Context-Aware Self-Adaptation for Domain Generalization

Hao Yan [1] **Yuhong Guo** [1 2]

Abstract

Domain generalization aims at developing suitable learning algorithms in source training domains such that the model learned can generalize well on a different unseen testing domain. We present a novel two-stage approach called Context-Aware Self-Adaptation (CASA) for domain generalization. CASA simulates an approximate meta-generalization scenario and incorporates a self-adaptation module to adjust pre-trained meta-source models to the meta-target domains while maintaining their predictive capability on the meta-source domains. The core concept of self-adaptation involves leveraging contextual information, such as the mean of mini-batch features, as domain knowledge to automatically adapt a model trained in the first stage to new contexts in the second stage. Lastly, we utilize an ensemble of multiple meta-source models to perform inference on the testing domain. Experimental results demonstrate that our proposed method achieves state-of-the-art performance on standard benchmarks.

1. Introduction

Deep models generalize well if the training and testing data share the same data distribution. However, when the testing data distribution deviates from the training data, model fine-tuning becomes necessary, requiring additional computational resources that may not be available in resource-constrained applications. Moreover, collecting and labeling additional training data can be time-consuming and impractical for real-time inference scenarios. To address this challenge, domain generalization (DG) methods have emerged as a promising solution. DG aims to train models on training domains with the expectation that they will generalize well to unseen testing domains. Existing DG methods employ various approaches (Wang et al., 2022; Zhou et al., 2022), such as enriching the source data distribution through data augmentation techniques (Volpi et al., 2018; Zhou et al., 2020; Wang et al., 2020; Zhou et al., 2021; Kim et al., 2023) or learning domain-invariant models (Li et al., 2018b;c; Akuzawa et al., 2019; Li et al., 2020; Mahajan et al., 2021; Bui et al., 2021). Some methods utilize meta-learning to mimic the generalization process, enabling the learning of models that exhibit generalization capabilities (Li et al., 2018a; Du et al., 2020). Additionally, the ensemble of multiple source models has been explored as a means to enhance DG performance (Arpit et al., 2022). The primary challenge in domain generalization lies in the lack of access to the testing domain and the limited quantity of source domains available.

In light of these challenges, we propose a novel two-stage method, named Context-Aware Self-Adaptation (CASA) for domain generalization . CASA simulates an approximate meta-generalization scenario and designs a simple but fresh self-adaptation module to adjust pre-trained meta-source models to the meta-target domains while maintaining their predictive capability on the meta-source domains. This is realized through a two-stage training process: A regular model is trained in the first stage on the meta-source domain, while a self-adaptation module is added in the second stage that leverages contextual information, such as the mean of mini-batch features, as domain knowledge to automatically adapt the model trained in the first stage to new contexts in the second stage. To ensure that the adapted meta-target feature vectors remain in the same vector space as the original vectors, which plays a key factor for maintaining generalizability on the original training data, we propose a context-aware feature-wise linear modulation (CaFiLM) mechanism for self-adaptation. This involves learning dimension-specific weights and biases from the feature vector and the context information. These learned weights and biases are then used to linearly modulate the meta-target features. Finally, we perform inference on the testing domain by utilizing an ensemble of multiple adapted meta-source models. To demonstrate the effectiveness of our proposed method, we evaluate it on widely used domain generalization benchmarks. The experimental results show that the proposed

[1]School of Computer Science, Carleton University, Ottawa, Canada [2]Canada CIFAR AI Chair, Amii, Canada. Correspondence to: Hao Yan <haoyan6@cmail.carleton.ca>, Yuhong Guo <yuhong.guo@carleton.ca>.

2nd AdvML Frontiers workshop at 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.
Copyright 2023 by the author(s).

