More is Better: A Novel Multi-view Framework for Domain Generalization

Jian Zhang1  Lei Qi2*  Yingshuan Shi1,*  Yang Gao1
1 Nanjing University  2 Southeast University
zhangjian7369@smail.nju.edu.cn, {syh, gaoy}@nju.edu.cn, qilei@seu.edu.cn

Abstract

Aiming to generalize the model trained in source domains to unseen target domains, domain generalization (DG) has attracted lots of attention recently. The key issue of DG is how to prevent overfitting to the observed source domains because target domain is unavailable during training. We investigate that overfitting not only causes the inferior generalization ability to unseen target domains but also leads unstable prediction in the test stage. In this paper, we observe that both sampling multiple tasks in training stage and generating augmented images in test stage largely benefit generalization performance. Thus, by treating tasks and images as different views, we propose a novel multi-view DG framework. Specifically, in training stage, to enhance generalization ability, we develop a multi-view regularized meta-learning algorithm that employs multiple tasks to produce a suitable optimization direction during updating model. In test stage, to alleviate unstable prediction, we utilize multiple augmented images to yield multi-view prediction, which significantly promotes model reliability via fusing the results of different views of a test image. Extensive experiments on three benchmark datasets validate our method outperforms several state-of-the-art approaches.

1. Introduction

Traditional supervised learning assumes that training and test data are independent and identically distributed. However, this assumption does not always satisfy in real world when there exists the domain shift between training and test data. In recent years, learning a robust and effective model against domain shift has raised considerable attention [2,22]. As one of the most representative paradigms of learning under domain shift, unsupervised domain adaption (UDA) aims to tackle the adaptation from labeled source domain to unlabeled target domain with domain shift. Despite the great success of current UDA models [32, 36, 56], when deploying the previously trained UDA model to other unseen domains, we should re-train the model by incorporating the newly collected data from the unseen domain. This re-training process not only increases extra space/time costs but also violates privacy policy in some cases (e.g., clinical data), rendering these UDA methods being not applicable to some scenarios.

The aforementioned dilemma motivates us to focus on a more applicable yet challenging setting, namely domain generalization (DG) [35]. In DG, by only learning the related knowledge from existing source domains, the trained model is required to be directly applied to previously unseen domains without any re-training procedure. To guarantee the effectiveness of the model on unseen target domain, previous DG methods [26, 35, 41] intend to reduce domain-specific influence from observed source domains by learning domain-invariant representations.

However, well fitting source domain is easy to realize whereas well generalizing to unseen target domain is harder to achieve. By concentrating only on fitting source domains, suffering from the overfitting is inevitable to some extent in most previous methods. Therefore, the meta-learning-based methods [9, 25] have arisen as one of the most popular methods to address overfitting in training stage, which simulates the domain shift episodically to perform regular-
ization. However, these methods train their model with a single task at each iteration, which could cause a biased and noisy optimization direction.

Besides, by investigating the predictions of trained model during the test stage, we notice that overfitting also results in unstable prediction. We conducted an experiment by perturbing (e.g., random crop and flip) the test images. As shown in Fig. 1, their predictions changed after being perturbed. It is because the feature representations of unseen images learned by the overfitted model are more likely to lie near the decision boundary, as shown in Fig. 6. These representations are easily perturbed across the boundary, thus, producing different predictions. This phenomenon is especially serious in DG due to the domain discrepancy between training and test samples.

As aforementioned, the overfitting problem not only appears in training but also largely influences the following testing procedure. To fight against overfitting, we innovatively propose a multi-view framework to better deal with the inferior generalization ability and unstable prediction.

Specifically, in the training stage, we design a multi-view regularized meta-learning algorithm that can regularize the network in a simple yet effective way. This algorithm contains two steps. The first step is to guide the model to pursue a suitable optimization direction via exploiting multi-view information. Unlike the traditional meta-learning algorithms that train the model using only a single task with limited single-view information, we propose to train the model using multiple tasks to find a more accurate direction to perform optimization by integrating multi-view information. In the second step, we update model with the learned direction. Instead of taking a large step along the direction during optimization, our method takes a small step, which can effectively reduce the risk of overfitting.

