Order-preserving Consistency Regularization for Domain Adaptation and Generalization

Jing, M.; Zhen, X.; Li, J.; Snoek, C.G.M.

DOI
10.48550/arXiv.2309.13258
10.1109/ICCV51070.2023.01734

Publication date
2023

Document Version
Author accepted manuscript

Published in
2023 IEEE/CVF International Conference on Computer Vision

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Download date: 12 Apr 2024
Order-preserving Consistency Regularization
for Domain Adaptation and Generalization

Mengmeng Jing\textsuperscript{1,2,*}, Xiantong Zhen\textsuperscript{2,†}, Jingjing Li\textsuperscript{1}, Cees G. M. Snoek\textsuperscript{2}
\textsuperscript{1}University of Electronic Science and Technology of China
\textsuperscript{2}University of Amsterdam
jingmengl1992@gmail.com zhenxt@gmail.com lijin117@yeah.net
c.g.m.snoek@uva.nl

Abstract

Deep learning models fail on cross-domain challenges if the model is oversensitive to domain-specific attributes, e.g., lightning, background, camera angle, etc. To alleviate this problem, data augmentation coupled with consistency regularization are commonly adopted to make the model less sensitive to domain-specific attributes. Consistency regularization enforces the model to output the same representation or prediction for two views of one image. These constraints, however, are either too strict or not order-preserving for the classification probabilities. In this work, we propose the Order-preserving Consistency Regularization (OCR) for cross-domain tasks. The order-preserving property for the prediction makes the model robust to task-irrelevant transformations. As a result, the model becomes less sensitive to the domain-specific attributes. The comprehensive experiments show that our method achieves clear advantages on five different cross-domain tasks.

1. Introduction

Deep neural networks have demonstrated their power in many computer vision tasks, especially when the training and test sets follow the same distribution. However, when we deploy a model in a real-world environment, we often encounter domain shifts between the training and test sets, which reduces the expected test-set performance and makes us unable to deploy with confidence \cite{50}. For some safety-critical applications, e.g., tumor recognition \cite{22} and autonomous driving \cite{19}, a failing model is fatal.

Image data consists of a variety of attributes such as shape, color, background, texture, shooting angle, etc. We refer to one or more task-related attributes as label attributes, and the remaining irrelevant ones as domain-specific attributes. Wiles \textit{et al.} \cite{69} demonstrate the domain-specific attributes cause the distribution shifts, thus weakening the generalization of the model. Data augmentation coupled with consistency regularization is commonly employed to make a model invariant to the domain-specific attributes \cite{68, 8, 27, 59, 4, 10, 11, 21}. Data augmentation perturbs the data so that the domain-specific information is incorporated into the perturbed image. By imposing a consistency regularization on the representations of the same image before and after perturbation, the model becomes less sensitive to domain-specific attributes.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{The required output of three consistency regularizations. Different shapes represent different categories. For different greens, the darker the color, the larger the classification probability. Representation-based method requires the output to be the same as the original. OCR only requires an order-preserving output and allows the output to vary. The prediction-based method is not order-preserving, which may cause the probability of the horse being classified to mouse is higher than that of donkey although donkeys are obviously more similar to horses than mice.}
\end{figure}
The existing consistency regularization methods can be divided into two categories: representation-based methods [32, 61, 57] and prediction-based methods [4, 44, 71]. For the representation-based methods, usually the $\ell_1$ or $\ell_2$ loss is employed to enforce the model to output the same representation, even though two different views are fed into the model. This constraint, however, is too strict, which may bring difficulties to the training of the model. For example, different works on self-supervised learning [10, 11, 21] have reached a consensus that one of the representations needs to go through a non-linear prediction head before performing consistency regularization with the other. With the network model being a symmetric structure, directly imposing consistency regularization on the two representations will result in a model collapse.

Alternatively, the prediction-based methods [4, 44, 71] employ the cross-entropy loss to regularize the maximum classification probability of two representations to be the same. In other words, they ignore the order of the other classes, which would reduce the discriminability of the model. For example, consider a classification problem of three classes: horse, donkey and mouse. As illustrated in Fig. 1, for an image of a horse, the cross-entropy loss only regularizes the maximum classification probability of two representations to be horse, but it ignores the classification probability of donkey and mouse. If the order of classification probability is horse > donkey > mouse before augmentation, it may become horse > mouse > donkey after augmentation. Although the classification results have not changed, the discrimination of the model has reduced as donkeys are obviously more similar to horses than mice.

In view of these problems, we propose Order-preserving Consistency Regularization (OCR) for cross-domain tasks. OCR is able to enhance the model robustness to domain-specific attributes without the need of an asymmetric architecture or a stop gradient. Specifically, we compute the residual component which is the variation in the augmented representation relative to the original representation. We postulate that if the model is robust to domain-specific attributes, the residual component should contain little or no task-related information. For example, in the classification task, when we classify the residual component, we regularize it to have the same probability to be classified into each category. In this way, the classification probabilities of the augmented representation are order-preserving compared to the original representation. As a result, the model becomes less sensitive to the domain-specific attributes. The core idea of OCR is that we allow the model to output different representations for two views of the same image, as long as the residual component contains as little task-related information as possible.

