AI Open 3 (2022) 200–207

Contents lists available at ScienceDirect

AI Open

journal homepage: www.keaipublishing.com/en/journals/ai-open

Domain generalization by class-aware negative sampling-based contrastive learning☆

Mengwei Xie, Suyun Zhao *, Hong Chen, Cuiping Li

Renmin University of China, No. 59, Zhongguancun Street, haidian District, Beijing, 100872, China

ARTICLE INFO

Keywords:
Transfer learning
Domain generalization
Data corruption

ABSTRACT

When faced with the issue of different feature distribution between training and test data, the test data may differ in style and background from the training data due to the collection sources or privacy protection. That is, the transfer generalization problem. Contrastive learning, which is currently the most successful unsupervised learning method, provides good generalization performance for the various distributions of data and can use labeled data more effectively without overfitting. This study demonstrates how contrast can enhance a model's ability to generalize, how joint contrastive learning and supervised learning can strengthen one another, and how this approach can be broadly used in various disciplines.

1. Introduction

Artificial intelligence requires the ability to transfer knowledge to serve more conveniently and intelligently. Supervised learning techniques depend on a lot of labeled data, but gathering them is expensive. Deep neural networks suffer from covariance shift between training sets and testing sets even if there are enough training data for deep learning. Methods for domain adaptation (Ganin and Lempitsky, 2015; Hoffman et al., 2018; Venkateswara et al., 2017) attempt to solve the problem, however they still require the unlabeled target data to direct the training procedure and require retraining for new application targets. This work focuses on domain generalization, a more difficult issue where the target data is not observed during training. It is a more realistic scenario that one may have no access to target data but needs to train a precise model for data in any style.

With the presumption that each domain has a unique style, we frequently approach the problem of style-variant domain generalization by using training data from a variety of source domains with different styles and target data from a different domain with an unknown style. In the job of object recognition, learning a generalized feature space without respect to styles is the key to solving the style-variant domain generalization problem.

There are some existing methods on domain generalization including methods introducing semantic alignment loss to regularize feature space (Motiian et al., 2017), and methods adopting meta-learning (Balaji et al., 2018; Li et al., 2018c) or adversarial learning (Li et al., 2018a,b) to generalize across domains. But all the above approaches is fulfilled under supervision. Their abilities to explore the various feature is limited despite the delicate design since there are more intrinsic features than what supervised labels lead to capture. Traditional supervised learning learns limited features discriminative only enough to classify but may cause the network to ignore some visual invariances and regularities of images, while unsupervised learning prefers to directly lead to visual information independent from specific domain styles.

Recently, contrastive self-supervised learning has recently garnered a lot of interest after proving to be incredibly effective in machine learning (He et al., 2020; Misra and Maaten, 2020; Zbontar et al., 2021). Without requiring human annotation, a model learns the feature representation of inputs by separating similar from dissimilar data during training on a contrastive pretext task. The learned representation must be transferred to complete the downstream task. When a model simultaneously receives input data representing an elephant and a giraffe, for example, their dissimilarity would prompt the model to determine what shared elements can be used to distinguish between the two. In this way, such methods may be used to explore different feature spaces. Such a learning system might learn to distinguish objects such as animals on the basis of color, or varying length of specific body parts such as neck or nose.

☆ This research is funded by National Natural Science Foundation, China (62276270).

* Corresponding author.
E-mail addresses: vm@ruc.edu.cn (M. Xie), zhaosuyun@ruc.edu.cn (S. Zhao), chong@ruc.edu.cn (H. Chen), licuiping@ruc.edu.cn (C. Li).
URLs: https://github.com/vmmm123 (M. Xie), https://dblp.uni-trier.de/pid/15/5578.html (S. Zhao), http://bi.ruc.edu.cn/chen/index.html (H. Chen), https://dblp.uni-trier.de/pid/03/6827-1.html (C. Li).

https://doi.org/10.1016/j.aiopen.2022.11.004
Received 8 November 2022; Received in revised form 23 November 2022; Accepted 23 November 2022
Available online 5 December 2022

Fig. 1 presents the simplified version of our model, emphasizing on the contrast module. Our method consists of contrast, classification, and adversarial modules. In the contrast module, the feature embeddings of two images transformed from a given input image constitute similar (positive) instances, in contrast to dissimilar negatives from differently-labeled images in various source domains. Notably, we utilize class information for contrastive learning, because the images with identical labels have similar intrinsic features and those with different labels usually have dissimilar features, whereas common contrastive self-supervised learning methods choose dissimilar negatives at random owing to the lack of labels. In strong contrast to such methods, our proposed approach selects the most informative and representative negatives that are valuable for capturing the general domain-invariant features across domains, directly embedding images of distinct classes as negative pairs. In addition, in the adversarial module, the domain discriminator Ganin and Lempitsky (2015), Li et al. (2018b) is introduced to compete with the feature extractor and obtain domain-invariant features.

SupCon (Khosla et al., 2020) also proposed contrastive learning with utilizing labels, but their approach was completely different from our proposed method involving joint contrastive learning and supervised learning. We aim to classify an input image from any domain, which requires our model to classify and generalize, whereas SupCon is designed for learning representation. First, SupCon modified self-supervised contrastive learning, but still performed two-stage training, which implied a requirement of fine-tuning the model with cross-entropy loss of labels for downstream tasks. This may be considered an indirect and inefficient process because label information had been obtained and supervised learning could have been performed in the first stage. Second, SupCon proposed a complicated contrastive loss function to improve the contrastive effect by contrasting an instance with all other instances with the same label, whereas we adopt noise contrastive estimator (NCE) loss with different negative-sampling strategy. Third, we aim to address the domain generalization problem rather than the self-supervised problem, and experimental results demonstrate that our method outperformed SupCon in terms of generalization.

The proposed novel contrastive supervised learning method is referred to as ConSL. The contributions of this study are summarized as follows:

- The present work solves domain generalization by a joint contrastive learning and adversarial learning.
- The model adopts contrastive learning by the negative instances sampled from different classes instead of the random negative sampling strategy.
- The theoretical analysis is provided based on the class-aware negative-sampling contrastive learning

2. Related works

2.1. Contrastive learning

Contrastive learning (He et al., 2020; Misra and Maaten, 2020) has been extensively and effectively used as a pretext task in self-supervised learning. It has proven superior to some other pretext tasks in self-supervised learning, such as training models using rotation degree (Feng et al., 2019; Gidaris et al., 2018) or jigsaw puzzle order (Noroozi and Favaro, 2016) as pseudo labels. Its performance was even comparable to the supervised ResNet-50 in Chen et al. (2020). Contrastive learning constitutes similar and dissimilar instances, and by contrasting these, such models learn corresponding feature representations.

