}\hat{\mathcal{L}}_{in}\left(\boldsymbol{\psi}^{(n)};\mathcal{B}^{tr}_i\right)\right)\cdot\nabla_{\boldsymbol{\psi}}\hat{\mathcal{L}}_{out}\left(\boldsymbol{\psi},(\mathcal{B}^{tr}_i,\mathcal{B}^{te}_i)\right)\big|_{\boldsymbol{\psi}=\boldsymbol{\psi}^{(m)}_i}.$$

The first-order MAML Finn et al. (2017) treats $\boldsymbol{\psi}^{(m)}_i$ by the inner-loop updates as a constant and thus neglects the second-order gradients, and then by the definition of empirical $\mathcal{Y}$-discrepancy, the updating rule can be written as:

$$\begin{aligned}
\boldsymbol{\phi} &= \boldsymbol{\phi} - \gamma\eta\sum_{i=1}^{M}\nabla_{\boldsymbol{\psi}}\hat{\mathcal{L}}_{out}\left(\boldsymbol{\psi},(\mathcal{B}^{tr}_i,\mathcal{B}^{te}_i)\right)\big|_{\boldsymbol{\psi}=\boldsymbol{\psi}^{(m)}_i}\\
&= \boldsymbol{\phi} - \gamma\eta\sum_{i=1}^{M}\nabla_{\boldsymbol{\psi}}\sum_{S^k_i\in\mathcal{B}^{tr}_i}\hat{\mathrm{disc}}\left(f_{\boldsymbol{\psi}}\left(\mathcal{B}^{te}_i,S^k_i\right)\right)\big|_{\boldsymbol{\psi}=\boldsymbol{\psi}^{(m)}_i}\\
&= \boldsymbol{\phi} - \gamma\eta\sum_{i=1}^{M}\sum_{S^k_i\in\mathcal{B}^{tr}_i}\nabla_{\boldsymbol{\psi}}\left|\hat{\epsilon}_{\mathcal{B}^{te}_i}\left(g^*\circ f_{\boldsymbol{\psi}}\right)-\hat{\epsilon}_{S^k_i}\left(g^*\circ f_{\boldsymbol{\psi}}\right)\right|_{\boldsymbol{\psi}=\boldsymbol{\psi}^{(m)}_i}\\
&= \boldsymbol{\phi} - \frac{\gamma\eta}{b}\sum_{i=1}^{M}\sum_{S^k_i\in\mathcal{B}^{tr}_i}\mathrm{sgn}(\cdot)\Bigg[\sum_{(x,y)\in\mathcal{B}^{te}_i}\nabla_{\boldsymbol{\psi}}\ell(g^*\circ f_{\boldsymbol{\psi}}(x),y)\big|_{\boldsymbol{\psi}=\boldsymbol{\psi}^{(m)}_i}\\
&\qquad - \sum_{(x',y')\in S^k_i}\nabla_{\boldsymbol{\psi}}\ell(g^*\circ f_{\boldsymbol{\psi}}(x'),y')\big|_{\boldsymbol{\psi}=\boldsymbol{\psi}^{(m)}_i}\Bigg],
\end{aligned}$$

where $g^*$ denotes the optimized classifier for supremum in Def. 1 and $\mathrm{sgn}(\cdot)$ denotes the sign function for $\mathrm{sgn}(\hat{\epsilon}_{\hat{D}_i^{te}}(\cdot) - \hat{\epsilon}_{S_i^k}(\cdot))$. $\boldsymbol{\psi}_i^{(m)}$ is computed in the inner-loop, for each inner-loop step $l \in [m-1]$:

$$
\begin{aligned}
&\boldsymbol{\psi}_i^{(l+1)} = \boldsymbol{\psi}_i^{(l)} - \alpha\eta\nabla_{\boldsymbol{\psi}_i^{(l)}}\hat{\mathcal{L}}_{in}\left(\boldsymbol{\psi}_i^{(l)};\mathcal{B}_i^{tr}\right)\\
=&\boldsymbol{\psi}_i^{(l)} - \alpha\eta\nabla_{\boldsymbol{\psi}_i^{(l)}}\sum_{S_i^k,S_i^t\in\mathcal{B}_i^{tr}}\hat{\mathrm{disc}}_{\mathcal{Y}}\left(f_{\boldsymbol{\psi}_i^{(l)}}\left(S_i^k,S_i^t\right)\right)\\
=&\boldsymbol{\psi}_i^{(l)} - \alpha\eta\sum_{S_i^k,S_i^t\in\mathcal{B}_i^{tr}}\nabla_{\boldsymbol{\psi}_i^{(l)}}\left|\hat{\epsilon}_{S_i^k}\left(g^*\circ f_{\boldsymbol{\psi}_i^{(l)}}\right) - \hat{\epsilon}_{S_i^t}\left(g^*\circ f_{\boldsymbol{\psi}_i^{(l)}}\right)\right|\\
=&\boldsymbol{\psi}_i^{(l)} - \frac{\alpha\eta}{b}\sum_{S_i^k,S_i^t\in\mathcal{B}_i^{tr}}\mathrm{sgn}(\cdot)\left[\sum_{(x,y)\in S_i^k}\nabla_{\boldsymbol{\psi}_i^{(l)}}\ell\left(g^*\circ f_{\boldsymbol{\psi}_i^{(l)}}(x),y\right) - \sum_{(x',y')\in S_i^t}\nabla_{\boldsymbol{\psi}_i^{(l)}}\ell\left(g^*\circ f_{\boldsymbol{\psi}_i^{(l)}}(x'),y'\right)\right],
\end{aligned}
$$

where $b$ is sample size of each meta-training domain or meta-test domain of in each minibatch.

In practice, the size of meta-sample $M$ is equal to the number of source domains $N$ in Eq. 1. Thus, each inner-loop step has $\mathcal{O}(bN^2)$ number of gradient computation. Since first-order MAML treats updates of the inner-loop as a constant for outer-loop gradient computing, thus the gradient operations in the inner-loop and outer-loop for each iteration can be sequential. Therefore, the total number of gradient computation is $\mathcal{O}(N\cdot(mbN^2+bN)) = \mathcal{O}(mbN^3)$.

## Appendix 3: Proof of proposition 1

***Proof*** At the beginning, we introduce the following useful lemma.

□

**Lemma 1** *Let $\mathbb{P}$ and $\mathbb{Q}$ denote two domains and $\hat{\mathbb{P}}$ and $\hat{\mathbb{Q}}$ denote the associated empirical samples with cardinality $n$. Let $A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})$ denote a feature embedding with an arbitrary meta-parameter $\boldsymbol{\phi}$ and arbitrary training samples $\hat{\mathcal{S}}$. Let $\mathcal{G}$ denote the classifier class with a finite VC-dimension $d$. Then, we have for any $\delta > 0$, with probability at least $1-2\delta$:*

$$
\mathrm{disc}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}\left(\mathbb{P},\mathbb{Q}\right) \le \hat{\mathrm{disc}}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}\left(\hat{\mathbb{P}},\hat{\mathbb{Q}}\right) + 2\sqrt{\frac{8d\log(2en/d)+8\log(4/\delta)}{n}}.
$$

***Proof***

$$
\begin{aligned}
&\mathrm{disc}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}\left(\mathbb{P},\mathbb{Q}\right) - \hat{\mathrm{disc}}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}\left(\hat{\mathbb{P}},\hat{\mathbb{Q}}\right)\\
=&\sup_{g\in\mathcal{G}}\left|\epsilon_{\mathbb{P}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}}(g) - \epsilon_{\mathbb{Q}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}}(g)\right| - \sup_{g\in\mathcal{G}}\left|\hat{\epsilon}_{\hat{\mathbb{P}}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}}(g) - \hat{\epsilon}_{\hat{\mathbb{Q}}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}}(g)\right|\\
\le&\sup_{g\in\mathcal{G}}\left|\epsilon_{\mathbb{P}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}}(g) - \epsilon_{\mathbb{Q}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}}(g) - \hat{\epsilon}_{\hat{\mathbb{P}}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}}(g) + \hat{\epsilon}_{\hat{\mathbb{Q}}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}}(g)\right|\\
\le&\sup_{g\in\mathcal{G}}\left|\epsilon_{\mathbb{P}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}}(g) - \hat{\epsilon}_{\hat{\mathbb{P}}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}}(g)\right| + \sup_{g\in\mathcal{G}}\left|\epsilon_{\mathbb{Q}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}}(g) - \hat{\epsilon}_{\hat{\mathbb{Q}}_{A_{\boldsymbol{\phi}}^f(\hat{\mathcal{S}})}}(g)\right|
\end{aligned}
$$