The contributions of this paper are threefold:

1. We propose Order-preserving Consistency Regularization (OCR) to enhance model robustness to domain-specific attributes. Compared with representation-based methods, OCR relaxes the constraints on model training, i.e., it allows the model to output different representations for two views of the same image. Compared with prediction-based method, OCR maintains the order of the classification probabilities before and after augmentation, which helps the model to be less sensitive to the domain-specific attributes.

2. We provide a theoretical analysis for OCR. We prove that the representation-based method is a special case of OCR. Moreover, OCR can reduce the mutual information between the domain-specific attributes and the label attribute.

3. We test our method on five different cross-domain vision tasks to demonstrate the effectiveness of our method. In particular, OCR helps to enhance the robustness of the model against adversarial attacks.

2. Related Work

Consistency Regularization. Consistency regularization [2, 32, 57] is a common self-supervised learning method which enforces the model to output the same prediction even when the input is perturbed. Since it can enhance the robustness of the model to domain-variant styles, it has recently been used to address cross-domain challenges [8, 68]. To generate the perturbed version of the image, some methods employ adversarial training [44] or dropout [32, 61], while others add perturbations by applying heuristic data augmentations [32, 57, 5, 71], such as color jitter, Gaussian blur, rotate, cutout, etc. To measure the consistency, the $\ell_1$ or $\ell_2$ norm [32, 61, 57] are adopted. Given the images of the original version and the perturbed version, some methods [71, 57, 8] employ the same model to extract representations for the two images, and then impose consistency regularization. We believe that this strategy is too strict, thereby increasing the difficulty of model training.

To solve this problem, some works [10, 11, 21] have designed an asymmetric architecture for the model, where one representation needs to go through an additional non-linear layer, which makes two images go through different paths in the same model. Although effective, these methods increase the complexity of the model architecture, and often require a large number of training data to unleash their power. Another line of work feeds one of the images (usually the original version) into the running average model or past model and then applies consistency regularization [32, 61]. These methods allow two versions of the images to pass through two similar yet different models, alleviating the problem of overly strict regularization to some extent. However, these methods require the storage of multiple copies of the model,
thereby increasing the GPU memory consumption. Different from the above methods, our method does not require an asymmetric architecture, nor does it need to store multiple copies of the models. Our method allows the model to output different representations for two versions of the images, as long as the residual component obtained from these two representations does not contain task-related information.

**Domain Adaptation and Generalization.** Both domain adaptation (DA) [41, 56, 38, 15, 63, 79] and domain generalization (DG) [46, 18, 49, 45, 48] are cross-domain tasks, but their task settings are different. In the DA task, we are given a labeled source domain and an unlabeled target domain. We use the joint training of source and target samples to make the model adapt to the shifts between domains. The recent focus on privacy and copyright has given rise to OCR, which increases the divergence of samples and helps to improve the generalization of the model. Data augmentation [2, 57, 32, 44, 71] is a commonly used technology, which increases the diversity of samples and helps to improve the generalization of the model. Given a sample \( x_o \), we obtain its augmented version \( x_a = \mathcal{N}(x_o) \) using transformations \( \mathcal{N} \). For a clearer narration, we split the hypothesis \( h \) into two parts, i.e., \( h = F \circ G \), where \( G \) is the backbone model and \( F \) is the classifier. We feed \( x_o \) and \( x_a \) into \( G \) to get two different repre-

**3. Methodology**

**Problem Formulation.** In many computer vision challenges, be it image classification or semantic segmentation, we are given a dataset \( D_{train} = \{(x, y) \mid x \in \mathcal{X}_{train}, y \in \mathcal{Y}_{train}\} \) where \( \mathcal{X}_{train} \) and \( \mathcal{Y}_{train} \) are the image set and label set for training and we need to establish the relationship between the data \( \mathcal{X}_{train} \) and the ground-truth label \( \mathcal{Y}_{train} \). In the classical Empirical Risk Minimization (ERM) [64], the training objective is to choose a hypothesis \( h : \mathcal{X} \to \mathcal{Y} \) from a predefined hypothesis space \( \mathcal{H} \). However, when deployed to the test set \( D_{test} \), the model would suffer from performance degradation since there may be domain shifts between the training set \( D_{train} \) and the test set \( D_{test} \), i.e., \( P(\mathcal{X}_{train}) \neq P(\mathcal{X}_{test}) \). For example, samples of the same category in the training set and the test set often have varying appearance, caused by lightning conditions, camera angle, background, etc. These accidental attributes are irrelevant to our task, but will cause domain shifts. To generalize well, we need to train the model to be invariant to these domain-specific attributes.