For domain generalization, Carlucci et al. (2019) conducted self-supervised domain generalization by solving the jigsaw puzzle as an auxiliary task; the class label information was not used in their self-supervised module. The separation between the self-supervised module and the classification module may cause some discriminative features to be missed. Because contrastive learning methods outperform methods that sort jigsaw puzzles in self-supervised learning, our proposed approach is able to construct a more general feature space.

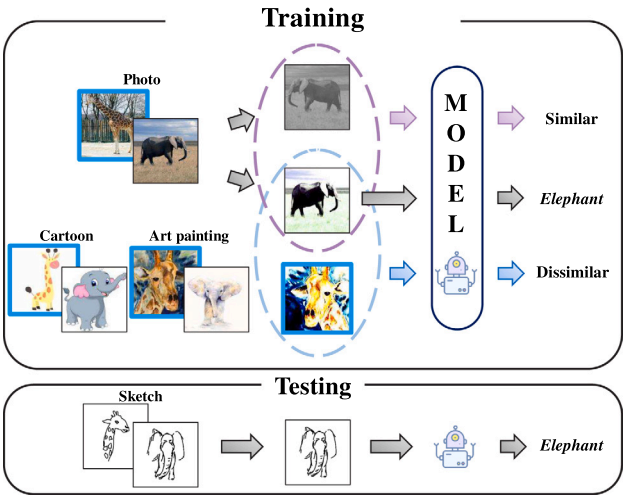

Fig. 1. Simplified version of our proposed method, ConSL. Taking instances from a given input as a similar pair and instances with different labels as a dissimilar pair, the proposed model learns jointly to classify the two images under supervision and to distinguish whether they are similar, improving generalization capabilities of the model for testing on unknown target domains.

2.2. Domain generalization

There are some representative methods presented to solve domain generalization. As Li et al. (2019), we can divide them into the four categories according to their motivations. (1) Methods searching for a domain-invariant feature space to cover new target instances: Li et al. (2018a) introduces maximum mean discrepancy constraint within auto-encoder. Matsuura and Harada (2020) assigns pseudo domain labels and trains feature extractor via adversarial learning. (2) Methods learning a hierarchical set of model parameters to utilize domain-invariant ones: Li et al. (2017) develops a low-rank parameterized model and an aggregation layer is used in D'Innocente and Caputo (2018) to merge generic and specific information. (3) Methods adopting data augmentation to generate new samples to possibly get closer to the target: an adversarial data augmentation method is adopted in Volpi et al. (2018) to synthesize diverse data. (4) Methods using other algorithms: Balaji et al. (2018) focuses on meta learning and episodic training is conducted in Li et al. (2019). Carlucci et al. (2019) takes solving jigsaw puzzle as an auxiliary task while classifying the patch-shuffled images. Different from all of these, our method combines contrastive learning with supervised learning creatively to generalize the model.

3. Problem definition

According to the conventional setting of domain generalization, given S variously distributed source domains $D_S = \{D_i\}_{i=1}^{S}$ covering the same set of k categories, we aim to train a model capable of performing well on the unseen target domain D_t. The ith domain contains N_i labeled instances $\left\{\left(x_j^i, y_j^i, d_j^i\right)\right\}_{j=1}^{N_i}$, where x_j^i indicates the jth image in the ith domain. y_j^i and d_j^i are respectively the one-hot representations of the class and domain labels. The challenge of domain generalization is that data from target domain will not participate in the training process but in the testing process.

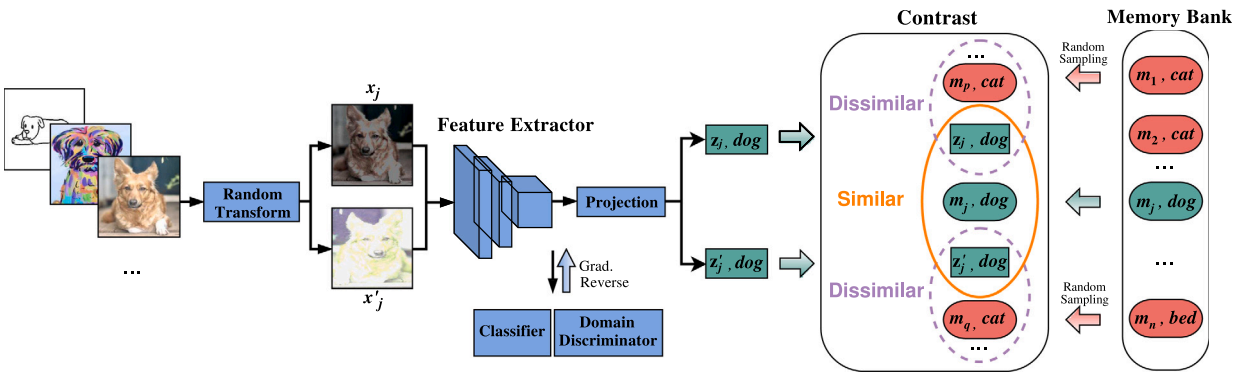

Fig. 2. Illustration of the proposed method ConSL. Each input image is transformed twice at random to produce two diverse images. The feature extractor processes them and feeds the results to a projection network, which maps the features into z_j^i and $z_j^{i\prime}$ (briefly represented by z_j and z_j' in the figure). The embeddings update the memory bank and constitute two similar pairs with the corresponding embedding from the memory bank, which is contrasted with the other instances of different labels randomly sampled from the memory bank.