In the test stage, we propose to deal with the unstable prediction caused by the overfitted model using the multi-view prediction. We argue that current test images with a single view cannot prevent the unstable prediction. Nevertheless, different perturbations applied to the test images can bring abundant information from different views. Thus, if using the image pre-processing perturbations in the test procedure (e.g., the cropping operation), we can obtain multi-view information for a single image. Therefore, in this paper, we augment each test image into multiple views during the test stage and then ensemble their predictions as the final output. By exploiting the multi-view predictions of a single image, we can eliminate the unreliability of prediction and obtain a more robust and accurate prediction.

In summary, we propose a novel multi-view framework to enhance the generalization and stabilize the prediction in both training and test stages. Our contribution can be summarized as follows:

- In the training stage, we design a simple yet effective multi-view regularized meta-learning scheme to prevent overfitting and help find a better weight space.
- During the test stage, we introduce the multi-view prediction to boost the reliability of the predictions by exploiting information from multiple views.
- The effectiveness of our method is validated by extensive experiments on multiple DG benchmark datasets. The proposed method outperforms other state-of-the-art methods.

2. Related Work

**Domain Generalization.** Domain generalization (DG) has been proposed recently to deal with learning to generalize to unseen target domain. Current DG methods can be roughly classified into three categories: Domain-invariant feature learning, Data augmentation and Regularization.

Domain-invariant feature learning based methods aim to extract domain-invariant features by reducing the effect of domain-specific factor. Earlier methods [12, 26, 35] learn the features by projecting original features into a common space where the domain-invariant features lie. With the rise of deep learning, adversarial training [26, 29, 41, 58] has been widely employed as a major technique to learn the domain-invariant features by reducing domain gap between multiple source domains. In addition, other methods [3, 6, 39] intend to extract domain-invariant features by disentangling it from domain-specific features.

Data augmentation based methods wish to simulate samples from unseen target domains to train a robust model. Most of them perform image-level augmentation [27, 54, 55, 59] with generative adversarial networks [13] or adaptive instance normalization [16]. Since feature statistics contains style information, some methods try to augment features by modifying its statistics [20, 38, 61] or injecting generated noise to the statistics [28, 45] in different layers.

Since overfitting largely hurts the generalization ability of a model, regularization based methods prevent this dilemma by regularizing the network during training. Several works [4, 50] add auxiliary self-supervised loss to perform regularization. [9, 25] adopt meta-learning framework to regularize the network by simulating domain shift during training. Our simple yet effective meta-learning algorithm belongs to the regularization based method that can better prevent overfitting as validated in Sec. 4.

**Meta-Learning.** Meta-learning [47] is a long standing topic that learns how to learn. Recently, MAML [11] has been proposed as a simple model-agnostic method to learn the meta-knowledge, which has attracted lots of attention. Due to the large computational cost of second-order derivative, first-order methods are thus developed [11, 37] to reduce the cost. Later on, meta-learning is introduced into DG to learn how to transfer generalizable representation across
domains. Earlier methods employ meta-learning to perform regularization. For example, MLDG [25] utilizes the episodic training paradigm and updates the network with simulated meta-train and meta-test data. MetaReg [1] explicitly learns regularization weights with episodic training. Feature-critic [30] designs a critic loss to ensure that the updated network should perform better than the original network. Currently, most methods [9, 10, 40] simply apply the episodic training scheme to train their models.

Although these methods can alleviate overfitting, by modeling only on a single task, they may produce biased optimization direction during training. Our method could well address this issue by employing our proposed multi-view regularized meta-learning.

Test Time Augmentation. The test time augmentation (TTA) is originally proposed in [23], which integrates the predictions of several augmented images for improving the robustness of final prediction. However, TTA increases the computational cost in the test stage. Recent methods try to reduce the cost by designing adaptive versions of TTA [21, 33]. Besides, some methods [34, 46] try to learn automatic augmentation strategy. However, according to our best knowledge, TTA has not been explored in DG, and it can alleviate the uncertainty of prediction via generating multi-view augmented test images.

3. Our Method

3.1. Espisodic Training Framework

Given the data space and label space are $\mathcal{X}$ and $\mathcal{Y}$, respectively, we denote $N$ different source domains as $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N$. $\mathcal{D}_i$, $1 \leq i \leq N$, is defined on the Cartesian product $\mathcal{X} \times \mathcal{Y}$, i.e., $\mathcal{D}_i = \{x_k, y_k\}_{1}^{N_i}$, where $N_i$ is the number of samples in the $i$-th source domain $\mathcal{D}_i$. We denote the model as $f$ parameterized by $\theta$.