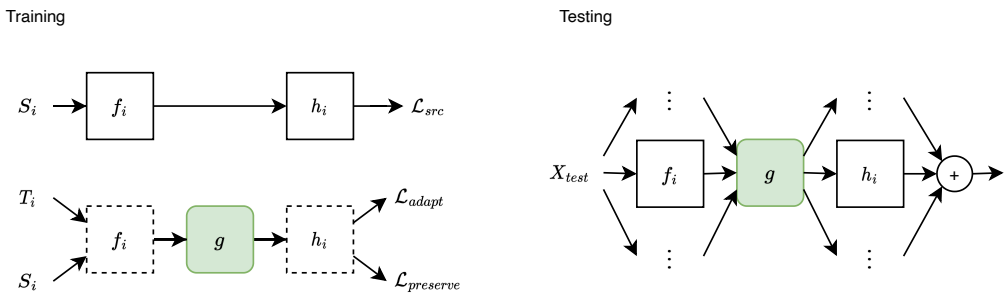

Figure 1. Diagram of the proposed two-stages Context-Aware Self-Adaptation method for domain generalization. Left upper: One model is trained for each meta-source domain. Left lower: Context-Aware Self-Adaptation module is trained to adapt the pre-trained meta-source model to the meta-target domain while preserving the prediction ability on the meta-source domain. Right: Ensemble of the adapted meta-source models are used for testing.

CASA method surpasses the state-of-the-art approaches on small-scale datasets and achieves comparable results on large-scale datasets, producing the best overall performance among a great number of competitors.

2. Proposed Method

This paper addresses the following domain generalization (DG) setting. Given the labeled data from d source training domains $D = \{D_j\}_{j=1}^d$, the goal is to train a model that generalizes well on an unseen testing domain. To tackle this problem, we propose a novel two-stages method called Context-Aware Self-Adaptation (CASA). CASA simulates an approximate meta-generalization scenario by contructing a set of meta-source and meta-arget domain pairs from the training domains. In the first stage, each meta-source domain is used to train a meta-source model. In the second stage, we introduce a context-aware self-adaption module between the pre-trained meta-source feature extractor and classifier. This module is designed to adapt the pre-trained model to the meta-target domain while preserving the prediction ability on the meta-source domain. Finally, we employ an ensemble of multiple adapted meta-source models for inference on the testing domain. The proposed method is illustrated in Figure 1, and will be elaborated below.

2.1. Two-Stage Generalizable Learning

In view of the main obstacle of domain generalization—the test domain is unknown and not accessible during the training stage, we propose to simulate a meta-generalization environment for learning generalizable mechanisms. In this meta-generalization environment, we simulate a set of meta-source and meta-target domain pairs, and design a two-stage training procedure to allow automatic adaptation of pre-trained models in the second stage with a self-adaptation module.

Specifically, given the data from d training domains $D = \{D_j\}_{j=1}^d$, we pick one or multiple domains to form the meta-source domain S_i and keep the rest as the corresponding meta-target domain T_i:

$$S_i = \cup_{j=1}^d \boldsymbol{\delta}_i(j) D_j, \quad T_i = D \backslash S_i. \tag{1}$$

where $\boldsymbol{\delta}_i(j)$ is 1 if D_j is picked to form S_i and 0 otherwise. Repeating this process, we can construct a set of meta-source and meta-target domain pairs: $\mathcal{T} = \{(S_i, T_i)\}$.

For the two-stage training procedure, we train a regular meta-source model for each meta-source domain S_i in the first stage. Considering the classification task, we denote the classification model as the concatenation of a feature extractor $f_i(\cdot; \boldsymbol{\theta}_{f_i})$ and a classifier $h_i(\cdot; \boldsymbol{\theta}_{h_i})$. The classification model is thus denoted as $h_i \circ f_i(\cdot; \boldsymbol{\theta}_{f_i}, \boldsymbol{\theta}_{h_i})$. For each meta-source domain S_i, we train one model $h_i \circ f_i$ with empirical risk minimization, i.e.

$$\begin{aligned} \boldsymbol{\theta}^*_{f_i}, \boldsymbol{\theta}^*_{h_i} &= \arg\min_{\boldsymbol{\theta}_{f_i}, \boldsymbol{\theta}_{h_i}} \mathcal{L}_{src}, \\ \mathcal{L}_{src} &= \mathbb{E}_{(x,y)\in S_i} \ell_{ce}(h_i \circ f_i(x; \boldsymbol{\theta}_{f_i}, \boldsymbol{\theta}_{h_i}), y) \end{aligned} \tag{2}$$

where $\boldsymbol{\theta}^*_{f_i}, \boldsymbol{\theta}^*_{h_i}$ denotes the optimal parameters of the feature extractor and classifier, (x, y) denotes a pair of instance and label sampled from the training data of the meta-source domain S_i and ℓ_{ce} denotes the cross-entropy loss.