4. Preliminary

4.1. Adversarial domain generalization

Domain generalization is a object recognition, which is usually achieved by standard cross-entropy loss. Besides, traditional entropy (Grandvalet and Bengio, 2005) $H(p) = -p \cdot log(p)$ as part of the loss for the output of the classifier $C : \mathcal{X} \to \mathbb{R}^k$. The loss is aimed at low-density separation of object classes, namely, making the output of the classifier sparser to enhance the classification accuracy. We use F to represent the feature extractor and present the classification loss function as follows.

$$L_{cls} = \sum_{i=1}^{S} \frac{1}{N_i} \sum_{j}^{N_i} [-y_j^i \cdot \log C(F(x_j^i))] \tag{1}$$

In addition, adversarial learning has been developed from generative adversarial networks (GAN) (Goodfellow et al., 2014) with promising results; moreover, it has been used in domain adaption (Ganin and Lempitsky, 2015) and generalization (Li et al., 2018a). Here, we use a discriminator $D : \mathcal{X} \to \mathbb{R}^S$ with a gradient reverse layer (Ganin and Lempitsky, 2015), which was trained to discriminate the domain labels of inputs, whereas the feature extractor F attempts to render them indistinguishable. Their competition helps extract the domain-invariant features and generalize the model to the target domain in an unknown distribution. The adversarial loss L_{adv} is the standard cross-entropy loss, as given here.

$$L_{adv} = -\sum_{i=1}^{S} \frac{1}{N_i} \sum_{j}^{N_i} d_j^i \cdot \log D(F(x_j^i)) \tag{2}$$

4.2. Contrastive learning

Contrastive Learning Loss

Contrastive learning is conducted by contrasting positives to negatives for obtaining feature representations. In general, a positive pair is constituted by an image z_i and its transform z_i', while a negative pair (z_i, z_j) is two images chosen at random. The similarity of positive pairs should be as high while that of negative pairs should be low. Thus, positive instances are drawn as close together as possible in the feature space, whereas negative instances are mapped to more distant positions. Cosine similarity $s(.)$ is used to measure the similarity of two feature embeddings that apply softmax with temperature parameter τ. A Noise contrastive estimator (NCE) (Gutmann and Hyvärinen, 2010) is introduced in the loss of contrastive self-supervised learning. Each positive pair is compared with negative pairs in noise contrastive estimator (NCE). The aim of contrast learning is to minimize the following loss.

$$L_{NCE}(z_i, z_i') = -log \frac{exp(\frac{s(z_i, z_i')}{\tau})}{exp(\frac{s(z_i, z_i')}{\tau}) + \sum_{i \neq j} exp(\frac{s(z_i, z_j)}{\tau})} \tag{3}$$

Memory Bank

Prior works (Oord et al., 2018; Wu et al., 2018) proved that a large number of negatives in contrastive learning are significant. In a mini-batch SGD optimizer, the number of negatives is limited unless the batch size increases significantly. However, this may lead to disastrous growth in parameter size. To better utilize all negative instances, memory bank M is used to store the feature embedding of every instance in all source domains.

The memory bank does not directly store the representation z_i, except in the first epoch. In accordance with (Misra and Maaten, 2020; Wu et al., 2018), m_i is calculated as an exponential moving average of feature representation z_i in previous epochs, which helps m_i contain more information in previous epochs.

In the pth epoch ($p \neq 1$), the update rule of m_i is:

$$\begin{aligned} \mu^p &= 1 + \lambda \mu^{p-1} \\ \omega_i^p &= z_i + \lambda \omega_i^{p-1} \\ m_i &= \frac{\omega_i}{\mu^p} \end{aligned} \tag{4}$$

Herein, μ^1, λ are hyper-parameters, and $\omega_i^1 = z_i$ in the first epoch.

5. ConSL

5.1. Joint contrastive learning and adversarial learning

The key point in domain generalization is the extraction of the domain-invariant features to train a classifier to classify an image across domains. An overview of the proposed method is presented in Fig. 2.

Adversarial learning has been adopted in domain adaptation to explore the domain-irrelevant features. Similarly, we utilize it for the same purpose. The first objective of our approach is to maximize the classification accuracy and explore domain-invariant feature by minimizing $L_{adv} + L_{cls}$. Noting that there are two feature embeddings transformed randomly from the same original image, we choose only one for classification and adversarial learning to simplify the model.

What is more, to explore the invariances and regularities of the image, we conducted contrastive learning to assist the feature extractor. First, for every image, the transformer produces two diverse images by

data augmentation and conveys them to the domain-invariant feature extractor F, because later contrasting twice with two diverse images of the same input simultaneously can help the current network capture features with more discrimination, as shown in the ablation study section. Then, the projection network reduces the dimensions of the extracted features and outputs z_j^i and $z_j^{i\prime}$.

To better use information in the previous epochs and expand the comparison range, the model uses a memory bank to store the embedding of every image and update the memory bank as the training continues. These two embeddings from the same input together with the corresponding embedding taken from the memory bank constitute two positive pairs. With m_j^i taken from the memory bank, both constitute two positive pairs of instances (z_j^i, m_j^i) and $(z_j^{i\prime}, m_j^i)$. Each positive pair corresponds to many negative pairs for the next contrastive part. The representations in the memory bank are updated as Eq. (4) while $\mu^1 = 0$, $\lambda = 0.5$, and $\omega_j^{i,1} = z_j^i$ in the first epoch. In addition, to better use class label information and later realize a more distinct contrast, the memory bank is upgraded to store the class label of every image.

5.2. Class-aware negative-sampling contrastive learning

In general contrastive learning, because no class information is available, a negative pair is generally composed of two random images, which may result in similar negatives, such as images from the same class. In supervised learning, from the perspective of cross-entropy loss (1), the classifier C tries to distinguish the labels of the inputs $F(x_j^i)$, whereas F tends to extract the most easily identifiable features. Thus, there is a trend that the outputs $F(x)$ with the same label could be similar, which is specifically one of the reasons for the development of supervised learning techniques. However, conventional contrastive self-supervised learning chooses negatives at random and aims to make their representations dissimilar, which may take those with the same label as hard but false negatives, resulting in conflicts with supervised learning.