**Order-preserving Consistency Regularization.** For a global understanding, we provide the overview of our method in Fig. 2. OCR consists of three steps, i.e., data augmentation, residual component separation, and residual entropy maximization. Data augmentation [2, 57, 32, 44, 71] is a commonly used technology, which increases the diversity of samples and helps to improve the generalization of the model. Given a sample \( x_o \), we obtain its augmented version \( x_a = \mathcal{N}(x_o) \) using transformations \( \mathcal{N} \). For a clearer narration, we split the hypothesis \( h \) into two parts, i.e., \( h = F \circ G \), where \( G \) is the backbone model and \( F \) is the classifier. We feed \( x_o \) and \( x_a \) into \( G \) to get two different repre-

**Figure 2. Method overview.** Given the original image and its augmented counterpart, we feed them into the backbone model to obtain the original representation \( z_o \) and the augmented representation \( z_a \). Then, we compute their residual component \( z_n \) and feed it into the classifier to get the classification results. Finally, we maximize the entropy of the classification probabilities for the residual component to reduce the task-related information in the residual component. As a result, the model become less sensitive to the domain-specific attributes.
sentations of the same sample: \( z_o = G(x_o), z_a = G(N(x_o)) \).

We define the residual component as the variation in the augmented representation relative to the original representation. To separate the residual component, an intuitive method is to subtract the original representation from the augmented representation. In order to control the proportion of the residual component more flexibly, here we consider the following linear relations:

\[
z_a = \lambda z_o + (1 - \lambda) z_n, \tag{1}
\]

where \( z_n \) is the residual component, \( \lambda \in (0, 1) \) represents the proportion of \( z_o \) in \( z_a \). From the perspective of Occam’s razor, linearity is a good inductive bias, as also used in mixup [77]. Another reason we choose the relation in Eq. (1) is that it is an invertible operation so that we can easily infer \( z_n \) given \( z_a \) and \( z_o \):

\[
z_n = \frac{z_a - \lambda z_o}{1 - \lambda}. \tag{2}
\]

With the residual component \( z_n \), we try to maximize the uncertainty of \( z_n \)'s prediction so that it does not contain too much classification-related information. As the entropy can be regarded as the measure of the prediction uncertainty, we maximize the conditional entropy \( \mathcal{H}(y|z_n) \) to enlarge the uncertainty of \( z_n \)'s prediction. Therefore, our objective is as follows:

\[
\mathcal{L}_{OCR} = -\mathcal{H}(y|z_n) = -\mathcal{H}([\text{Softmax}(F(z_n))]), \tag{3}
\]

where \( F(z_n) \in \mathbb{R}^{B \times C} \) is the prediction of \( z_n \), \( B \) is the batch size, \( C \) is the category number. \( \mathcal{H} \) is the entropy. By minimizing Eq. (3), we regularize \( z_n \) to have equal probability of being classified into each category.

During training, we use \( \lambda \) to control the proportion of the residual component and the original representation in the augmented representation. \( \lambda \) should change dynamically to match the process of model training. At the beginning of training, the model would be sensitive to the domain-specific attributes, so the difference between \( z_o \) and \( z_n \) would be large. Then, \( \lambda \) should be a small value so that the proportion of \( z_o \) in \( z_a \) is lower. As the training goes on, the model gradually becomes less sensitive to the domain-specific attributes, at this time, \( z_o \) and \( z_n \) would be similar and \( \lambda \) should increase to a larger value accordingly. Inspired by Ganin et al. [16], we adopt an annealing strategy for \( \lambda \):

\[
\lambda = \lambda_0 [1 - (1 + \frac{t}{T})^{-\beta}], \tag{4}
\]

where \( \alpha = 10, \beta = 0.75, t \) is the current iteration number and \( T \) is the total number of iterations. \( \lambda_0 \) is the initial value of \( \lambda \). In this way, \( \lambda \) is more likely to be sampled to a small value at the beginning of training and then gradually becomes larger as the training goes on. In the ablations we illustrate that this strategy could achieve better performance than that of a fixed \( \lambda \) value.

Now we prove three properties of the proposed OCR:

**1. OCR is order-preserving.** In previous methods with consistency regularization, e.g., [8, 57, 71], the similarity between the representations and the prototype feature of a class in classifier \( F \) is computed as:

\[
\hat{y}_a^i = \text{sim}(P_i, z_a), \hat{y}_o^i = \text{sim}(P_i, z_a), \tag{5}
\]

where \( \text{sim}(\cdot) \) is the similarity function, i.e., the inner-product, \( P_i \) is the prototype feature of class \( i \), \( \hat{y}_a^i \) and \( \hat{y}_o^i \) are probabilities of representations \( z_a \) and \( z_o \) belonging to class \( i \), respectively. When substituting Eq. (1) into Eq. (5), we get:

\[
\hat{y}_a^i = \lambda \text{sim}(P_i, z_o) + (1 - \lambda) z_n \]
\[
= \lambda \text{sim}(P_i, z_o) + (1 - \lambda) \text{sim}(P_i, z_n) \]
\[
= \lambda \hat{y}_a^i + (1 - \lambda) \hat{y}_o^i. \tag{6}
\]

In Eq. (3), when the conditional entropy \( \mathcal{H}(y|z_n) \) is maximized, the residual component will have equal probability of being classified into each category, i.e., \( \hat{y}_n = \hat{y}_n^1 = \cdots = \hat{y}_n^K = K \). Therefore, the relation between \( \hat{y}_a^i \) and \( \hat{y}_o^i \) is:

\[
\hat{y}_a^i = \lambda \hat{y}_a^i + (1 - \lambda) K = f(\hat{y}_o^i; \lambda, K). \tag{7}
\]

Within the same iteration, \( K \) and \( \lambda \) are two constants. Then, \( f(\hat{y}_o^i; \lambda, K) \) is an order-preserving mapping, which guarantees that if \( \hat{y}_o^i > \hat{y}_o^k \), then \( \hat{y}_a^i > \hat{y}_a^k \). Therefore, OCR is order-preserving.