As previously reviewed, meta-learning based methods usually train the model with an episodic training paradigm. Being consistent to the meta-learning based methods [25] that splits the training domains into meta-train and meta-test sets at each iteration, we leave one domain out as the meta-test domain $\mathcal{D}_{te}$ and the remaining domains as meta-train domains $\mathcal{D}_{tr}$. Hereafter, we sample mini-batches from these domains and obtain meta-train data $\mathcal{B}_{tr}$ and meta-test data $\mathcal{B}_{te}$. A task is defined as a pair of them $(\mathcal{B}_{tr}, \mathcal{B}_{te})$ in traditional meta-learning. We define the cross-entropy loss on a batch $B \in \{\mathcal{B}_{tr}, \mathcal{B}_{te}\}$ with parameter $\theta$ as:

$$L_{ce}(B|\theta) = \sum_{D_i \in B} \sum_{(x_k, y_k) \in D_i} \ell(f(x_k|\theta), y_k),$$

where $\ell$ is the traditional cross-entropy loss.

A model is first trained on meta-train data $\mathcal{B}_{tr}$, with meta-train loss $L_{ce}(\mathcal{B}_{tr}|\theta)$. The parameter $\theta$ is updated as $\theta' = \theta - \alpha \nabla_{\theta} L_{ce}(\mathcal{B}_{tr}|\theta)$, where $\alpha$ is the learning rate in the meta-train stage. Then we use the updated parameter $\theta'$ to test the meta-test data $\mathcal{B}_{te}$ and obtain meta-test loss $L_{ce}(\mathcal{B}_{te}|\theta')$.

Finally, we update the model with both meta-train and meta-test losses as follows:

$$\theta_{\text{new}} = \theta - \beta \nabla_{\theta}(L_{ce}(\mathcal{B}_{tr}|\theta) + L_{ce}(\mathcal{B}_{te}|\theta')),$$

where $\beta$ is learning rate. Intuitively, this training paradigm acts as simulating domain shift during the training stage. The meta-test loss computes the second derivative to the original weights, which can be viewed as a regularization term that implicitly aligns the gradients between meta-train and meta-test data [37, 44].

However, this training paradigm has several drawbacks. Firstly, its way of computing second-order gradients bears a high computational cost. Secondly, it trains the model with a single task, which might produce biased and noisy optimization direction. Thus, it still owns a risk of overfitting.

3.2. Multi-view Regularized Meta-Learning

To prevent overfitting problem and meanwhile reduce high computational cost, we develop a simple yet effective multi-view regularized meta-learning (MVRML) algorithm. We adopt Reptile [37] as base algorithm—a first-order meta-learning framework that simplifies the meta-learning process and reduce the computational cost. It first obtains the future updated model with meta-test loss:

$$\theta_{\text{future}} = \theta' - \alpha \nabla_{\theta} L_{ce}(\mathcal{B}_{te}|\theta').$$

And then, the final model parameter is calculated by taking a small step towards the future model parameters:

$$\theta_{\text{new}} = \theta + \beta(\theta_{\text{future}} - \theta).$$

In this work, we treat Eq. (1) and Eq. (2) as regularization operations to alleviate the overfitting issue. They involve the following merits: 1) by training a model with sampled tasks, we know the landscape of its weight space and where the model will arrive in the future, and thus it can help us
find a suitable optimization direction to update the model.  
2) As we only take a small step towards the optimization target, we can reduce the risk of overfitting and move to the target weight space more efficiently.  

However, in the traditional meta-learning procedure, we only train the model using a single task at a time. As the sampled task is a partial view of the source domains, training with it only explores a small part of weight space. Besides, it is easy to obtain a noisy and biased direction to the optimum weight space. Aiming at better exploring the weight space to obtain a more accurate optimization direction and erase the impact of overfitting, we propose to perform multi-view regularization in the meta-learning framework by exploiting multi-view information at each iteration. To be specific, we first train multiple tasks sequentially to find a suitable optimization direction, and then update the model with Eq. (2), as shown in Algorithm 1 and Fig. 2.

Besides, if we change the formulation of Eq. (2) as:  
\[ \theta_{\text{new}} = (1 - \beta) \theta + \beta \theta_{\text{future}}, \]
we can obtain an ensemble algorithm that combines the weights of the future and current model. Therefore, this training paradigm implicitly ensembles models in the weight space and can lead to a more robust model [19].