Then, by VC-dimension generalization bound (Corollary 3.19 in the book (Mohri et al., 2018)), we complete the proof. □

By the triangle inequality of $\mathcal{Y}$-discrepancy,

$$\mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\big(\mathbb{T}, \mathrm{conv}(\mathcal{S})\big) = \mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\big(\overline{\mathbb{T}}^*, \mathbb{T}\big) \le \mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\big(\mathbb{T}, \widetilde{\mathcal{S}}\big) + \mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\big(\overline{\mathbb{T}}^*, \widetilde{\mathcal{S}}\big),$$

where $\overline{\mathbb{T}}^* = \arg\min_{\overline{\mathbb{T}} \in \mathrm{conv}(\mathcal{S})} \mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}(\overline{\mathbb{T}}, \mathbb{T})$ is defined as the nearest point to $\mathbb{T}$ in the convex hull of sources $\mathrm{conv}(\mathcal{S})$ w.r.t. the $\mathcal{Y}$-discrepancy and $\widetilde{\mathcal{S}}$ is defined as $\widetilde{\mathcal{S}} = \frac{1}{N}\sum_{i\in[N]} \mathbb{S}^i$.

We can bound the second term of the RHS as follows:

By the definition of $\overline{\mathbb{T}}^*$, we can equivalently define $\overline{\mathbb{T}}^*$ as an optimal combination of the source domains $\{\mathbb{S}^i\}_{1\le i\le N}$: $\overline{\mathbb{T}}^* = \sum_{i=1}^N \boldsymbol{\beta}_i^* \mathbb{S}^i$, such that $\boldsymbol{\beta}^* := \arg\min_{\boldsymbol{\beta}\in\Lambda^N} \mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\big(\sum_{i=1}^N \boldsymbol{\beta}_i \mathbb{S}^i, \mathbb{T}\big)$, where $\Lambda^N$ represents an $N$-dimensional simplex. Then, we have:

$$\begin{aligned}
&\mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\big(\overline{\mathbb{T}}^*, \widetilde{\mathcal{S}}\big) \\
=&\sup_{g\in\mathcal{G}} \left| \sum_{j=1}^N \boldsymbol{\beta}_j \epsilon_{\mathbb{S}^j_{A_\phi^f(\hat{\mathcal{S}})}}(g) - \frac{1}{N}\sum_{i=1}^N \epsilon_{\mathbb{S}^i_{A_\phi^f(\hat{\mathcal{S}})}}(g) \right| \quad \triangleright \text{ definitions of } \overline{\mathbb{T}}^*, \widetilde{\mathcal{S}} \\
=&\sup_{g\in\mathcal{G}} \left| \frac{1}{N}\sum_{j=1}^N \sum_{i=1}^N \boldsymbol{\beta}_j \epsilon_{\mathbb{S}^j_{A_\phi^f(\hat{\mathcal{S}})}}(g) - \frac{1}{N}\sum_{j=1}^N \sum_{i=1}^N \boldsymbol{\beta}_j \epsilon_{\mathbb{S}^i_{A_\phi^f(\hat{\mathcal{S}})}}(g) \right| \\
\le&\sup_{g\in\mathcal{G}} \frac{1}{N}\sum_{j=1}^N \sum_{i=1}^N \boldsymbol{\beta}_j \left| \epsilon_{\mathbb{S}^j_{A_\phi^f(\hat{\mathcal{S}})}}(g) - \epsilon_{\mathbb{S}^i_{A_\phi^f(\hat{\mathcal{S}})}}(g) \right| \quad \triangleright \text{ triangle inequality} \\
\le&\frac{1}{N}\sum_{j=1}^N \sum_{i=1}^N \boldsymbol{\beta}_j \sup_{g\in\mathcal{G}} \left| \epsilon_{\mathbb{S}^j_{A_\phi^f(\hat{\mathcal{S}})}}(g) - \epsilon_{\mathbb{S}^i_{A_\phi^f(\hat{\mathcal{S}})}}(g) \right| \quad \triangleright \text{ Jensen's inequality} \\
=&\frac{1}{N}\sum_{j=1}^N \sum_{i=1}^N \boldsymbol{\beta}_j \mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\big(\mathbb{S}^i, \mathbb{S}^j\big) \quad \triangleright \text{ definition of } \mathrm{disc}_{A_\phi^f}(\cdot,\cdot) \\
\le&\frac{1}{N}\sum_{i=1}^N \max_{j\in[N]} \mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\big(\mathbb{S}^i, \mathbb{S}^j\big) \le \frac{1}{N}\sum_{i=1}^N \sum_{j\in[N]} \mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\big(\mathbb{S}^i, \mathbb{S}^j\big) \\
=&\frac{2}{N}\sum_{i<j}^N \mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\big(\mathbb{S}^i, \mathbb{S}^j\big).
\end{aligned}$$