**2. Representation-based consistency regularization is a special case of OCR.** Previous cross-domain methods, e.g., [71, 57, 8], optimize the \( \ell_1 \) or \( \ell_2 \) loss to impose consistency regularization between \( z_o \) and \( z_n \). We use \( z_t = z_n - z_o \) to represent the unnormalized residual component, \( \Delta y^t \) to represent the prediction of \( z_t \), belonging to class \( i \). Therefore, the objective of representation-based consistency regularization is to make \( z_t \) close to the zero vector:

\[
z_t = 0 \Rightarrow \text{sim}(P_i, z_t) = 0 \Rightarrow \Delta y^t = \cdots = \Delta y^K = 0. \tag{8}
\]

We believe this regularization is too strict and may increase the training difficulty. It is very reasonable for the model to output different representations for different images. The goal of our method is not to enforce \( z_t \) to be close to the zero vector, but to make \( z_t \) contain no task-related information. Our method relaxes the constraint in Eq. (8) as:

\[
\Delta y^1 = \Delta y^2 = \cdots = \Delta y^K. \tag{9}
\]

Obviously, \( z_t \) of the zero vector in Eq. (8) can also match the condition in Eq. (9), making representation-based consistency regularization a special case of our method.
OCR can make the model less sensitive to the domain-specific attributes. The mutual information between the residual component and the label attribute is:

\[
I(Z_n; Y) = KL[p(z_n,y)||p(z_n)p(y)] = \int dz_n dy p(z_n, y) \log \frac{p(z_n, y)}{p(z_n)p(y)}.
\]

Minimizing this mutual information can be regarded as the proxy of domain-specific attributes for the original representation. By minimizing this mutual information, we can decorrelate the domain-specific attributes and the label attribute. As a result, the problem of sensitivity to domain-specific attributes is alleviated.

According to the Information Bottleneck principle [62], an optimal representation \( z \) of input \( x \) should satisfy two properties: sufficiency and minimality. Achille and Soato [1] have demonstrated that being invariant to domain-specific attributes is helpful to guarantee the minimality. Therefore, the proposed OCR is helpful to learn a better representation, which could improve the performance of the model on cross-domain tasks.

### 4. Experiments

#### 4.1. Tasks, Datasets and Setup

To evaluate our method, we consider five different cross-domain tasks: domain adaptation, test-time adaptation, domain generalization classification, domain generalization detection, and domain generalization semantic segmentation. Different tasks involve different datasets and setups, which we summarize in Table 1.

**Domain Adaptation Classification.** For domain adaptation classification we report on Office-Home [65]. It consists of four domains: Art, Clipart, Product and Real-world. There are about 15,500 images categorized into 65 classes. We consider two different settings, i.e., source-dependent [41, 56] and source-free [38, 72]. For the source-dependent setting, we use all labeled source samples and all unlabeled target samples for training. For the source-free setting, only the model trained in the source domain and the unlabeled target samples are given. Upon evaluation, we test the models in the unlabeled target samples. For the hyper-parameter, we set \( \lambda_0 = 0.7 \).

**Test-Time Adaptation.** For test-time adaptation we report on CIFAR100-C [24]. This dataset includes 15 different corruption types with five levels of severity categorized into 100 classes. These corruptions were added to clean images from CIFAR100 [30]. There are 10,000 images for each corruption type. We used the ResNet-29 model pre-trained in the clean CIFAR100 dataset from [25]. This task involves two settings: online [67] and continual online [68]. In both settings, we conduct the experiments on CIFAR100-C in an online fashion without the need for labels. The difference between the two settings is that the online setting will initialize the model to the state of pre-training on the clean dataset before adapting to each corruption type, while the continual online setting will continuously adapt data of different corruption types. In this task, we evaluate our method on images with the highest severity, i.e., level 5. For the hyper-parameter, we set \( \lambda_0 = 0.8 \).

**Domain Generalization Classification.** For this task we report on PACS [34]. A commonly used domain generalization benchmark which includes four domains: Art Painting, Cartoon, Photo and Sketch. There are 9,991 images categorized into seven classes. We train the model on 3 of 4 domains and evaluate it on the remaining one. In this task, we set \( \lambda_0 = 0.5 \).

**Domain Generalization Semantic Segmentation.** For this task we follow the Semantic segmentation benchmark [12], which includes five datasets. GTAV [54] is a large-scale synthetic dataset consisting of 24,966 images.
Table 2. **Domain Adaptation.** Accuracy (%) on Office-Home with ResNet-50 backbone. All per-domain results are in the supplementary material. R- and P-Cons. Reg. means representation-based and prediction-based consistency regularization. Results with are implemented by us.