The fact that images of the same class have higher similarity than those of different classes inspired us to utilize class information in contrastive learning to enhance the distinctness of contrast between classes. To better introduce contrastive learning to supervised learning, we choose negatives from different classes in the memory bank. Then, we use cosine similarity s to measure the similarity of two feature embeddings that apply softmax with temperature parameter τ. Following Gutmann and Hyvärinen (2010), we contrast each positive pair with negative pairs in our new NCE and aim to minimize the following loss.

$$L_{NCE}(z_j^i, m_j^i) = -log \frac{exp(\frac{s(z_j^i, m_j^i)}{\tau})}{exp(\frac{s(z_j^i, m_j^i)}{\tau}) + \sum_{m^* \in M_{i,j}} exp(\frac{s(z_j^i, m^*)}{\tau})} \tag{5}$$

M_j^i is a set of negative instances m^* sampled randomly from items satisfying $y \neq y_j^i$ in memory bank M. Because there are two positive pairs for each input image, we implement the constructed loss function L_{sup} using a combination of two NCE loss functions. The following supervised contrastive loss L_{sup} encourages two feature embeddings of transformed images from the same embedding to be similar, and the feature embedding of images from different classes to be dissimilar.

$$L_{sup} = \sum_{i=1}^{S} \frac{1}{N_i} \sum_{j}^{N_i} [L_{NCE}(z_j^i, m_j^i) + L_{NCE}(z_j^{i\prime}, m_j^i)] \tag{6}$$

Traditional contrastive learning takes two steps to minimize the loss $L_{NCE}(x_j, x_k)$, where $j \neq k$, and then fine-tuning the model by minimizing cross-entropy loss. Our method ConSL adopted the end2end approach, and the total training objective is as follows, where α and β are hyper-parameters.

$$\min_{F,C} \max_{D} L_{cls} + \alpha L_{sup} - \beta L_{adv} \tag{7}$$

5.3. Theoretical analysis

Inspired by Khosla et al. (2020), which shows hard positives contributes more, we prove why excluding false negative is necessary for generalization.

The following expression for the gradient of $\mathbf{L_{NCE}}$ on the basis of Eq. (5) is

$$\begin{aligned}
\frac{\partial \mathbf{L_{NCE}}(z_j^i, m_j^i)}{\partial z_j^i} &= -\frac{\partial}{\partial z_j^i} log \left[\frac{exp(\frac{s(z_j^i, m_j^i)}{\tau})}{exp(\frac{s(z_j^i, m_j^i)}{\tau}) + \sum_{m^* \in M_{i,j}} exp(\frac{s(z_j^i, m^*)}{\tau})} \right] \\
&= -\frac{1}{\tau} \left[\frac{\partial s(z_j^i, m_j^i)}{\partial z_j^i} - \frac{exp(\frac{s(z_j^i, m_j^i)}{\tau}) \frac{\partial s(z_j^i, m_j^i)}{\partial z_j^i} + \sum_{m^* \in M_{i,j}} exp(\frac{s(z_j^i, m^*)}{\tau}) \frac{\partial s(z_j^i, m^*)}{\partial z_j^i}}{exp(\frac{s(z_j^i, m_j^i)}{\tau}) + \sum_{m^* \in M_{i,j}} exp(\frac{s(z_j^i, m^*)}{\tau})} \right] \\
&= -\frac{1}{\tau} \left[(1 - \mathbf{Q}(z_j^i, m_j^i)) \frac{\partial s(z_j^i, m_i^j)}{\partial z_j^i} - \sum_{m^* \in M_{i,j}} \mathbf{Q}(z_j^i, m^*) \frac{\partial s(z_j^i, m^*)}{\partial z_j^i} \right] \\
&= -\frac{1}{\tau \|z_j^i\|} (I - \frac{z_j^i (z_j^i)^T}{\|z_j^i\|^2}) \left[(1 - \mathbf{Q}(z_j^i, m_j^i)) \frac{m_j^i}{\|m_j^i\|} - \sum_{m^* \in M_{i,j}} \mathbf{Q}(z_j^i, m^*) \frac{m^*}{\|m^*\|^2} \right] \\
&= \frac{\mathbf{Q}(z_j^i, m_j^i) - 1}{\tau \|z_j^i\| . \|m_j^i\|} (m_j^i - \frac{z_j^i (z_j^i)^T m_j^i}{\|z_j^i\|^2}) + \sum_{m^* \in M_{i,j}} \frac{\mathbf{Q}(z_j^i, m^*)}{\tau \|z_j^i\| . \|m^*\|^2} (m^* - \frac{z_j^i (z_j^i)^T m^*}{\|z_j^i\|^2})
\end{aligned} \tag{8}$$

where $\mathbf{Q}(z_j^i, m) = \frac{exp(\frac{s(z_j^i, m)}{\tau})}{exp(\frac{s(z_j^i, m_j^i)}{\tau}) + \sum_{m^* \in M_{i,j}} exp(\frac{s(z_j^i, m^*)}{\tau})}$.

The gradient concerned about the positive pairs and negative pairs can be separated:

$$\begin{aligned}
&\frac{\partial \mathbf{L_{NCE}}(z_j^i, m_j^i)}{\partial z_j^i} = \mathbf{G_{pos}} + \sum_{m^* \in M_{i,j}} \mathbf{G_{neg}} \\
s.t.\ \ &\mathbf{G_{pos}} = \frac{\mathbf{Q}(z_j^i, m_j^i) - 1}{\tau \|z_j^i\| . \|m_j^i\|} (m_j^i - \frac{z_j^i (z_j^i)^T m_j^i}{\|z_j^i\|^2}) \\
&\mathbf{G_{neg}} = \frac{\mathbf{Q}(z_j^i, m^*)}{\tau \|z_j^i\| . \|m^*\|^2} (m^* - \frac{z_j^i (z_j^i)^T m^*}{\|z_j^i\|^2})
\end{aligned} \tag{9}$$

Here, $\mathbf{s}$ is cosine similarity function. According to Lemma 1 $AA^TB = (A.B)A$, we can conclude:

$$\begin{aligned}
&(m - \frac{(z_j^i . m) z_j^i}{\|z_j^i\|^2})^T (m - \frac{(z_j^i . m) z_j^i}{\|z_j^i\|^2}) \\
&= m^T m - \frac{(z_j^i . m) m^T z_i^j}{\|z_j^i\|^2} - \frac{(z_j^i . m)(z_j^i)^T m}{\|z_j^i\|^2} + \frac{(z_j^i . m)^2}{\|z_j^i\|^2} \\
&= \|m\|^2 \left[\frac{m^T m}{\|m\|^2} - \frac{(z_j^i . m)^2}{\|z_j^i\|^2 \|m\|^2} \right] \\
&= \|m\|^2 \left[1 - \mathbf{s}^2(z_j^i, m) \right]
\end{aligned} \tag{10}$$

Then $\|m - \frac{(z_j^i . m) z_j^i}{\|z_j^i\|^2}\| = \|m\| \sqrt{1 - \mathbf{s}^2(z_j^i, m)}$.