Thus, we have:

$$\mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}(\mathbb{T}, \mathrm{conv}(\mathcal{S})) \le \mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\Big(\mathbb{T}, \widetilde{\mathcal{S}}\Big) + \frac{2}{N}\sum_{i<j}^N \mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\big(\mathbb{S}^i, \mathbb{S}^j\big). \tag{15}$$

Taking expectation on target domain $\mathbb{T}$ according to the meta-distribution $\mathscr{P}$,

$$\mathbb{E}_{\mathscr{P}}\Big[\mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}(\mathbb{T}, \mathrm{conv}(\mathcal{S}))\Big] \le \mathbb{E}_{\mathscr{P}}\Big[\mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\Big(\mathbb{T}, \widetilde{\mathcal{S}}\Big)\Big] + \frac{2}{N}\sum_{i<j}^N \mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\big(\mathbb{S}^i, \mathbb{S}^j\big). \tag{16}$$

We bound the first term in the RHS by Hoeffding's inequality. Given a set of meta-test domains $\{\mathcal{D}_i^{te}\}_{1\le i\le M}$, where $\mathcal{D}_i^{te} \overset{i.i.d.}{\sim} \mathscr{P}$, then, we have for any $\delta > 0$, with probability at least $1-\delta$,

$$\mathbb{E}_{\mathscr{P}}\left[\mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\left(\mathbb{T},\widetilde{\mathcal{S}}\right)\right] \le \frac{1}{M}\sum_{i=1}^{M}\mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\left(\mathcal{D}_i^{te},\widetilde{\mathcal{S}}\right) + \sqrt{\frac{\log\delta^{-1}}{2N}}. \tag{17}$$

Since the set of meta-training domains $\mathcal{D}_i^{tr}$, for each $1 \le i \le M$, is equal to $\mathcal{S}$ except for the meta-test domain of $\mathcal{D}_i^{te}$, thus we have $\widetilde{\mathcal{S}} \approx \widetilde{\mathcal{D}}_i^{tr}$, where $\widetilde{\mathcal{D}}_i^{tr} = \frac{1}{|\mathcal{D}_i^{tr}|}\sum_{\mathbb{S}_i^j\in\mathcal{D}_i^{tr}}\mathbb{S}_i^j$. Then, we have for any $1 \le i \le M$:

$$\widetilde{\mathcal{S}} \approx \widetilde{\mathcal{D}}_i^{tr} \Longrightarrow \mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\left(\mathcal{D}_i^{te},\widetilde{\mathcal{S}}\right) \approx \mathrm{disc}_{A_\phi^f(\hat{\mathcal{D}}_i^{tr})}\left(\mathcal{D}_i^{te},\widetilde{\mathcal{D}}_i^{tr}\right)$$

By the triangle inequality, we have

$$\mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\left(\mathcal{D}_i^{te},\widetilde{\mathcal{S}}\right) \lessapprox \frac{1}{|\mathcal{D}_i^{tr}|}\sum_{\mathbb{S}_i^j\in\mathcal{D}_i^{tr}}\mathrm{disc}_{A_\phi^f(\hat{\mathcal{D}}_i^{tr})}\left(\mathcal{D}_i^{te},\mathbb{S}_i^j\right)$$