<table>
<thead>
<tr>
<th>Method</th>
<th>Source-Use</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCD [56]</td>
<td>64.1</td>
<td></td>
</tr>
<tr>
<td>w/ OCR [56]</td>
<td>66.6</td>
<td></td>
</tr>
<tr>
<td>CDAN [41]</td>
<td>65.8</td>
<td></td>
</tr>
<tr>
<td>w/ OCR</td>
<td>68.0</td>
<td></td>
</tr>
<tr>
<td><strong>Source-Free</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ResNet-50 [23]</td>
<td>46.1</td>
<td></td>
</tr>
<tr>
<td>Source-only</td>
<td>60.2</td>
<td></td>
</tr>
<tr>
<td>NRC [72]</td>
<td>71.9</td>
<td></td>
</tr>
<tr>
<td>w/ R-Cons. Reg.</td>
<td>71.5</td>
<td></td>
</tr>
<tr>
<td>w/ P-Cons. Reg.</td>
<td>72.1</td>
<td></td>
</tr>
<tr>
<td>w/ OCR</td>
<td>72.6</td>
<td></td>
</tr>
<tr>
<td>SHOT [38]</td>
<td>71.8</td>
<td></td>
</tr>
<tr>
<td>w/ R-Cons. Reg.</td>
<td>71.4</td>
<td></td>
</tr>
<tr>
<td>w/ P-Cons. Reg.</td>
<td>72.0</td>
<td></td>
</tr>
<tr>
<td>w/ OCR</td>
<td>72.8</td>
<td></td>
</tr>
<tr>
<td>SHOT++ [39]</td>
<td>72.8</td>
<td></td>
</tr>
<tr>
<td>w/ OCR</td>
<td>73.2</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. **Test-Time Adaptation.** Test error (%) for CIFAR100-to-CIFAR100C adaptation. The backbone model is ResNeXt-29. The corruption severity is 5. OCR can improve the baselines on both online setting and continual online setting.

<table>
<thead>
<tr>
<th>Method</th>
<th>Online</th>
<th>Continual Online</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>46.4</td>
<td>46.4</td>
</tr>
<tr>
<td>BN Adapt [37]</td>
<td>35.8</td>
<td>35.4</td>
</tr>
<tr>
<td>TENT [67]</td>
<td>34.4</td>
<td>35.6</td>
</tr>
<tr>
<td>w/ OCR</td>
<td>31.3</td>
<td>32.4</td>
</tr>
<tr>
<td>CoTTA [68]</td>
<td>36.8</td>
<td>32.5</td>
</tr>
<tr>
<td>w/ OCR</td>
<td>34.6</td>
<td>31.6</td>
</tr>
</tbody>
</table>

driving-scene images generated from the Grand Theft Auto V game. There are 19 objects in the images. **SYNTHIA** [55] is another synthetic dataset containing 9,400 photo-realistic synthetic images with a resolution of 960×720. **Cityscapes** [13] is a large-scale real-world dataset consisting of 3,450 finely-annotated images and 20,000 coarsely-annotated images collected from urban scenes of 50 cities in Germany. We use the finely-annotated set for training and testing. **BDD-100K** [75] is also a real-world dataset which consists of urban driving scene images collected from the US. We use 7,000 images for training and 1,000 images for evaluation. **Mapillary** [47] is the last real-world dataset containing 25,000 images collected from locations around the world. For this task, we follow the protocol in [12]. Specifically, the model is trained in GTA V for 40K iterations and evaluated on the remaining datasets. In this task, we set λ₀=0.1.

**Domain Adaptation Object Detection.** In this task, we report on **Cityscapes** [13] and **FoggyCityscapes** [58]. **FoggyCityscapes** [58] is a synthetic foggy dataset based on Cityscapes. Each image is rendered with a clear Cityscapes image and the depth map. There are 8 categories in both domains. For the hyper-parameter, we set λ = 0.5.

For the data augmentations used in our method, we apply RandomCrop and RandomHorizontalFlip for the original image. For the augmented images, we further apply ColorJitter, RandomGrayscale and GaussianBlur. The detailed parameters for these augmentations are in the supplementary material.

To test the effectiveness of our method, in all the cross-domain tasks, our method is inserted into the existing method as a plug-and-play module. We choose the weight of OCR through importance-weighted cross validation [60]. Our method is implemented with PyTorch [52] and MindSpore1. Code is available at https://github.com/mmjing/OCR.

1https://www.mindspore.cn
Table 5. **Domain Generalization Semantic Segmentation.** All models are trained on GTA V and evaluated on BDD100K, Cityscapes, SYNTHIA and Mapillary. We use ResNet-50 with output stride 16. Results with † are from [12]. Best mIoU (%) results highlighted in bold. OCR can improve all the baseline methods.