Given the pre-trained meta-source model $h_i \circ f_i$ and the meta-target domain T_i, a typical strategy to adapt to the target domain is to deploy the fine-tuning technique. However, the fine-tuning process only seeks to adapt to the specific target domain, providing no generalization capacity in light of unknown test domains. In order to enable generalization to unseen target domains, we need to learn an adaptation mechanism that captures the intrinsic nature of adapting the pre-trained meta-source models to different meta-target domains. To this end, we plan to deploy a feature representation adaptation module that can automatically adapt

the features extracted from the pre-trained meta-source feature extractor in a context-aware manner, allowing the pre-trained meta-source classifier to work effectively with the adapted features from the meta-target domain. Simultaneously, in order to ensure genuine context-aware adaptation, it is reasonable to expect that the upgraded model will continue to perform well in the original meta-source domain.

Following this design scheme, we introduce a task-invariant adaptation module denoted as g in the second stage, which is inserted between the feature extractor f_i and classifier h_i, and upgrades each pre-trained model $h_i \circ f_i$ to a combination model $h_i \circ g \circ f_i$, as shown in Figure 1. This module g is shared across the set of meta adaptation tasks $\mathcal{T}$. The purpose of this adaptation module g is to learn how to adapt the features extracted by f_i to work well with a fixed classifier h_i, given the current task context $\mathcal{C}(\cdot)$. The objective for training the adaptation module g on the set of meta-target domains from $\mathcal{T}$ is defined as follow,

$$\begin{aligned}\min_{\boldsymbol{\theta}_g} \mathcal{L}_{adapt} = \mathbb{E}_{T_i \in \mathcal{T}} \mathbb{E}_{(x,y) \in T_i} \\ \ell_{ce}(h_i \circ g_{\mathcal{C}(T_i)} \circ f_i(x; \boldsymbol{\theta}_g, \boldsymbol{\theta}^*_{f_i}, \boldsymbol{\theta}^*_{h_i}), y)\end{aligned} \tag{3}$$

where (x, y) is a pair of instance and label sampled from the meta-target domain T_i, and $\mathcal{C}(T_i)$ denotes the context information for domain T_i. The model parameters of the pre-trained meta-source feature extractor and classifier remain fixed and the optimization is focused solely on the parameters of the adaptation module g.

However, when adapting pre-trained meta-source models using the objective above, there is a risk that the additional adaptation module and the adaptation process might compromise the prediction ability on the original meta-source domains. This means that the best-performing adaptation module may not necessarily be the one that retains strong performance on all the training domains. To address this issue, it is crucial to introduce an additional objective that specifically preserves the prediction ability on the original meta-source domain as follow:

$$\begin{aligned}\min_{\boldsymbol{\theta}_g} \mathcal{L}_{preserve} = \mathbb{E}_{S_i \in \mathcal{T}} \mathbb{E}_{(x,y) \in S_i} \\ \ell_{ce}(h_i \circ g_{\mathcal{C}(S_i)} \circ f_i(x; \boldsymbol{\theta}_g, \boldsymbol{\theta}^*_{f_i}, \boldsymbol{\theta}^*_{h_i}), y)\end{aligned} \tag{4}$$

where (x, y) is a pair of instance and label sampled from the training data of the meta-source domain S_i, and $\mathcal{C}(S_i)$ denotes the context information for domain S_i. This objective is also essential to induce genuine context-aware adaptation that can adjust the prediction based on the current given context. The overall training objective in the second stage for learning the adaptation module is

$$\min_{\boldsymbol{\theta}_g} \mathcal{L}_{adapt} + \lambda \mathcal{L}_{preserve} \tag{5}$$

where λ is the trade-off hyper-parameter. We expect such a two-stage learning procedure can induce a context-aware generalizable model that can readily adapt to new contexts.