It is worth noting that, after each epoch, we re-estimate batch normalization (BN) [18] statistics by forwarding the training dataset. We notice that the test accuracy is unstable, which is caused by BN statistics. During training, we only update the model weights, leaving BN statistics unchanged. It does not matter in the training stage since BN employs the statistics calculated from training data. However, in the test stage, BN uses statistics accumulated in the training stage, which is calculated by the model before updating (i.e., \( \theta \)) instead of the updated model \( \theta_{\text{new}} \). Thus, when we use the weights (of already updated model) and the BN statistics (of the model without updating), their distributions do not match, leading the test performance fluctuates.

### 3.3. Multi-view Prediction

Although we can alleviate the inferior generalization result caused by the overfitting problem in the training stage, it still exists and will cause prediction instability and performance degradation in the test stage.

As our model is trained in source domains, the feature representations of learned data are well clustered. However, when unseen images come, because of the overfitting and domain discrepancy, they are more likely to be near the decision boundary, leading their feature representations unstable, as shown in Fig. 6 in Sec. 4. When we apply small perturbations to the test images, their feature representations will be pushed across the boundary, as shown in Fig. 1.

However, current test images only have a single view (i.e., the original image) with limited information. As a result, it cannot completely prevent the unstable prediction caused by overfitting. Besides, we argue that different views of a single image could bring in more information compared to the single view. Therefore, instead of only using a single view to conduct the test, we propose to perform multi-view prediction (MVP). By performing multi-view predictions, we can integrate complementary information of these views and obtain a more robust and reliable prediction. Assuming we have an image \( x \) to be tested, we can generate different views of this image with some stochastic weak transformations \( T(\cdot) \). Then the image prediction \( p \) is obtained by:

\[ p = \text{softmax} \left( \frac{1}{m} \sum_{i=1}^{m} T(f(x|\theta)) \right), \]

where \( m \) is the number of views for a test image. Note that we only apply the weak transformations (e.g., random flip) for MVP because we find that the strong augmentations (e.g., the color jittering) make the augmented images drift off its original manifold, resulting in unsatisfactory prediction accuracy. We will verify it in Sec. 4.

### 4. Experiments

We now report both quantitative and qualitative results of our evaluation. Specifically, we first describe the details of datasets and implementation. Then, we extensively compare our method with state-of-the-art methods. Moreover, we conduct the ablation study to confirm the effectiveness of each module used in our framework. Lastly, we analyze the properties of our method systematically.

### 4.1. Datasets

To evaluate the performance of our method, we consider the popularly used domain generalization datasets:

```markdown
Algorithm 1 Multi-view Regularized Meta-Learning
Input: source data \( D_{\text{src}} \), network parametrized by \( \theta \), hyperparameters: inner loop learning rate \( \alpha \) outer loop learning rate \( \gamma \) and the number of inner loops \( s \).
Output: the trained network
1: while not converged do
2: \( \theta_0 \) ← Initialized by \( \theta \)
3: for \( i \in \{1, \ldots, s\} \) do
4: Random split \( D_{\text{src}} \):
5: \( D_{\text{tr}} \cup D_{\text{te}} = D_{\text{src}}, D_{\text{tr}} \cap D_{\text{te}} = \emptyset \)
6: Sample mini-batch \( B_{\text{tr}}, B_{\text{te}} \) from \( D_{\text{tr}}, D_{\text{te}} \)
7: \( \theta_{i-1}' \leftarrow \theta_{i-1} - \alpha \nabla_\theta \left( L_{\text{ce}}(B_{\text{tr}}; \theta_i) \right) \)
8: \( \theta_i \leftarrow \theta_i' - \alpha \nabla_\theta \left( L_{\text{ce}}(B_{\text{te}}; \theta_i') \right) \)
9: end for
10: \( \theta \leftarrow \theta_s + \beta (\theta_s - \theta_0) \)
11: Re-estimate BN statistics
12: end while
```
• PACS [24] contains 9,991 images with 7 classes and 4 domains (i.e., Photo, Art Paintings, Cartoon, and Sketches), and there is the large distribution discrepancy across domains. We apply the official split to conduct the experiment for a fair comparison.

• VLCS [48] consists of 10,729 images including 5 classes and 4 domains (i.e., CALTECH, LABELME, PASCAL, SUN) with the small domain gap. Also, we use the official split in the experiment.

• OfficeHome [49] contains 15,500 images, covering 4 domains (Art, Clipart, Product and Real-World) and 65 categories which are significantly larger than PACS and VLCS dataset, which is more challenging. Following [24], we randomly split OfficeHome dataset into 90% training set and 10% validation set.