<table>
<thead>
<tr>
<th>Source Target</th>
<th>GTAV BDD100K</th>
<th>Cityscapes</th>
<th>SYNTHIA</th>
<th>Mapillary</th>
<th>Mean</th>
<th>Boost</th>
</tr>
</thead>
<tbody>
<tr>
<td>†DeepLabv3+ [9]</td>
<td>25.1</td>
<td>29.0</td>
<td>26.2</td>
<td>28.2</td>
<td>27.1</td>
<td></td>
</tr>
<tr>
<td>w/ OCR</td>
<td>34.7</td>
<td>34.8</td>
<td>25.1</td>
<td>39.8</td>
<td>33.6</td>
<td>6.5</td>
</tr>
<tr>
<td>†IBN-Net [51]</td>
<td>32.3</td>
<td>33.9</td>
<td>27.9</td>
<td>37.8</td>
<td>33.0</td>
<td></td>
</tr>
<tr>
<td>w/ OCR</td>
<td>34.9</td>
<td>41.7</td>
<td>27.6</td>
<td>38.7</td>
<td>35.7</td>
<td>2.7</td>
</tr>
<tr>
<td>†RobustNet [12]</td>
<td>35.2</td>
<td>36.6</td>
<td>28.3</td>
<td>40.3</td>
<td>35.1</td>
<td></td>
</tr>
<tr>
<td>w/ OCR</td>
<td>37.2</td>
<td>38.9</td>
<td>27.0</td>
<td>39.7</td>
<td>35.7</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Table 6. **Domain Adaptation Object Detection.** mAP (%) on Cityscapes → FoggyCityscapes.

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP</th>
<th>Boost</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDAT [53]</td>
<td>37.5</td>
<td></td>
</tr>
<tr>
<td>w/ OCR</td>
<td>39.1</td>
<td>1.6</td>
</tr>
<tr>
<td>SUDA [78]</td>
<td>42.8</td>
<td></td>
</tr>
<tr>
<td>w/ OCR</td>
<td>44.2</td>
<td>1.4</td>
</tr>
</tbody>
</table>

4.2. Results

**Domain Adaptation.** In Table 2, OCR is inserted as a plug-and-play module in each of the compared methods. In the source-dependent setting, OCR improves MCD [56] by 2.5% and CDAN [41] by 2.2%. In the source-free setting, OCR is still effective, it improves NRC [72] by 0.7% and SHOT [38] by 1.0%. In addition, as a comparison, OCR outperforms the prediction-based consistency regularization. We observe that the representation-based method does not offer clear advantage over the baseline NRC [72] and SHOT [38], which may be due to the strict regularization that increases the difficulty of model training. OCR is feature-based and independent of specific architectures, so it can be applied to transformer-based methods as well. We test SDAT [53] (ViT-B/16) and SDAT with OCR on Office-Home and achieve results of 84.3% and 85.0%, respectively. OCR can also achieve performance improvements on transformer-based architectures.

**Test-Time Adaptation.** In Table 3, for the online setting, OCR achieves 3.1% and 2.2% lower test errors than TENT [67] and CoTTA [68], respectively. For the continual online setting, TENT [67] and CoTTA [68] are also improved by 3.2% and 0.9% after adding OCR. This shows that OCR can enhance the robustness of the model against various types of corruptions. In fact, the augmented data can be regarded as data with corruption applied. Our OCR can effectively reduce the task-related information residing in the residual component in the augmented representations, thus enhancing the robustness of the model.

**Domain Generalization Classification.** In Table 4, OCR outperforms the vanilla ResNet-18 with a large margin. Note that Mixup and Manifold Mixup do not improve the vanilla ResNet-18. The reason why Mixup is ineffective here is because Mixup mainly encourages the model to be robust to the combination of the existing patterns, but does not enhance the ability to handle the unseen styles. MixStyle regularizes the model to be robust to the unseen
styles, however, it does not explicitly minimize the domain-specific information in the representation, leading to its worse performance than OCR. We also test our method on PACS based on ResNet-50 and SWAD [7]. OCR improves SWAD from 87.8% to 88.5%. OCR achieves consistent performance advantages on both ResNet-18 and ResNet-50.

Domain Generalization Semantic Segmentation. In Table 5, the Baseline is DeepLabV3+ [9]. OCR improves the Baseline by 6.5%. For IBN-Net [51], the improvement is 2.7%, which is also impressive. For RobustNet [12], we observe that OCR has a small improvement of 0.6%, this may be because RobustNet also enhances the generalization of the model by eliminating domain specific information. OCR, however, is different from RobustNet since RobustNet disentangles the domain-specific and domain-invariant part in the feature covariance while OCR does this based on the assumption of linear combination.

Cross-domain Object Detection. In Table 6, we report results of OCR and the compared methods on the object detection task of City [13] -> FoggyCity [58]. OCR achieves improvements of 1.6% and 1.4% compared to SDAT [53] and SUDA [78], respectively. Therefore, OCR is effective on object detection tasks.