2.2. Context-Aware Self-Adaptation Module

The design of the adaptation module g involves determining the input information, in particular, the form of context representation, and the structure of the module itself. One straightforward approach is to use a multi-layer perceptron (MLP) model as a universal approximation function to process each feature vector corresponding to an instance. However, relying solely on instance-wise processing does not guarantee generalization to unseen testing domains, since an identical instance has the potential to be perceived or classified differently within varying domain contexts. To address this challenge, the module should possess the capability to swiftly decode the domain context in the inference process and dynamically adapt to the provided data. This places significant demands on the choice of the domain context information and the design of the adaptation module.

2.2.1. Choice of Context Information

One of the most common and easily acquired local context information is the mini-batch feature mean. Deep models are typically trained using randomly sampled mini-batches, which are considered to represent the global data distribution. Similarly, during testing, deep models often process data in mini-batches to achieve faster inference speeds. This mini-batch mechanism conveniently allows us to obtain the context information in the form of mini-batch feature mean, which can be considered as an approximation of the domain feature mean during the training and inference processes. By incorporating the mini-batch feature mean as an additional input to g during training, the self-adaptation module can learn to interpret useful context information from it to facilitate the instance feature adaptation. As a result, the input information to the self-adaptation module includes both the instance feature vector and the mini-batch feature mean; and the g function can be expressed as:

$$g_{\mathcal{C}(A_i \in \mathcal{T})} = g_{\mathcal{C}(x)} = g(f_i(x), \mathbb{E}_{x \in X_b} f_i(x); \boldsymbol{\theta}_g) \tag{6}$$

where A_i denotes either a meta-source domain S_i or a meta-target domain T_i, and $\mathcal{C}(x)$ denotes the local context information around x that can be computed as the mini-batch data X_b around instance x in the corresponding domain A_i.

Incorporating the mini-batch feature mean as context information does not introduce complex processing during the training procedure. Unlike requiring the frequent calculation of a global domain feature mean at each iteration step, using the mini-batch feature mean is a computationally efficient approach. Furthermore, it does not alter the inference procedure as there is no need to calculate a global domain feature mean prior to inference. It is important to note that the context information provided by the mini-batch feature mean is not an unfair additional advantage acquired from the testing domain. The use of mini-batch information,

including the feature mean, is a common practice in deep models, as exemplified by techniques like batch normalization. It is interesting that batch normalization eliminates the first-order statistic information from the features to aid in generalization while our method incorporates it as the domain information for the same purpose of enhancing generalization.

2.2.2. Module Structure Design

The structure of the adaptation module g is another crucial consideration. One straightforward approach is to concatenate the feature vector and the mini-batch feature mean, and then employ an MLP to generate a new vector with the same dimension as the original feature vector. However, this design maps the original feature vectors into a completely different vector space. Since the pre-trained meta-source classifier remains fixed for the purpose of generalizable self-adaptation, the ideal objective is to modify the feature vectors in a controlled manner while keeping them in the same vector space as their original representation. Moreover, from the perspective of preserving the prediction ability on the meta-source domain, the output vectors from the self-adaptation module should also reside in the same vector space as the input vectors.

For this purpose, we draw inspiration from the concept of feature-wise linear modulation (FiLM) (Perez et al., 2018) and propose a context-aware feature-wise linear modulation (CaFiLM) method for our self-adaptation module. Specifically, let $\boldsymbol{z} = f_i(x)$ represent the feature vector with dimension size C extracted from the pre-trained meta-source feature extractor f_i for the input instance x. Each individual feature dimension, denoted as $\{z_c\}_{c=1}^{C}$, is modulated using a pair of dimension-specific weight γ_c and bias β_c, i.e.

$$CaFiLM(z_c) = \gamma_c z_c + \beta_c \tag{7}$$

where c denotes the dimension index. Importantly, the weight γ_c and bias β_c are learned from the c-th dimensional feature z_c and the c-th dimensional mini-batch feature mean μ_c,

$$\begin{bmatrix}\gamma_c \\ \beta_c\end{bmatrix} = A^{(2\times 2)} \begin{bmatrix} z_c \\ \mu_c \end{bmatrix} + \boldsymbol{b}^{(2\times 1)} \tag{8}$$

where the mini-batch feature mean is $\boldsymbol{\mu} = \mathbb{E}_{x\in X_b} f_i(x)$. Matrix A and vector $\boldsymbol{b}$ are learnable parameters shared across feature dimensions.