Then, by Lemma 1, and using $|\mathcal{D}_i^{tr}| = |\hat{\mathcal{D}}_i^{tr}|$ equivalently to denote the number of meta-training domains, we have for any $\delta > 0$, with probability at least $1-2\delta$:

$$\mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\left(\mathcal{D}_i^{te},\widetilde{\mathcal{S}}\right) \lessapprox \frac{1}{|\hat{\mathcal{D}}_i^{tr}|}\sum_{\hat{\mathbb{S}}_i^j\in\hat{\mathcal{D}}_i^{tr}}\hat{\mathrm{disc}}_{A_\phi^f(\hat{\mathcal{D}}_i^{tr})}\left(\hat{\mathcal{D}}_i^{te},\hat{\mathbb{S}}_i^j\right) + 2\sqrt{\frac{8d\log(2en/d)+8\log(4/\delta)}{n}} \tag{18}$$

Insert Eq. 18 into Eq. 17 and by the union bound, we have with probability at least $1-3\delta$:

$$\begin{aligned}\mathbb{E}_{\mathscr{P}}\left[\mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\left(\mathbb{T},\widetilde{\mathcal{S}}\right)\right] \lessapprox & \frac{1}{M}\sum_{i=1}^{M}\frac{1}{|\hat{\mathcal{D}}_i^{tr}|}\sum_{\hat{\mathbb{S}}_i^j\in\hat{\mathcal{D}}_i^{tr}}\hat{\mathrm{disc}}_{A_\phi^f(\hat{\mathcal{D}}_i^{tr})}\left(\hat{\mathcal{D}}_i^{te},\hat{\mathbb{S}}_i^j\right) \\ & + 2\sqrt{\frac{8d\log(2en/d)+8\log(4/\delta)}{n}} + \sqrt{\frac{\log\delta^{-1}}{2N}}\end{aligned} \tag{19}$$

Similarly, we can bound the second term in the RHS of Eq. 16 by Lemma 1: for any $\delta$ 0, with probability at least $1-2\delta$:

$$\frac{2}{N}\sum_{i<j}^{N}\mathrm{disc}_{A_\phi^f(\hat{\mathcal{S}})}\left(\mathbb{S}^i,\mathbb{S}^j\right) \le \frac{2}{N}\sum_{i<j}^{N}\hat{\mathrm{disc}}_{A_\phi^f(\hat{\mathcal{S}})}\left(\hat{\mathbb{S}}^i,\hat{\mathbb{S}}^j\right) + 2\sqrt{\frac{8d\log(2en/d)+8\log(4/\delta)}{n}} \tag{20}$$

Finally, we insert Eqs. 19 and 20 into Eq. 16 and by the union bound, then, we complete the proof of Eq. 12. □

**Author contributions** All authors contributed to the study conception and design. CJ mainly contributed on methodology, experimental evaluation, writing; YZ mainly contributed on revision, supervision, funding. All authors read and approved the final manuscript.

**Funding** National Natural Science Foundation of China Grant No. (61976180).

**Data availability** The data is publicly available online.

**Code availablity** https://github.com/jiachenwestlake/MLIR.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Consent to participate** All authors consent to participation.

**Consent for publication** All authors consent to publish this manuscript.

## References


Albuquerque, I., Monteiro, J., Darvishi, M., Falk, T.H., & Mitliagkas, I. (2020). Generalizing to unseen domains via distribution matching. arXiv preprint arXiv:1911.00804

Arjovsky, M., Bottou, L., Gulrajani, I., & Lopez-Paz, D. (2019). Invariant risk minimization. arXiv preprint arXiv:1907.02893

Balaji, Y., Sankaranarayanan, S., Chellappa, R., & Metareg, R. (2018). Towards domain generalization using meta-regularization. In: NeurIPS.

Baxter, J. (2000). A model of inductive bias learning. *JAIR, 12,* 149–198.

Beery, S., Van Horn, G., & Perona, P. (2018). Recognition in terra incognita. In: ECCV (pp. 472–489).