Implementation details. We choose ResNet-18 [14] pretrained on ImageNet [8] as our backbone. All images are resized to 224×224, and the batch size is set to 64. The data augmentation consists of random resize and crop with an interval of [0.8, 1], random horizontal flip, and random color jittering with a ratio of 0.4. The model is trained for 30 epochs. We use SGD as our outer loop optimizer and Adam as the inner loop optimizer, both with a weight decay of 5e − 4. Their initial learning rates are 0.05 and 0.001 for the first 24 epochs, respectively, and they are reduced to 5e − 3 and 1e − 4 for the last 6 epochs. The β1 and β2 are 0.9 and 0.999 for Adam optimizer, respectively. For our multi-view prediction, we only apply weak augmentations, i.e., the random resized crop with an interval of [0.8, 1] and random horizontal flip. The augmentation number t is set to 32. If not specially mentioned, we adopt this implementation as default. We also utilize our method to a strong baseline by applying RandAugment [7] (denoted as SBL) to further improve the effectiveness of the model. This way augments an image by randomly selecting from 14 transformations. We set the number of transformations N and the magnitude of transformation M to 5 and 4, respectively.

We adopt the leave-one-out [24] experimental protocol that leaves one domain as an unseen domain and other domains as source domains. We conduct all experiments five times and average the results for the final results. Following the way in [24], we select the best model on the validation set. DeepAll indicates that the model is trained without any other domain generalization modules.

4.2. Comparison with state-of-the-art Methods

We evaluate our method on several benchmarks and compare it to different kinds of recent state-of-the-art DG methods to demonstrate its effectiveness.

PACS. We perform evaluation on PACS with ResNet-18 and ResNet-50 as our backbone. We compare with several meta-learning based methods (i.e., MLDG [25], MASF [9], MetaReg [1]), augmentation based methods (i.e., MixStyle [61], FACT [53], FSDCL [20]), ensemble learning based methods (i.e., DSON [42], SWAD [5]) and domain-invariant feature learning (i.e., VDN [51]). As shown in Tab. 1, our method can surpass traditional meta-learning methods even without using multi-view prediction (MVP) by 3.25% (84.95% in Tab. 4 vs. 81.70%) on ResNet-18 and 4.55% (87.22% vs. 82.67%) on ResNet-50. Besides, our method also achieves SOTA performance compared to recent other methods. Note that the improvement of our method on the hardest “sketch” domain is significant compared to the DeepAll (i.e., 83.55% vs. 66.21%), owing to its better regularization and robustness.

VLCS. To verify the trained model can also general-
Table 3. Domain generalization accuracy (%) on OfficeHome dataset. The best performance is marked as bold.

| Method          | A    | C    | P    | R    | Avg. |
|-----------------|------|------|------|------|------|
| DeepAll         | 58.65| 50.35| 73.74| 75.67| 64.60|
| JiGen [4]       | 53.04| 47.51| 71.47| 72.29| 61.20|
| DSON [42]       | 59.37| 45.70| 71.84| 74.68| 62.90|
| RSC [17]        | 58.42| 47.90| 71.63| 74.54| 63.12|
| CrossGrad [43]  | 58.40| 49.40| 73.90| 75.80| 64.38|
| DAEL [60]       | 59.40| 55.10| 74.00| 75.70| 66.10|
| FSDCL [20]      | 60.24| 53.54| 74.36| 76.66| 66.20|
| FACT [53]       | 60.34| 54.85| 74.48| 76.55| 66.56|
| Ours            | 61.44| 51.16| 74.72| 77.72| 66.26|
| Ours (SBL)      | 61.46| 57.03| 73.86| 76.43| 67.20|

As seen in Tab. 2, our method is comparable to recent SOTA methods and achieves the best performance in two domains (CALTECH and PASCAL), demonstrating that our method can also perform well in this case.

**OfficeHome.** We compare our method with SOTA methods on OfficeHome to prove the adaptation of our method to the dataset with a large number of classes. The result is reported in Tab. 3. Our method is able to achieve comparable performance to current SOTA methods. When employing the strong baseline, our method can break the SOTA records by 0.94% (66.26% vs. 67.20%).

### 4.3. Ablation Study

To further investigate our component, we conduct an ablation study on each component: *i.e.*, 1) Reptile implementation of Meta-Learning method (ML), 2) multi-view regularized meta-learning (MVRML), 3) multi-view prediction (MVP). We also perform the ablation study on a strong baseline to validate the generalization ability of our method.