For implementation, feature vector $\boldsymbol{z}$ and mini-batch feature mean $\boldsymbol{\mu}$ are firstly stacked into a two-row matrix. Secondly, a one-dimensional convolutional layer with parameters of weight matrix A and bias vector $\boldsymbol{b}$ is applied along the stacking dimension to produce the weight vector $\boldsymbol{\gamma}$ and bias vector $\boldsymbol{\beta}$. Finally, the feature vector $\boldsymbol{z}$ is linearly modulated through Hadamard product (element-wise product) with $\boldsymbol{\gamma}$ and addition with $\boldsymbol{\beta}$.

Note that the parameters $\boldsymbol{\theta}_g$ of the self-adaptation module g consist solely of the weight matrix A and bias vector $\boldsymbol{b}$, totally only 6 parameters. This simple structure ensures that the self-adaptation module adds minimal extra cost to the classification model. However, when processing large-scale complex datasets, there may be concerns about the ability of the self-adaptation module to handle the increased complexity and diversity of the data. We propose to handle this issue by unfreezing the pre-trained meta-source classifier h_i and fine-tuning its parameters during the training of the self-adaptation module g.

2.3. Inference with Ensemble Mechanism

Once the context-aware self-adaptation module has been trained, we obtain multiple adapted models denoted as $\{(h_i \circ g \circ f_i)\}_{i=1}^{|\mathcal{T}|}$, where $|\mathcal{T}|$ denotes the number of meta-tasks. We deploy an ensemble strategy for the inference process to further improve the test performance. In particular, a simple averaging ensemble strategy can be applied to the predicted probability vectors generated by the multiple adapted models, i.e.

$$\hat{y} = \arg\max_c \left[\mathbb{E}_{i\in\{1:|\mathcal{T}|\}}[(h_i \circ g_{\mathcal{C}(x)} \circ f_i(x))]\right]_c \tag{9}$$

It's important to note the distinction between the proposed approach and EoA (Ensemble of Averages) (Arpit et al., 2022). While EoA also employs ensemble methods for inference, the key differences lie in the training process and the use of training data. In EoA, multiple prediction models are trained using the same training data but with different random seeds and hyper-parameters. The ensemble of these models aims to capture diverse perspectives and variations that arise from the stochasticity of the training process. By contrast, in the proposed approach, the multiple models are trained with different training data from meta-source domains and subsequently adapted using the context-aware self-adaptation module. The use of models trained from different meta-source and meta-target data provides distinct aspects and viewpoints on the testing data, mimicking the benefits of hearing from experts with different levels and areas of knowledge. It shares similarity with the concept of random forests (Breiman, 2001), except our approach can work with a limited number of ensemble entities. Moreover, the context-aware self-adaptation module contributes to the generalization ability of each adapted model. By incorporating the context information and preserving the prediction ability on the meta-source domain, the self-adaptation module further improves the inference performance. Overall, the combination of training models with different meta-source data and applying the context-aware self-adaptation module enables a comprehensive and robust inference process, leveraging both the diversity of training data and the generalization capabilities of the adaptation module.

Table 1. Average accuracy (%) on the testing domain of all tasks for each dataset. Each experiment is conducted 3 times with random seeds and the standard errors is reported with the average values. Results for the comparison methods are cited from the DomainBed and the original papers. The same backbone ResNet-50 is used for all the methods.

