Promoting Semantic Connectivity: Dual Nearest Neighbors Contrastive Learning for Unsupervised Domain Generalization

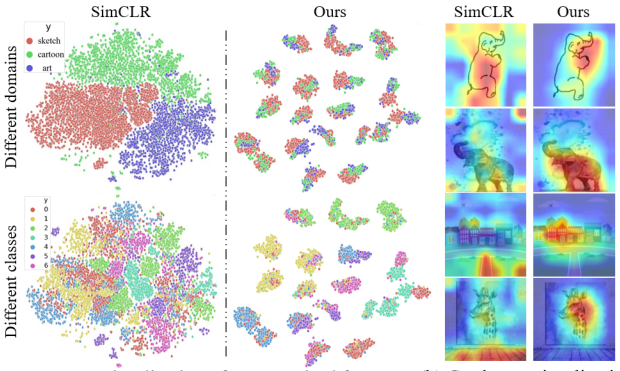
Yuchen Liu[1†], Yaoming Wang[1†], Yabo Chen[2], Wenrui Dai[2*], Chenglin Li[1], Junni Zou[2], and Hongkai Xiong[1]
[1]Department of Electronic Engineering, Shanghai Jiao Tong University, China
[2]Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
{liuyuchen6666, wang_yaoming, chenyabo, daiwenrui, lcl1985, zoujunni, xionghongkai}@sjtu.edu.cn

Abstract

Domain Generalization (DG) has achieved great success in generalizing knowledge from source domains to unseen target domains. However, current DG methods rely heavily on labeled source data, which are usually costly and unavailable. Since unlabeled data are far more accessible, we study a more practical unsupervised domain generalization (UDG) problem. Learning invariant visual representation from different views, i.e., contrastive learning, promises well semantic features for in-domain unsupervised learning. However, it fails in cross-domain scenarios. In this paper, we first delve into the failure of vanilla contrastive learning and point out that semantic connectivity is the key to UDG. Specifically, suppressing the intra-domain connectivity and encouraging the intra-class connectivity help to learn the domain-invariant semantic information. Then, we propose a novel unsupervised domain generalization approach, namely Dual Nearest Neighbors contrastive learning with strong Augmentation (DN^{2}A). Our DN^{2}A leverages strong augmentations to suppress the intra-domain connectivity and proposes a novel dual nearest neighbors search strategy to find trustworthy cross domain neighbors along with in-domain neighbors to encourage the intra-class connectivity. Experimental results demonstrate that our DN^{2}A outperforms the state-of-the-art by a large margin, e.g., 12.01% and 13.11% accuracy gain with only 1% labels for linear evaluation on PACS and DomainNet, respectively.

1. Introduction

Deep learning methods have yielded prolific results in various tasks in recent years. However, they are tailored for experimental cases, where training and testing data share the same distribution. When transferred to practical applications, these methods perform poorly on out-of-distribution data due to domain shifts [27, 30]. To tackle this issue, domain generalization (DG) methods [25, 35] are proposed to learn transferable knowledge from multiple source domains to generalize on unseen target domains. Despite the promising results of DG, they are restricted to supervised training with large amounts of labeled source data. However, large-scale labeled source data are often unavailable due to the laborious and expensive annotation capture, while unlabeled data are far more accessible. Thus, we study the more practical unsupervised domain generalization (UDG) [34] problem to learn domain-invariant features in an unsupervised fashion.



(a) T-sne visualization of unsupervised features (b) Grad-cam visualization

Figure 1. (a) t-SNE visualization of unsupervised features learned by SimCLR and our DN^{2}A on PACS. (b) Grad-cam visualization of linear probing for SimCLR and ours with 10% labeled data.

Recent advances in unsupervised learning prefer contrastive learning (CL) [2, 14, 28, 32], which learns semantic representation by enforcing similarity over different augmentations of the same image. However, most CL methods are designed for i.i.d. datasets and can hardly accommodate the cross-domain scenario in UDG. As depicted in Fig. 1, vanilla SimCLR fails to learn domain-invariant semantic features but learns domain-biased features. For further understanding, we dive into this phenomenon and propose the semantic connectivity for UDG to measure the intra-domain and intra-class similarity. From *augmentation graph* view [12, 31], semantic connectivity is the support overlap of augmented samples within the same semantic class. We further find that the degraded semantic connectivity is responsible for

*Corresponding author: Wenrui Dai. †Equal contribution.

the failure of vanilla CL in UDG, which is reflected in two folds, i.e., large intra-domain connectivity and small intra-class connectivity. Positive samples generated by standard augmentations under the i.i.d. hypothesis share too much domain-relevant information, which induces the model to learn domain-related features for alignment, resulting in large intra-domain connectivity. Moreover, it is hard to capture domain invariance via handcrafted transformations due to significant distribution shifts across domains. For example, one can hardly transform a cat from sketch to photo. Small intra-class connectivity occurs in cross-domain scenarios.