**Reptile:** As seen in Tab. 4, the performance of Reptile can achieve satisfactory performance compared to DeepAll. Note that, although its performance in “sketch” domain improves a lot (*i.e.*, (77.65% vs. 66.21%)), the performance in the “photo” domain decreases. We hypothesize that the feature spaces learned by meta-learning and DeepAll are different. Since ResNet-18 is pretrained on ImageNet (photolike dataset), it shows a high performance in the “photo” domain at the beginning. When the training procedure continues, the model is hard to move far away from its initial weight space. Thus its performance is promising in the “photo” domain. However, when trained with meta-learning, it can obtain a good performance by the episodic training scheme but with a little sacrifice of its original performance in the “photo” domain.

When Reptile is applied to a strong baseline (*i.e.*, with strong data augmentation), it does not improve the performance compared to DeepAll. Then we check the last checkpoint during training and calculate its average performance.

| Method          | A    | C    | P    | R    | Avg. |
|-----------------|------|------|------|------|------|
| DeepAll         | 58.65| 50.35| 73.74| 75.67| 64.60|
| +ML             | 80.49| 76.23| 94.91| 77.65| 82.34|
| +MVRML          | 83.69| 79.29| 94.87| 81.95| 84.95|
| +MVRML+MVP      | 84.59| 79.22| 95.38| 83.55| 85.69|
| Ours (SBL)      | 84.68| 79.84| 94.90| 86.78| 86.55|

We find that selecting the model in the last epoch can improve the performance by 0.74% (83.74% vs. 84.48%). Note that our model selection policy selects the best model on the validation set, whose distribution is the same as the source domains. Thus, when the model overfits to the source domains, this policy has more chance to select the overfitted model than the model that can generalize better (but may perform not very well in source domains). Therefore, the overfitting problem hurts the model selection and degrades the performance of Reptile.

**Multi-view regularized meta-learning:** When we apply multi-view regularized meta-learning (MVRML), the performance is improved a lot on the baseline in Tab. 4. It shows its efficacy of dealing with the overfitting issue. We observe that the “photo” domain also decreases a little. Considering the discussion above, it may be caused by the better weight space produced by meta-learning algorithm, which is far away from the initial weight space (*i.e.*, the initial model trained by ImageNet).

**Multi-view prediction:** Finally, we employ multi-view prediction (MVP) to enhance model reliability. As shown in Tab. 4, the performance improves on both baselines. We notice that there is a large improvement in the “sketch” domain because the “sketch” domain has only the outline of the object, and thus it is more sensitive to the small perturbations. With MVP, the prediction of the model can be more reliable and accurate.

### 4.4. Further Analysis

We further analyze the properties of our method. Our model is ResNet-18 without using strong augmentation.

**Reptile vs. traditional meta-learning:** To verify
whether Reptile can reduce the computational cost, we compare it with traditional meta-learning. As seen from Tab. 5, although its performance is similar to the second-order meta-learning, its training time (i.e., 12 min. vs. 22 min.) and memory consumption (i.e., 5G vs. 14G) are largely decreased due to the better training paradigm.

**Longer training:** In our training scheme, we need to train a model with more tasks than traditional meta-learning, resulting in a longer training time. To investigate whether it improves the performance, we train Reptile with a large epoch (i.e., 120 epochs). As shown in Fig. 3a, the longer training does not bring any drastic performance gain compared to the original training epochs, which also validates the efficacy of our method.

**Impact of the number of tasks:** For multi-view regularized meta-learning (MVRML), we sample a sequence of tasks to train the model. The larger number of tasks, the better weight space might be found. Therefore, we train our model with a different number of tasks, and the result is shown in Fig. 3b. With the increasing number of tasks, the performance first improves and then plateaus when the number of tasks is larger than 4. We suspect that with more tasks to be learned, the optimization direction does not change too much. Thus, more tasks cannot largely improve the performance.

**The effectiveness of re-estimating BN:** As mentioned in Sec. 3.2, when we train the model using Reptile, the training process is unstable because the optimization procedure of Reptile is only performed on the model weights instead of the BN statistics, which causes a distribution mismatch between them. We plot the mean (the solid line) and standard deviation (the shaded area) of model accuracy in the target domain. As seen on the orange line of Fig. 4, the performance of the model trained without re-estimating BN statistics fluctuates violently, making it hard to select the best model on validation set. However, after we re-estimate BN statistics at the end of each epoch, its accuracy becomes more stable, and a clear performance gains can be obtained in the “art” domain, as seen on the blue line.