Algorithm	PACS	VLCS	OfficeHome	TerraIncognita	DomainNet	Avg
ERM (Gulrajani & Lopez-Paz, 2021)	85.5 $\pm$ 0.2	77.5 $\pm$ 0.4	66.5 $\pm$ 0.3	46.1 $\pm$ 1.8	40.9 $\pm$ 0.1	63.3
I-Mixup (Xu et al., 2020)	84.6 $\pm$ 0.6	77.4 $\pm$ 0.6	68.1 $\pm$ 0.3	47.9 $\pm$ 0.8	39.2 $\pm$ 0.1	63.4
MLDG (Li et al., 2018a)	84.9 $\pm$ 1.0	77.2 $\pm$ 0.4	66.8 $\pm$ 0.6	47.7 $\pm$ 0.9	41.2 $\pm$ 0.1	63.6
SagNet (Nam et al., 2021)	86.3 $\pm$ 0.2	77.8 $\pm$ 0.5	68.1 $\pm$ 0.1	48.6 $\pm$ 1.0	40.3 $\pm$ 0.1	64.2
CORAL (Sun & Saenko, 2016)	86.2 $\pm$ 0.3	78.8 $\pm$ 0.6	68.7 $\pm$ 0.3	47.6 $\pm$ 1.0	41.5 $\pm$ 0.1	64.5
mDSDI (Bui et al., 2021)	86.2 $\pm$ 0.2	79.1 $\pm$ 0.4	69.2 $\pm$ 0.4	48.1 $\pm$ 1.4	42.8 $\pm$ 0.1	65.1
ITL-Net (Gao et al., 2022)	86.4	78.9	69.3	51.0	41.6	65.4
MIRO (Cha et al., 2022)	85.4 $\pm$ 0.4	79.0 $\pm$ 0.0	70.5 $\pm$ 0.4	50.4 $\pm$ 1.1	44.3 $\pm$ 0.2	65.9
SWAD (Cha et al., 2021)	88.1 $\pm$ 0.1	79.1 $\pm$ 0.1	70.6 $\pm$ 0.2	50.0 $\pm$ 0.3	46.5 $\pm$ 0.1	66.9
DNA (Chu et al., 2022)	88.4 $\pm$ 0.1	79.0 $\pm$ 0.1	71.2 $\pm$ 0.1	52.2 $\pm$ 0.4	47.2 $\pm$ 0.1	67.6
EoA (Arpit et al., 2022)	88.6	79.1	72.5	52.3	**47.4**	68.0
Ensemble (Ours)	87.8	80.9	73.0	49.0	44.7	66.8
CASA (Ours)	**89.7** $\pm$ **0.1**	**81.5** $\pm$ **0.1**	**73.5** $\pm$ **0.2**	52.0 $\pm$ 0.2	47.2 $\pm$ 0.1	**68.8**

3. Experiments

3.1. Experimental settings

Experiments were carried out using the domain generalization test-bed called DomainBed (Gulrajani & Lopez-Paz, 2021). We conducted evaluations of our method on five diverse datasets: PACS (Li et al., 2017) , VLCS (Fang et al., 2013) , OfficeHome (Venkateswara et al., 2017) , TerraIncognita (Beery et al., 2018) , and DomainNet (Peng et al., 2019) . These datasets provide a wide range of domains and classes, allowing us to assess the performance and generalization capabilities of our method across different scenarios.

We perform multiple training and single testing domain generalization tasks. For each task, we create meta-source and meta-target domains using the training domains for meta-source model training and meta-target adaptation. We use ResNet-50 as the classification model for all experiments. For meta-source model training, models are optimized with the Adam optimizer with learning rate 5×10^{-5}, batch size 32 for the first 4 datasets and 16 for DomainNet dataset. For the context-aware self-adaptation module training, we employ the same optimizer and batch size. The parameters of the meta-source classifiers are frozen for the PACS, VLCS, and OfficeHome datasets. For the TerraIncognita and DomainNet datasets, we fine-tune the parameters of the meta-source classifiers when training the self-adaptation module. The learning rate is set to 5×10^{-5} for fine-tuning the meta-source classifier parameters and 10^{-3} for the self-adaptation module parameters. The trade-off parameter for the preserving loss is set as 0.1 for PACS dataset and 1 for other datasets. The model selection method here is to use the held-out validation sets from the training data.

3.2. Results

To assess the effectiveness of the proposed method, domain generalization experiments are conducted on all five datasets. The results are reported as the average accuracy of all tasks for each dataset. Table 1 presents the results obtained from these experiments. Firstly, we evaluate the ensemble of the multiple pre-trained meta-source models. The ensemble model, as shown in the "Ensemble (Ours)" row, demonstrates a notable improvement in performance compared to the baseline ERM approach. On average, our ensemble method achieves a significant gain of 3.5 percentage points (pp) in accuracy. This highlights the remarkable impact of model ensemble. The diversity of the meta-source models, obtained through training on different combinations of domains, allows them to provide varied perspectives and insights into the unseen testing data. By aggregating the predictions of these models, we obtain more accurate decisions than relying solely on a single expert with the most knowledge. Furthermore, when comparing our simple ensemble approach to previous domain generalization methods, it outperforms all methods prior to MIRO. This further emphasizes the effectiveness of the ensemble model in improving generalization performance in the domain generalization setting.