So the length of $\mathbf{G_{pos}}$ and $\mathbf{G_{neg}}$ is:

$$\begin{aligned}
\|\mathbf{G_{pos}}\| &= \frac{1 - \mathbf{Q}(z_j^i, m_j^i)}{\tau \|z_j^i\| . \|m_j^i\|} \|m_j^i - \frac{z_j^i (z_j^i)^T m_j^i}{\|z_j^i\|^2}\| \\
&= \frac{1 - \mathbf{Q}(z_j^i, m_j^i)}{\tau \|z_j^i\| . \|m_j^i\|} \|m_j^i - \frac{(z_j^i . m_j^i) z_j^i}{\|z_j^i\|^2}\| \\
&= \frac{1 - \mathbf{Q}(z_j^i, m_j^i)}{\tau \|z_j^i\|} \sqrt{1 - \mathbf{s}^2(z_j^i, m_j^i)}
\end{aligned} \tag{11}$$

$$\begin{aligned}\|\mathbf{G_{neg}}\| &= \frac{\mathbf{Q}(z_j^i, m^*)}{\tau\|z_j^i\|.\|m^*\|^2}\|m^* - \frac{z_j^i(z_j^i)^T m^*}{\|z_j^i\|^2}\| \\ &= \frac{\mathbf{Q}(z_j^i, m^*)}{\tau\|z_j^i\|.\|m^*\|^2}\|m^* - \frac{(z_j^i.m^*)z_j^i}{\|z_j^i\|^2}\| \\ &= \frac{\mathbf{Q}(z_j^i, m^*)}{\tau\|z_j^i\|}\sqrt{1-\mathbf{s}^2(z_j^i, m^*)}\end{aligned} \tag{12}$$

Foregoing shows that the similarity of positive or negative pairs does affect the gradient. As $\mathbf{s}(z_j^i, m_j^i) \to 0$, $\mathbf{s}(z_j^i, m^*) \to 0$ results in the larger gradient for $\|\mathbf{G_{neg}}\|$ and $\|\mathbf{G_{pos}}\|$ respectively, the network would focus more on hard (similar) negatives and hard (dissimilar) positives, whose cosine similarity is closer to 0 than easy ones.

Suppose instances with the same label as negatives, they could be hard negatives since they are more similar and $\mathbf{s}(z_j^i, m^*) \to 0$. According to Eq. (12), the network concentrates more on them and tries harder to push them far away, whereas these instances should get together and be classified as the same label. So it is significant that our model choose negatives under supervision to avoid false hard negative, while only instances with different label can be chosen as negatives.

In addition, the network draws positives (z_j^i, m_j^i) as close together and maps negatives (z_j^i, m^*) to more distant positions, helping enrich and perfect the feature representations z_j^i. Due to the fact that $z_j^i = F(x_j^i)$, the better representation can assist the cross-entropy loss Eq. (1) fall down and the network F converge and generalize.

Lemma 1. *Given* $A = [a_0, a_1, \ldots, a_n]^T$, $B = [b_0, b_1, \ldots, b_n]^T$, $a_i \epsilon \mathbb{R}$, $b_i \epsilon \mathbb{R}$ *for* $\forall i \epsilon [1, n]$, *then* $AA^T B = (A.B)A$.

Proof. Since $(AA^T)_{i,j} = a_i a_j$, then $[(AA^T)B]_i = \sum_{k=0}^n a_i a_k b_k$. So $[(A.B)A]_i = (\sum_{k=0}^n a_k b_k)a_i = [(AA^T)B]_i$. $AA^T B = (A.B)A$ is proved. □

6. Experiments

6.1. Baselines and implementation details

We considered two common multiple-domain generalization benchmark datasets, in which each domain has a specific style. **PACS** (Li et al., 2017) covers 4 domains, including Photo, Art Paintings, Cartoon and Sketch, aggregating 7 object categories. **Office–Home** (Venkateswara et al., 2017) was a larger dataset, covering 4 domains, including Art, Clipart, Product and Real-World, aggregating 65 categories. Specifically, the images in the Product domain were from vendor websites and incorporated a white background, whereas the Real-World domain contained object images collected with a commonly available camera. Following the experiment in Li et al. (2017), we used three domains for training and the remaining one, which did not participate in the training, for testing. Using the experimental protocol of D'Innocente and Caputo (2018), we split the data of our source domains into two groups, using 90% as training and 10% as validation data by random selection from the overall dataset.

We compare our method **ConSL** with five state-of-the-art methods for multiple-domain generalization. By design, **Epi-FCR** (Li et al., 2019) broke networks up into feature extractor and classifier modules and episodically trains them. **MetaReg** (Balaji et al., 2018) adopted meta-learning to learn a regularizer to generalize a final classifier. **D-SAM** (D'Innocente and Caputo, 2018) used an aggregation layer strategy to merge generic and specific information. **JiGen** (Carlucci et al., 2019) trained two classifiers for predicting the label and the order of shuffled image patches. **Dgmmld** (Matsuura and Harada, 2020) assigned a pseudo domain label for each image and trains a feature extractor by adversarial learning. **Deep All** is the vanilla aggregation method that fine-tunes AlexNet or ResNet with all the source data.

For PACS, we directly obtain scores from the original paper of each method. Dgmmld provides results that vary the number of pseudo labels. We select those based on the highest average accuracy. For OfficeHome, we rerun Epi-FCR, MetaReg, and Dgmmld with their official codes.

Code is available at https://anonymous.4open.science/r/ConSL-7F59/README.md. We use AlexNet and ResNet-18 pre-trained on ImageNet with the last layer removed as the basic architecture. For AlexNet, we consider its feature as a feature extractor; its classifier adds one fully connected layer as a classifier in the proposed model, the number of whose output was the same as the object category number. We modify ResNet-18 as our feature extractor and initialized a fully connected layer for classification. The domain discriminator consists of 3 fully connected layers ($1024 \to 1024 \to$ number of domains S). The architecture is the same as that used in Ganin et al. (2016) for a fair comparison, and each of the 1024-unit layers used a ReLU activation function. Between the domain discriminator and the feature extractor, we insert a gradient reverse layer (Ganin and Lempitsky, 2015) to achieve adversarial learning, where the adversarial weight was 1. The projection network adopts a fully connected layer for feature dimension reduction, along with an L2 normalization technique.