**Unstable prediction:** In the previous sections, we argue that if the model overfits to the source domains, it is easy to produce unstable predictions by perturbing the test images slightly (random resized crop and flip). By contrast, a robust model can reduce this effect and perform well. To verify it, we test several models in the unseen domain: DeepAll without weak augmentations (i.e., color jittering, random crop and flip), DeepAll and the model trained with MVRML. We introduce prediction change rate (PCR), which is calculated by the ratio of the number of predictions changed after applying the augmentations and the number of total predictions. We compare the test accuracy and PRC in Tab. 6. The larger this measure, the more unstable the model in the unseen domains. As seen, DeepAll without augmentation produces the highest PCR and lowest Acc because this model overfits to source domains. Meanwhile, with data augmentation and a better training strategy, the performance of the model largely improves and PCR decreases drastically.

**MVP on SOTA methods:** MVP is a plug-and-play method, and it can be easily adapted to other methods. To validate its adaptation ability, we integrate it into three SOTA methods, i.e., Mixstyle [61], RSC [17] and FSR [52]. For RSC and FSR, we directly use their pre-trained model. For MixStyle, we implement it by ourselves. The result is shown in Tab. 7. MVP can improve all of these trained
The number of augmented images is set to 32. In Tab. 8, we validated the effectiveness of our method. Afterwards, during testing, we introduced multi-view prediction to generate different views of a single image set in the source domain and all images in the test domain to obtain the visualization result. The better the model can generalize, the more clustered the data should be. As shown in Fig. 6, DeepAll cannot well cluster the unseen samples since the plain training cannot prevent the model from overfitting. By contrast, MVRML can yield better clustering results, demonstrating its generalization ability.

5. Conclusion

In this paper, to resist overfitting, with the observation that the performance in DG models could be benefited by task-based augmentation in training and sample-based augmentation in testing, we propose a novel multi-view framework to boost generalization ability and reduce unstable prediction caused by overfitting. Specifically, during training, we designed a multi-view regularized meta-learning algorithm. Afterwards, during testing, we introduced multi-view prediction to generate different views of a single image for ensemble to stabilize its prediction. By conducting extensive experiments on three DG benchmarks datasets, we validated the effectiveness of our method.
References

[1] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In NeurIPS, 2018. 3, 5
[2] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. Advances in neural information processing systems, 19:137, 2007. 1
[3] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In NeurIPS, 2016. 2
[4] Fabio M Carlucci, Antonio D’Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In CVPR, 2019. 2, 5, 6
[5] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. arXiv preprint arXiv:2102.08604, 4, 2021. 5
[6] Prithvijit Chattopadhay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. In European Conference on Computer Vision, pages 301–318. Springer, 2020. 2
[7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. RandAugment: Practical automated data augmentation with a reduced search space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 702–703, 2020. 5
[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, 2009. 5
[9] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In NeurIPS, 2019. 1, 2, 3, 5
[10] Yingjun Du, Jun Xu, Huan Xiong, Qiang Qi, Xiantong Zhen, Cees GM Snoek, and Ling Shao. Learning to learn with variational information bottleneck for domain generalization. In European Conference on Computer Vision, pages 200–216. Springer, 2020. 3
[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In ICML, 2017. 2
[12] Chuang Gan, Tianbao Yang, and Boqing Gong. Learning attributes equals multi-source domain generalization. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 87–97, 2016. 2
[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In NeurIPS, 2014. 2
[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016. 5
[15] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. arXiv preprint arXiv:1912.02781, 2019. 8
[16] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In ICCV, 2017. 2
[17] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. 2020. 5, 6, 7, 8
[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv, 2015. 4
[19] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitrii Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. arXiv preprint arXiv:1803.05407, 2018. 4
[20] Seogkyu Jeon, Kibeom Hong, Pihyeon Lee, Jewook Lee, and Hyeran Byun. Feature stylization and domain-aware contrastive learning for domain generalization. In Proceedings of the 29th ACM International Conference on Multimedia, pages 22–31, 2021. 2, 5, 6
[21] Ildoo Kim, Younghoon Kim, and Sungwoong Kim. Learning loss for test-time augmentation. arXiv preprint arXiv:2010.11422, 2020. 3
[22] Wouter M Kouw and Marco Loog. A review of domain adaptation without target labels. IEEE transactions on pattern analysis and machine intelligence, 43(3):766–785, 2019. 1
[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25:1097–1105, 2012. 3
[24] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In ICCV, 2017. 5
[25] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In AAAI, 2018. 1, 2, 3, 5, 6
[26] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In CVPR, 2018. 1, 2
[27] Lei Li, Ke Gao, Juan Cao, Ziyao Huang, Xiyue Mi, Zhengze Yu, Xiaoya Li, and Boyang Xia. Progressive domain expansion network for single domain generalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 224–233, 2021. 2
[28] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for domain generalization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 8886–8895, 2021. 2
[29] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In ECCV, 2018. 2
[30] Yiyang Li, Yongxin Yang, Wei Zhou, and Timothy M Hospedales. Feature-critic networks for heterogeneous domain generalizations. arXiv, 2019. 3
[31] Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In AAAI, 2020. 5
[32] Luke Melas-Kyriazi and Arjun K Manrai. Pixmatch: Unsupervised domain adaptation via pixelwise consistency training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12435–12445, 2021.