Secondly, we evaluate the proposed Context-Aware Self-Adaptation method. Based on the evaluation of the proposed method, the "CASA (Ours)" row in the results table demonstrates a significant performance gain compared to the baseline ERM approach. Our method achieves an impressive performance improvement of 5.5 percentage points (pp) on average. This performance gain can be attributed to two key factors: the ensemble of multiple models and the context-aware self-adaptation module. As mentioned

Table 2. Average accuracy (%) on the testing domain of all tasks for each dataset. Ablation for adaptation training.

Algorithm	PACS	VLCS	OfficeHome	TerraIncognita	DomainNet	Avg
Ensemble ($h_i \circ f_i$)	87.8	80.9	73.0	49.0	44.7	66.8
Ensemble ($h_i \circ g_i \circ f_i$)	88.4	80.1	72.5	51.0	43.1	67.0
CASA	**89.7**	**81.5**	**73.5**	**52.0**	**47.2**	**68.8**

earlier, the ensemble approach, as shown in the "Ensemble (Ours)" results, already provides a 3.5 pp improvement compared to the simple ERM baseline. The additional 2 pp gain is a result of the context-aware self-adaptation module. CASA adapts each pre-trained meta-source model to the meta-target domain while preserving the prediction ability on the meta-source domain. The context information of mini-batch feature mean, treated as domain knowledge, is incorporated into the self-adaptation module to enable self-adaptation. Through exposure to multiple pairs of meta-source models and meta-target domain data, the module learns how to process domain knowledge and modulate the meta-target features, allowing the adapted meta-source model to perform well on both the meta-source and meta-target domains. In the face of new batches of unseen testing data, the CASA method automatically incorporates context information and aids in the adaptation of the meta-source models, enabling them to generalize well on the testing data. Compared to previous methods, our proposed CASA method achieves state-of-the-art results on the first three datasets and demonstrates the best average accuracy overall.

3.3. Ablation study

Is adaptation training necessary? To rigorously evaluate the necessity of self-adaptation module training, an ablation study is conducted. Despite the earlier observation that the CASA method outperforms simple ensemble model, a more comprehensive comparison is performed to ensure fairness. In this study, the self-adaptation module is added as an component of each meta-source feature extractor during meta-source model training, and the ensemble of the meta-source models is evaluated. This analysis aims to assess the impact of incorporating the self-adaptation module at different stages and determine its necessity in improving the overall performance of the ensemble model. The results are shown in Table 2. Upon integrating the adaptation module into the meta-source feature extractor, the ensemble model produces comparable results to the ensemble model without the adaptation module. This indicates that the module alone does not enhance the model's generalization ability. However, through the proposed CASA approach, the self-adaptation module is trained using multiple pairs of meta-source model and meta-target data. The results demonstrate that training the module with adaptation improves the model's generalization to unseen domains.

Table 3. Accuracy (%) on the 4 tasks of OfficeHome dataset.

Algorithm	A	C	P	R	Avg
w/o context	68.3	57.5	81.1	82.8	72.4
w/ context	**70.5**	**58.0**	**82.0**	**83.4**	**73.5**

Is context information necessary? Incorporating the context information of the mini-batch feature mean into the self-adaptation module serves as domain information and enhances the generalization ability. To determine the necessity of this context information, experiments are conducted by removing it from the self-adaptation module. Since only feature vectors are provided as input to the module, an MLP network is used as a replacement for the FiLM structure to adapt the meta-source models to the meta-target domains. The results of these experiments are presented in Table 3. The results demonstrate that self-adaptation training with context information significantly improves the performance across all the tasks. This indicates the importance and effectiveness of incorporating the context information into the self-adaptation module. The improved performance suggests that the module is able to leverage the domain-specific information provided by the mini-batch feature mean to enhance the model's generalization ability and achieve better results on various tasks.