We initialize the memory bank with features output from the model. The negative set M_j^i for each instance x_j^i contains $\epsilon \times N$ negatives, where ϵ is a hyper-parameter controlling the number of negatives, and N is the number of instances whose labels satisfy $y \neq y_j^i$ in the entire memory bank M.

First, we test different values for the trade-off parameters α and β between losses on PACS and determine $\alpha = 0.5, \beta = 0.5$, which is used for all other tasks in our method. In addition, $\epsilon = 0.1$, and the dimension of the feature embedding used for contrast is projected to 256. $\tau = 0.07$, a value other contrastive learning methods (Chen et al., 2020; Misra and Maaten, 2020) have commonly adopted. We train the proposed model for 30 epochs with the SGD solver having a momentum of 0.9, a weight decay of 5e-4, and a batch size of 96 (32 per source domain). We randomly choose samples to fill the batch because the numbers varied between the domains. The initial learning rate was 1e-3 and declined by a factor of 0.2, every 10 epochs. However, the learning rate of the domain discriminator, projection network, and additional layers in the classifier are 10 times larger because they are trained from scratch.

6.2. Results

Table 1 shows the experimental results using AlexNet on PACS datasets. Our method ConSL is superior to all the other methods in three out of four target cases and the average case, especially for the target Art-painting and Sketch with over 4% and 7% accuracy higher than the second best method. Even Deep All in our setting shows great gain with regard to most methods, indicating that generalization benefits from data augmentation. Tables 2 and 3 show results using ResNet-18 on PACS and Office–Home datasets. ConSL has the best accuracy in most cases, including the average case. We note that some elaborate methods show a limited improvement from their corresponding Deep All baselines. In some cases, they fare worse than the baselines, such as Epi-FCR and Dgmmld on Office–Home in Table 3. The basic network ResNet-18 is sufficiently strong and Office–Home is a larger dataset with 65 categories compared to the PACS, which is beneficial for generalization. However, ConSL performs better than Deep All on such strong baselines, implying that besides the effectiveness of data augmentation, our model is effective on domain generalization.

6.3. Ablation study

In this section, we present the result of an ablation study performed on the PACS datasets using AlexNet to explore the effect of each component in our proposed method. A description of every item in the first column in Table 4 is given as follows. ConSL w/o con.: The model that removes contrast components. ConSL con. once: This model contrasts only once for each input image, instead of constituting two

Table 1
Results on the PACS dataset using AlexNet. The title of each column indicates the name of the target domain, while the remaining three domains were used as the source for training. Deep All correspond to each method, disabling all the introduced domain generalization conditions. The best results of the generalization methods are highlighted in bold, whereas the result produced by Deep All are underlined when they are better than all others. The results of our approach are the average over five repetitions.

PACS	Art.	Cartoon	Photo	Sketch	Avg.
AlexNet					
Deep All	63.4	66.1	88.5	56.6	68.7
Epi-FCR	64.7	72.3	86.1	65.0	72.0
Deep All	67.21	66.12	88.47	55.32	69.28
MetaReg	69.82	70.35	**91.07**	59.26	72.62
Deep All	64.91	64.28	86.67	53.08	67.24
D-SAM	63.87	70.70	85.55	64.66	71.20
Deep All	66.68	69.41	89.98	60.02	71.52
JiGen	67.63	71.71	89.00	65.18	73.38
Deep All	68.09	70.23	88.86	61.80	72.25
Dgmmld	69.27	72.83	88.98	66.44	74.38
Deep All	69.92	70.08	88.86	69.43	74.57
ConSL	**72.73**	**72.91**	89.35	**73.71**	**77.18**

Table 2
Results on the PACS dataset using ResNet-18. For details, see Table 1.

PACS	Art.	Cartoon	Photo	Sketch	Avg.
ResNet-18					
Deep All	77.6	73.9	94.4	70.3	79.1
Epi-FCR	82.1	77.0	93.9	73.0	81.5
Deep All	79.9	75.1	95.2	69.5	79.9
MetaReg	**83.7**	77.2	95.5	70.3	81.7
Deep All	77.84	75.89	95.19	69.27	79.55
D-SAM	77.33	72.43	95.30	77.83	80.72
Deep All	77.85	74.86	95.73	67.74	79.05
JiGen	79.42	75.25	96.03	71.35	80.51
Deep All	78.34	75.02	96.21	65.24	78.70
Dgmmld	81.28	77.16	96.09	72.29	81.83
Deep All	79.71	74.50	96.33	72.25	80.70
ConSL	82.80	**77.83**	**96.15**	**80.31**	**84.27**

Table 3
Results on the OfficeHome dataset using ResNet-18. For details, see Table 1.

Office–Home	Art	Clipart	Product	Real.	Avg.
ResNet-18					
Deep All	56.32	49.42	73.19	74.52	63.36
Epi-FCR	52.53	49.78	69.02	72.71	61.01
Deep All	56.67	43.80	70.36	72.20	60.76
MetaReg	55.53	44.92	71.47	72.01	60.98
Deep All	55.59	42.42	70.34	70.86	59.81
D-SAM	58.03	44.37	69.22	71.45	60.77
Deep All	52.15	45.86	70.86	73.15	60.51
JiGen	53.04	47.51	71.47	72.79	61.20
Deep All	58.78	48.77	73.37	76.84	64.44
Dgmmld	58.48	49.75	73.29	**75.40**	64.23
Deep All	56.65	50.72	73.06	74.78	63.80
ConSL	**58.70**	**52.37**	**73.64**	74.48	**64.80**

Table 4
Results of further study on the PACS dataset using AlexNet. For details, see Table 1.