4.3. Ablations

Parameter $\lambda$ Analysis. In Fig. 3, we provide an analysis for parameter $\lambda$. In Fig. 3 (a), we illustrate the impact of different choices for $\lambda$ on PACS, where “random” indicates we choose a random value from range (0,1) as $\lambda$ during each iteration, “fix” means we fix $\lambda$ as 0.5, “$\lambda^*$” is the opposite strategy to “BT*”. As can be seen from Fig. 3 (a), using the strategy in Eq. (4), while “1 – $\lambda$” is the opposite strategy of “$\lambda$”. As can be seen from Fig. 3 (a), using the strategy in Eq. (4) helps to train an optimal model. This result is in line with our hypothesis, i.e., at the beginning of training, there is a small domain-specific ratio in the representation, so a small $\lambda$ is required, as OCR continuously minimizes the domain-specific information, the domain-invariant part gradually increases, so a larger $\lambda$ is required. In addition, we test the performance of OCR with the simple formulation, i.e., $z_n = z_0 - z_\omega$, on PACS and achieve an accuracy of 84.0%, which is close to that of the fixed proportion setting in Fig. 3 (a), but lower than our formulation which obtains 85.5%. In Fig. 3 (b), we report the impact of different initial values $\lambda_0$ on performance. From Fig. 3 (b), we observe that the results of Office-Home do not fluctuate much with different initial values, its best $\lambda_0$ is around 0.7. The results of PACS, are more sensitive to different initial values, its best $\lambda_0$ is around 0.5. Therefore, different tasks need different initial values.

Analysis on Other Layers. By default we apply OCR to representations of the penultimate layer of the model. OCR can be applied to the representations of other layers as well. We show results with a ResNet-50 in Table 7, we observe that: (1) In the image level, OCR cannot achieve ideal results, which may be because some attributes of the sample, e.g., lighting and shooting angle, cannot be separated in the image level; (2) In general, the deeper the layer, the more effective OCR will be. Prior works [40, 74] have found that representations extracted from the shallow layers are more generalized, while the representations extracted from the deep layers show strong task relevance. Therefore, shallow representations are not suitable for applying OCR, while deep representations can eliminate domain-specific information through OCR.

Fourier Perspective. Following [73], we investigate the sensitivity of our models to high and low frequency corruptions via a perturbation analysis in the Fourier domain. We plot the Fourier heat map in Fig. 4. The pixels closer to the center in the heat map represent the impact of low frequency noise, while the pixels outward represent the impact of high frequency noise. We observe that the model trained with OCR is more robust compared with the model learned by ERM, especially in the high frequency domain. High frequency information is often introduced by styles that vary significantly across domains. Therefore, OCR can effectively eliminate the domain-specific style information.

Robustness to Adversarial Attack. In Table 8 we report the adversarial robustness of our method against various white-box attacks, including FGSM [20], BIM [31] and PGD [43]. We impose the adversarial attacks through the Adversarial Robust Tool box\(^2\). For fair comparison, we set the iteration number as 10, adversarial strength as 0.01 and step size as 0.01, all other parameters remain at their default values. Compared with ERM and prediction-based consistency regularization, OCR achieves the best robustness to all the three adversarial attacks. Especially for the iterative-based methods with more powerful attacks, OCR achieves accuracies of 61.6% and 61.7% against PGD and BIM, which is remarkably higher than ERM and prediction-based consistency regularization. The superior robustness of OCR against the adversarial attack derives from explicitly eliminating the negative effects of the domain-specific attributes which causes the domain shifts.

Effect of Order-preserving Property. In Table 9, we report the top 1 to top 5 accuracies on Office-Home. Com-

---

Table 7. Analysis on Other Layers. Accuracies (%) on Office-Home, where “Input” means we apply OCR on pixel-level and “BT*” is the bottleneck block of ResNet-50. Generally, the deeper the layer, the more effective OCR will be.

<table>
<thead>
<tr>
<th>ResNet-50 layers</th>
<th>Input</th>
<th>Conv1</th>
<th>BT1</th>
<th>BT2</th>
<th>BT3</th>
<th>BT4</th>
<th>FC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>68.5</td>
<td>69.0</td>
<td>70.4</td>
<td>70.6</td>
<td>71.2</td>
<td>72.2</td>
<td>72.8</td>
</tr>
</tbody>
</table>

---

\(^2\)https://github.com/advboxes/AdvBox
Table 8. Robustness to Adversarial Attack. Accuracy (%) on PACS after different adversarial attacks. The results are all based on the leave-one-domain-out protocol [81]. Our method is effective to enhance the model robustness to the adversarial attacks.