[33] Luca Mecerino, Roberto G Rizzo, Valentino Peluso, Andrea Calimera, and Enrico Macii. Adaptive test-time augmentation for low-power cpu. arXiv preprint arXiv:2105.06183, 2021.

[34] Dmitry Molchanov, Alexander Lyzhov, Yuliya Molchanova, Arsenni Ashukha, and Dmitry Vetvov. Greedy policy search: A simple baseline for learnable test-time augmentation. arXiv preprint arXiv:2002.09103, 2020.

[35] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In ICML, 2013.

[36] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1094–1103, 2021.

[37] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. arXiv preprint arXiv:1803.02999, 2(3):4, 2018.

[38] Oren Nuriel, Sagie Benaim, and Lior Wolf. Permutated adain: Enhancing the representation of local cues in image classifiers. arXiv e-prints, pages arXiv–2010, 2020.

[39] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In International Conference on Machine Learning, pages 7728–7738. PMLR, 2020.

[40] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In CVPR, 2020.

[41] Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Bakhtashmotlagh, and Sridha Sridharan. Correlation-aware adversarial domain adaptation and generalization. PR, 2019.

[42] Seonguk Seo, Yumin Suh, Dongwan Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. ECCV, 2020.

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

[44] Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. arXiv preprint arXiv:2104.09937, 2021.

[45] Manli Shu, Zuxuan Wu, Micah Goldblum, and Tom Goldstein. Prepare for the worst: Generalizing across domain shifts with adversarial batch normalization. arXiv preprint arXiv:2009.08965, 2020.

[46] Maciej Sypetkowski, Jakub Jasulewicz, and Zbigniew Wojna. Augmentation inside the network. arXiv preprint arXiv:2012.10769, 2020.

[47] Sebastian Thrun. Lorien Pratt, editors. 1998. learning to learn.

[48] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In CVPR 2011, pages 1521–1528. IEEE, 2011.

[49] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In CVPR, 2017.

[50] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. Learning from extrinsic and intrinsic supervisions for domain generalization. In European Conference on Computer Vision, pages 159–176. Springer, 2020.

[51] Yufei Wang, Haoxiang Li, Lap-Pui Chau, and Alex C Kot. Variational disentanglement for domain generalization. arXiv preprint arXiv:2109.05826, 2021.

[52] Yue Wang, Lei Qi, Yinghuan Shi, and Yang Gao. Feature-based style randomization for domain generalization. arXiv preprint arXiv:2106.03171, 2021.

[53] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14383–14392, 2021.

[54] Zhenlin Xu, Deyi Liu, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. arXiv preprint arXiv:2007.13003, 2020.

[55] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In ICCV, 2019.

[56] Zhongqi Yue, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Transporting causal mechanisms for unsupervised domain adaptation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 8599–8608, 2021.

[57] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. Deep stable learning for out-of-distribution generalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5372–5382, 2021.

[58] Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. Domain generalization via entropy regularization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3532–3542, 2021.

[59] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. Learning from extrinsic and intrinsic supervisions for domain generalization. In European Conference on Computer Vision, pages 159–176. Springer, 2020.

[60] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Deep domain-ad image generation for domain generalisation. In AAAI, 2020.

[61] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Mixstyle neural networks for domain generalization and adaptation. arXiv preprint arXiv:2107.02053, 2021.