PACS	Art.	Cartoon	Photo	Sketch	Avg.
AlexNet					
Deep All	69.92	70.08	88.86	69.43	74.57
ConSL w/o con.	68.39	68.03	88.76	68.35	73.39
ConSL con. once	71.60	71.59	89.50	**74.14**	76.40
ConSL w/o sup.	71.06	71.70	89.48	73.18	76.36
ConSL w/o adv.	70.70	70.90	88.12	72.77	75.62
ConSL w/o ent.	72.12	72.78	90.24	70.78	76.48
ConSL	**73.00**	**72.91**	**89.51**	73.71	**77.28**

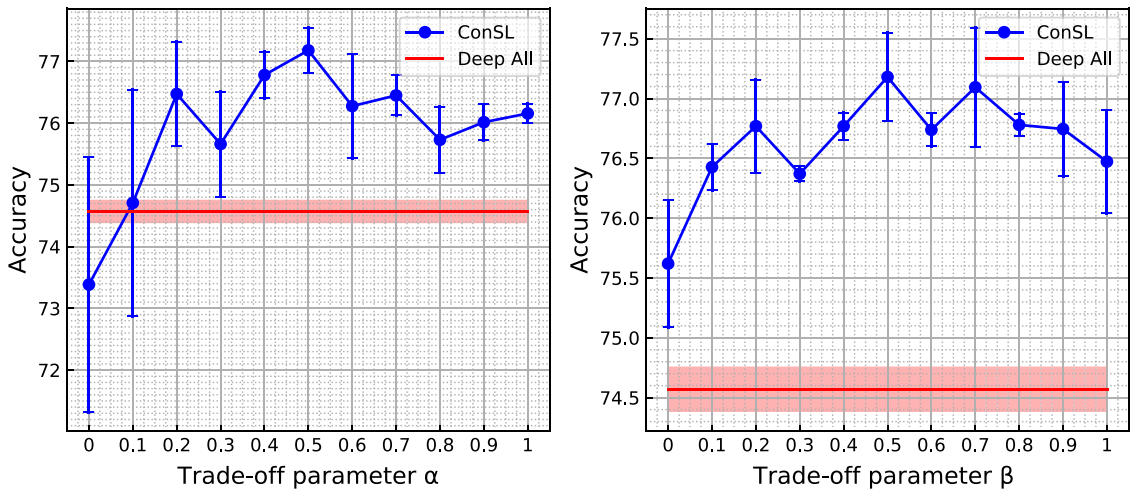

Fig. 3. The mean and standard deviation results of JCS and Deep All with regard to various trade-off parameters. The reported result is the global average over all the target domain cases. The red line represents our Deep All from Table 1.

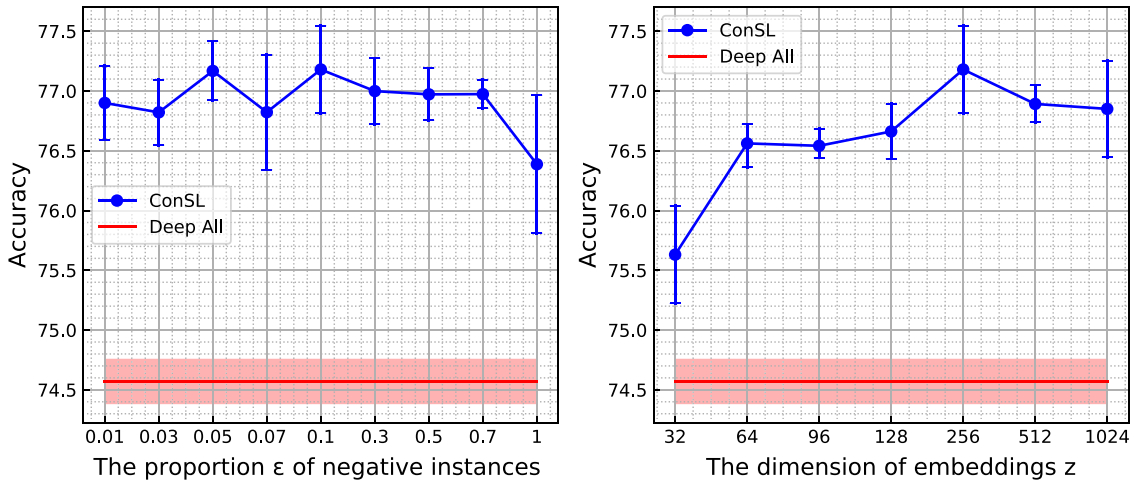

Fig. 4. The mean and standard deviation results of JCS and Deep All with regard to various negative instance size and feature embedding dimension. For details, see Fig. 3.

positive pairs for the same input. ConSL w/o sup.: This model contrasts instances without supervised information, that is, by choosing negatives entirely randomly. ConSL w/o adv.: A model that trains without a domain discriminator to perform adversarial learning. ConSL w/o ent: A model trained without entropy loss. ConSL: Our full proposed model.

We note the following observations from Table 4. (1) Accuracy of ConSL w/o con. indicates that contrastive loss plays an important role in our method, and the remaining adversarial module achieved lower accuracy than Deep All. Because Deep All was fine-tuned on the pretrained AlexNet, the extra adversarial module without combining with contrastive learning in ConSL w/o con. could reduce the generalization abilities of the pre-trained network. (2) Results of the ConSL con. once were closer but lower than the full model, which is reasonable because contrasting once is effective in extracting the feature representations, but contrasting twice can help the current network capture more representative features within two positive pairs. (3) ConSL w/o sup. demonstrated that contrastive learning benefits from supervised information, and our supervised contrastive condition improves performance. (4) Close accuracy between ConSL w/o ent. and the full model ConSL shows that the entropy loss does not play an important role in our method. (5) Finally, the results in ConSL w/o adv. are worse than those of the full model, showing that adversarial learning helps to extract domain-discriminative features for generalization.