<table>
<thead>
<tr>
<th></th>
<th>Art</th>
<th>Cartoon</th>
<th>Photo</th>
<th>Sketch</th>
<th>Mean</th>
<th>Boost</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No attack</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERM Baseline</td>
<td>78.3</td>
<td>76.0</td>
<td>95.0</td>
<td>72.7</td>
<td>80.5</td>
<td></td>
</tr>
<tr>
<td>w/ P-Cons. Reg.</td>
<td>79.2</td>
<td>80.2</td>
<td>95.9</td>
<td>79.3</td>
<td>83.7</td>
<td>3.2↑</td>
</tr>
<tr>
<td>w/ OCR</td>
<td><strong>85.4</strong></td>
<td><strong>81.1</strong></td>
<td><strong>96.2</strong></td>
<td><strong>81.2</strong></td>
<td><strong>86.0</strong></td>
<td><strong>5.5</strong>↑</td>
</tr>
<tr>
<td><strong>FGSM attack</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERM Baseline</td>
<td>19.4</td>
<td>55.8</td>
<td>51.5</td>
<td>48.7</td>
<td>43.9</td>
<td></td>
</tr>
<tr>
<td>w/ P-Cons. Reg.</td>
<td>32.0</td>
<td>63.8</td>
<td>66.2</td>
<td>66.6</td>
<td>57.2</td>
<td>13.3↑</td>
</tr>
<tr>
<td>w/ OCR</td>
<td><strong>45.5</strong></td>
<td><strong>69.5</strong></td>
<td><strong>73.3</strong></td>
<td><strong>73.2</strong></td>
<td><strong>65.4</strong></td>
<td><strong>21.5</strong>↑</td>
</tr>
<tr>
<td><strong>BIM attack</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERM Baseline</td>
<td>16.6</td>
<td>55.0</td>
<td>40.5</td>
<td>41.6</td>
<td>38.4</td>
<td></td>
</tr>
<tr>
<td>w/ P-Cons. Reg.</td>
<td>26.5</td>
<td>63.3</td>
<td>59.9</td>
<td>60.6</td>
<td>52.6</td>
<td>14.2↑</td>
</tr>
<tr>
<td>w/ OCR</td>
<td><strong>38.7</strong></td>
<td><strong>69.1</strong></td>
<td><strong>68.4</strong></td>
<td><strong>70.4</strong></td>
<td><strong>61.7</strong></td>
<td><strong>23.3</strong>↑</td>
</tr>
<tr>
<td><strong>PGD attack</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERM Baseline</td>
<td>16.8</td>
<td>54.9</td>
<td>40.4</td>
<td>41.5</td>
<td>38.4</td>
<td></td>
</tr>
<tr>
<td>w/ P-Cons. Reg.</td>
<td>26.1</td>
<td>63.4</td>
<td>60.1</td>
<td>60.8</td>
<td>52.6</td>
<td>14.2↑</td>
</tr>
<tr>
<td>w/ OCR</td>
<td><strong>38.5</strong></td>
<td><strong>69.0</strong></td>
<td><strong>68.2</strong></td>
<td><strong>70.6</strong></td>
<td><strong>61.6</strong></td>
<td><strong>23.2</strong>↑</td>
</tr>
</tbody>
</table>

Table 9. Top 1 to top 5 accuracies (%) on Office-Home. Baseline method is SHOT.

<table>
<thead>
<tr>
<th>Acc.(%)</th>
<th>Baseline</th>
<th>Baseline+P-Cons. Reg.</th>
<th>Baseline+OCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top 1</td>
<td>71.8</td>
<td>72.0</td>
<td>72.8</td>
</tr>
<tr>
<td>Top 3</td>
<td>85.5</td>
<td>86.1</td>
<td>87.6</td>
</tr>
<tr>
<td>Top 5</td>
<td>89.4</td>
<td>90.2</td>
<td>92.2</td>
</tr>
</tbody>
</table>

Table 10. Effect of data augmentations.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>All</th>
<th>ColorJitter</th>
<th>RandomGray.</th>
<th>GaussianBlur</th>
</tr>
</thead>
<tbody>
<tr>
<td>PACS (Ours)</td>
<td>85.5</td>
<td>83.4</td>
<td>84.6</td>
<td>83.9</td>
</tr>
<tr>
<td>PACS (P-Cons. Reg.)</td>
<td>83.7</td>
<td>82.0</td>
<td>83.1</td>
<td>82.7</td>
</tr>
<tr>
<td>+PGD (Ours)</td>
<td>61.6</td>
<td>52.9</td>
<td>55.3</td>
<td>47.1</td>
</tr>
<tr>
<td>+PGD (P-Cons. Reg.)</td>
<td>52.6</td>
<td>45.2</td>
<td>47.6</td>
<td>41.5</td>
</tr>
</tbody>
</table>

5. Conclusion and Future work

In this paper, we propose Order-preserving Consistency Regularization (OCR) to enhance model robustness to domain-specific attributes for cross-domain tasks. We first separate the residual component from the augmented representation. Then, we maximize the entropy of the residual component to enlarge the uncertainty of its prediction. As a result, the residual component contains little information about the task of interest, i.e., the model is less sensitive to the domain-specific attributes. Throughout the experiments, we have shown that OCR enhances the generalization of the model and provides better robustness to adversarial attacks. OCR is easy to implement and can be applied to any cross-domain task to improve the performance. Like any data-augmentation based method, our proposal fails when the augmentations are completely independent of the domain gaps. Therefore, exploring the most related data augmentations for specific cross-domain tasks would be a suitable future work.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 62250061, 62176042, 62276054, and in part by the Sichuan Science and Technology Program under Grant 2023YFG0156, and in part by CAAI-Huawei MindSpore Open Fund.

References


Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk,