4. Conclusion

In this paper, we propose a novel two-stage approach for domain generalization called Context-Aware Self-Adaptation (CASA). The proposed method simulates a meta-generalization scenario and incorporates a self-adaptation module to adjust pre-trained meta-source models to meta-target domains while preserving their performance on the meta-source domains. The self-adaptation process leverages contextual information, such as mini-batch feature means, as domain knowledge to automatically adapt the model to new contexts. To ensure the adapted meta-target feature vectors remain in the same vector space as the original vectors, we propose a feature-wise linear modulation mechanism to linearly modulate the meta-target features. Finally, we employ an ensemble of multiple meta-source models for inference on the testing domain. Experimental results validate the effectiveness of our approach, achieving state-of-the-art performance on standard benchmarks.

References

Akuzawa, K., Iwasawa, Y., and Matsuo, Y. Adversarial invariant feature learning with accuracy constraint for domain generalization. In *ECML PKDD*, 2019.

Arpit, D., Wang, H., Zhou, Y., and Xiong, C. Ensemble of averages: Improving model selection and boosting performance in domain generalization. In *NeurIPS*, 2022.

Beery, S., Van Horn, G., and Perona, P. Recognition in terra incognita. In *ECCV*, 2018.

Breiman, L. Random forests. *Machine learning*, 2001.

Bui, M.-H., Tran, T., Tran, A., and Phung, D. Exploiting domain-specific features to enhance domain generalization. In *NeurIPS*, 2021.

Cha, J., Chun, S., Lee, K., Cho, H.-C., Park, S., Lee, Y., and Park, S. Swad: Domain generalization by seeking flat minima. In *NeurIPS*, 2021.

Cha, J., Lee, K., Park, S., and Chun, S. Domain generalization by mutual-information regularization with pretrained models. In *ECCV*, 2022.

Chu, X., Jin, Y., Zhu, W., Wang, Y., Wang, X., Zhang, S., and Mei, H. Dna: Domain generalization with diversified neural averaging. In *ICML*, 2022.

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. In *ECCV*, 2020.

Fang, C., Xu, Y., and Rockmore, D. N. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *ICCV*, 2013.

Gao, B., Gouk, H., Yang, Y., and Hospedales, T. Loss function learning for domain generalization by implicit gradient. In *ICML*, 2022.

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. In *ICML*, 2021.

Kim, M., Li, D., and Hospedales, T. Domain generalisation via domain adaptation: An adversarial fourier amplitude approach. In *ICLR*, 2023.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Deeper, broader and artier domain generalization. In *ICCV*, 2017.

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

Li, H., Pan, S. J., Wang, S., and Kot, A. C. Domain generalization with adversarial feature learning. In *CVPR*, 2018b.

Li, H., Wang, Y., Wan, R., Wang, S., Li, T.-Q., and Kot, A. Domain generalization for medical imaging classification with linear-dependency regularization. In *NeurIPS*, 2020.

Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., and Tao, D. Deep domain generalization via conditional invariant adversarial networks. In *ECCV*, 2018c.

Mahajan, D., Tople, S., and Sharma, A. Domain generalization using causal matching. In *ICML*, 2021.

Nam, H., Lee, H., Park, J., Yoon, W., and Yoo, D. Reducing domain gap by reducing style bias. In *CVPR*, 2021.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV 2016 Workshops*, 2016.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

Volpi, R., Namkoong, H., Sener, O., Duchi, J. C., Murino, V., and Savarese, S. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*, 2018.

Wang, J., Lan, C., Liu, C., Ouyang, Y., Qin, T., Lu, W., Chen, Y., Zeng, W., and Yu, P. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Wang, Y., Li, H., and Kot, A. C. Heterogeneous domain generalization via domain mixup. In *ICASSP*, 2020.

Xu, M., Zhang, J., Ni, B., Li, T., Wang, C., Tian, Q., and Zhang, W. Adversarial domain adaptation with domain mixup. In *AAAI*, 2020.

Zhou, K., Yang, Y., Hospedales, T., and Xiang, T. Learning to generate novel domains for domain generalization. In *ECCV*, 2020.

Zhou, K., Yang, Y., Qiao, Y., and Xiang, T. Domain generalization with mixstyle. In *ICLR*, 2021.

Zhou, K., Liu, Z., Qiao, Y., Xiang, T., and Loy, C. C. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.