Blanchard, G., Lee, G., & Scott, C. (2011). Generalizing from several related classification tasks to a new unlabeled sample. In: NIPS.

Chattopadhyay, P., Balaji, Y., & Hoffman, J. (2020). Learning to balance specificity and invariance for in and out of domain generalization. In: ECCV (pp. 301–318).

Chen, J., Wu, X.-M., Li, Y., Li, Q., Zhan, L.-M., & Chung, F.-l. (2020). A closer look at the training strategy for modern meta-learning. In: NeurIPS.

Dou, Q., Coelho de Castro, D., Kamnitsas, K., & Glocker, B. (2019). Domain generalization via model-agnostic learning of semantic features. In: NeurIPS.

Fang, C., Xu, Y., & Rockmore, D.N. (2013). Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In: ICCV (pp. 1657–1664).

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML (pp. 1126–1135).

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., & Laviolette, F., et al. (2016). Domain-adversarial training of neural networks. *JMLR, 17*(1), 2030–2096.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In: NeurIPS.

Gulrajani, I., & Lopez-Paz, D. (2020). In search of lost domain generalization. In: ICLR.

Hoffman, J., Mohri, M., & Zhang, N. (2018). Algorithms and theory for multiple-source adaptation. In: NeurIPS.

Ji, K., Yang, J., & Liang, Y. (2022). Theoretical convergence of multi-step model-agnostic meta-learning. *JMLR, 23*(29), 1–41.

Li, D., Gouk, H., & Hospedales, T. (2022). Finding lost DG: Explaining domain generalization via model complexity. arXiv preprint arXiv:2202.00563

Li, H., Jialin Pan, S., Wang, S., & Kot, A.C. (2018). Domain generalization with adversarial feature learning. In: CVPR (pp. 5400–5409).

Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., & Tao, D. (2018). Deep domain generalization via conditional invariant adversarial networks. In: ECCV (pp. 647–663).

Li, D., Yang, Y., Song, Y.-Z., Hospedales, & T.M. Deeper. (2017). Broader and artier domain generalization. In: ICCV (pp. 5543–5551).

Li, D., Yang, Y., Song, Y.-Z., & Hospedales, T. (2018). Learning to generalize: Meta-learning for domain generalization. In: AAAI (pp. 3490–3497).

Mansour, Y., Mohri, M., & Rostamizadeh, A. (2008). Domain adaptation with multiple sources. In: NIPS.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of Machine Learning.

Muandet, K., Balduzzi, D., & Schölkopf, B. (2013). Domain generalization via invariant feature representation. In: ICML (pp. 10–18).
Nam, H., Lee, H., Park, J., Yoon, W., Yoo, D. (2019). Reducing domain gap via style-agnostic networks. arXiv preprint arXiv:1910.11645
Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., & Wang, B. (2019). Moment matching for multi-source domain adaptation. In: ICCV (pp. 1406–1415).
Rajeswaran, A., Finn, C., Kakade, S.M., & Levine, S. (2019). Meta-learning with implicit gradients. In: NeurIPS.
Shao, J.-J., Cheng, Z., Li, Y.-F., & Pu, S. (2021). Towards robust model reuse in the presence of latent domains. In: IJCAI (pp. 2957–2963).
Shui, C., Wang, B., & Gagné, C. (2022). On the benefits of representation regularization in invariance based domain generalization. *Machine Learning,* 111, 895–915.
Sun, B., & Saenko, K. (2016). Deep coral Correlation alignment for deep domain adaptation. In: ECCV Workshops (pp. 443–450).
Vapnik, V. N. (1999). An overview of statistical learning theory. *TNN, 10*(5), 988–999.
Venkateswara, H., Eusebio, J., Chakraborty, S., & Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. In: CVPR (pp. 5385–5394).
Xiao, Z., Shen, J., Zhen, X., Shao, L., & Snoek, C. (2021). A bit more bayesian: Domain-invariant learning with uncertainty. In: ICML (pp. 11351–11361).
Zhang, C., Zhang, L., & Ye, J. (2021). Generalization bounds for domain adaptation. In: NIPS.
Zhang, G., Zhao, H., Yu, Y., & Poupart, P. (2021). Quantifying and improving transferability in domain generalization. In: NeurIPS.