7. Conclusions

In this study, we have found and proved that contrastive learning benefits generalization. The key point of our method ConSL is to introduce contrastive learning for mining domain-invariant features focused on the object of an image rather than its specific appearance, using class supervised information to realize a more distinct contrast. There are many ways to enhance self-supervised learning, and it is natural to design a method that is an extension of contrastive learning, which suggests methods such as picking hard positives or negatives and replacing the memory bank, as in He et al. (2020), to make our methods suitable for large datasets. Because contrastive learning has been effectively and widely applied across computer vision, natural language processing, and speech recognition fields, among others, our method provides the possibility of generalizing in cases of domain shift, which is expected to be of benefit for future works in a wide variety of fields.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

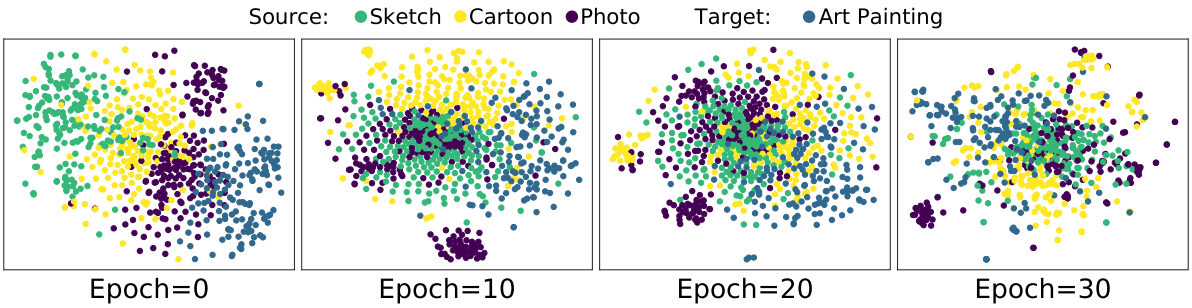

Fig. 5. Visualization of generalized features. Dots in Different color represent features of different domains. Source indicates the source datasets used in training while target dataset is only used for testing.

Appendix A. Hyper-parameter tuning

From Fig. 3, we find out that the trade-off parameter α is important and after it reaches 0.2, our method JCS is always better than Deep All, reach the peak at $\alpha = 0.5$. As for β, the accuracy is more stable and stays between 76 and 77.5, which means that the parameter is easy to be chosen.

Appendix B. Varying the proportion of negatives and the embedding dimension

We investigate the sensitivity of the size of negative instances. Usually, the previous methods (Misra and Maaten, 2020; Wu et al., 2018) take the number of negative instances as hyper-parameter, so they have to adjust it for every dataset. To improve the generalization and application convenience of our model, we use the ϵ to control negative sample size, which indicates that the selected negatives for each instance occupy ϵ proportion of the total different-labeled instances in train set. As presented in Fig. 4 we can see that our method outperforms Deep All throughout varying the proportion of negative instances. And as the proportion moderately increases, the corresponding accuracy reaches the best at $\epsilon = 0.1$. It shows that more negatives do not always bring the accuracy promotion. When taken all qualified instances as negatives ($\epsilon = 1$), the model performance is even worst and most unstable.

Since the contrastive learning plays a significant role in our method, we then explore the dimension number of embedding used for contrast. In Fig. 4, we observe only an overall variation of 1.7 while it is still higher than Deep All. The accuracy almost maintains at the same level excluding 32 dimension and obtains a bit higher accuracy when the dimension is set to 256, implying that our model is not sensitive to embedding dimension. So it can be considered to project the feature to lower-dimension space in demand to train on larger dataset for alleviating the time and computation spaces.

Appendix C. Visualization

We visualize the distributions of the features extracted by the domain-invariant feature extractor and present the results in Fig. 5, using t-SNE (Chan et al., 2019) to project them to two-dimension space. Before training when epoch = 0, the features from different domains have distinctly different distributions. As the training continues, their distributions become out of order and the distribution boundary fades away as dots in different colors start to coincide. In particular, the blue dots represent the target domain which does not participate in training, gradually integrating into the group of other-color dots representing the source domains along with the training. It proves that the features from source domains cover the space spanned by any kind of target.

References

Balaji, Y., Sankaranarayanan, S., Chellappa, R., 2018. MetaReg: Towards domain generalization using meta-regularization. In: NeurIPS.

Carlucci, F.M., D'Innocente, A., Bucci, S., Caputo, B., Tommasi, T., 2019. Domain generalization by solving jigsaw puzzles. CVPR.

Chan, D.M., Rao, R., Huang, F., Canny, J.F., 2019. GPU accelerated t-distributed stochastic neighbor embedding. J. Parallel Distrib. Comput. 131, 1–13.

Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E., 2020. A simple framework for contrastive learning of visual representations. arXiv arXiv:2002.05709.

D'Innocente, A., Caputo, B., 2018. Domain generalization with domain-specific aggregation modules. In: Pattern Recognition - 40th German Conference, GCPR vol. 11269. Springer, pp. 187–198.

Feng, Z., Xu, C., Tao, D., 2019. Self-supervised representation learning by rotation feature decoupling. CVPR.

Ganin, Y., Lempitsky, V., 2015. Unsupervised domain adaptation by backpropagation. In: ICML.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V., 2016. Domain-adversarial training of neural networks. J. Mach. Learn. Res. 17, 59:1–59:35.

Gidaris, S., Singh, P., Komodakis, N., 2018. Unsupervised representation learning by predicting image rotations. In: ICLR.

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

Grandvalet, Y., Bengio, Y., 2005. Semi-supervised learning by entropy minimization. In: CAP.

Gutmann, M., Hyvärinen, A., 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: AISTATS.

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.B., 2020. Momentum contrast for unsupervised visual representation learning. CVPR.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., Darrell, T., 2018. Cycada: Cycle-consistent adversarial domain adaptation. In: ICML.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D., 2020. Supervised contrastive learning. NeurIPS.

Li, H., Pan, S., Wang, S., Kot, A., 2018a. Domain generalization with adversarial feature learning. CVPR.

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

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

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

Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.-Z., Hospedales, T.M., 2019. Episodic training for domain generalization. ICCV.

Matsuura, T., Harada, T., 2020. Domain generalization using a mixture of multiple latent domains. In: AAAI.

Misra, I., Maaten, L.V.D., 2020. Self-supervised learning of pretext-invariant representations. CVPR.

Motiian, S., Piccirilli, M., Adjeroh, D., Doretto, G., 2017. Unified deep supervised domain adaptation and generalization. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 5716–5726.

Noroozi, M., Favaro, P., 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV.

Oord, A., Li, Y., Vinyals, O., 2018. Representation learning with contrastive predictive coding. arXiv arXiv:1807.03748.

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

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

Wu, Z., Xiong, Y., Yu, S., Lin, D., 2018. Unsupervised feature learning via non-parametric instance discrimination. CVPR.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S., 2021. Barlow twins: Self-supervised learning via redundancy reduction. In: ICML.