To address these issues, we propose to suppress the intra-domain connectivity and enhance intra-class connectivity. First, we leverage strong augmentations to generate positive samples with a small amount of shared information, where the domain nuisance information is suppressed. The suppressed domain-related information decreases intra-domain connectivity, and the learned unsupervised representation can achieve a higher degree of invariance against domain shifts. Besides, we employ cross domain nearest neighbors (NN) as positive samples to impose the domain invariance by enforcing the similarity between cross domain samples potentially belonging to the same category, which can increase the cross-domain intra-class connectivity. In addition, we improve cross domain NN by a dual NN strategy that further introduces in-domain NN as positives to overcome the intra-domain variances and increase the intra-domain intra-class connectivity. For cross-domain NNs, a direct search may result in many false matches, due to distribution shifts across domains. Since searching NN within a domain without distribution shift is more accurate than across domains, we propose a novel Cross Domain Double-lock NN (CD^2NN) search strategy that employs more accurate in-domain NN as a mediator to find more trustworthy cross domain neighbors for boosting the performance. For in-domain NN, since direct searching may fail to find sufficiently diverse samples to overcome intra-domain variances, we resort to more distinct cross domain NN as a mediator to find more diverse neighbors, namely In-domain Cycle NN (ICNN). Totally, our dual nearest neighbors, i.e., CD^2NN and ICNN, can increase the intra-class connectivity for UDG. In a nutshell, contributions of this paper are summarized as:

- We propose a novel semantic connectivity metric to indicate the inherent problem of contrastive learning in UDG, and propose a novel method DN^2A to increase the semantic connectivity with theoretical guarantees.
- We propose to leverage strong augmentations to suppress the intra-domain connectivity and use cross domain neighbors as positive samples to increase intra-class connectivity by enforcing the similarity over cross domain samples potentially from the same category.
- We propose a novel cross domain double-lock nearest neighbors search strategy to find more trustworthy cross domain neighbors and improve it by a novel in-domain cycle nearest neighbors search strategy to further boost the semantic connectivity.

Experiments show our DN^2A outperforms state-of-the-art methods by a large margin, e.g., 12.01% and 13.11% accuracy gains with only 1% labels for linear evaluation on PACS and DomainNet, respectively. Besides, with less than 4% samples compared to ImageNet for training, our method outperforms ImageNet pretraining, showing a promising way to initialize models for the DG problem.

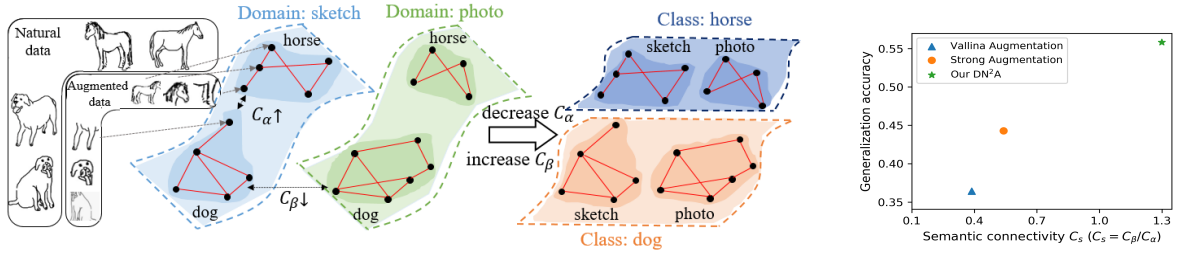
2. Related Work

Domain Generalization. Most domain generalization (DG) methods assume an adequate amount of labeled data for training. A common approach is domain invariant learning via kernel methods [9, 25] or adversarial learning [21, 22]. Many works adopt data augmentation [29, 35] to generate samples from fictitious domains. Several methods employ optimization-based methods, e.g., meta-learning [19] and Invariant Risk Minimization [1]. Despite promising results, the assumption of sufficient labeled data hinders DG from real applications. Similar to [34], we focus on a new task of Unsupervised DG that trains with unlabeled source data.

Unsupervised Learning. Recent progress in unsupervised learning focuses on contrastive learning, which maximizes the mutual information across different augmentations of the same image [2, 14, 32]. This augmentation invariance is achieved by enforcing similarity over different views of the same image while avoiding model collapse by introducing other images as negative samples. Besides augmented views, nearest neighbors in the learned embedding space are used as positive samples to achieve promising results [8, 18].

Unsupervised Domain Adaptation (UDA). UDA [7, 16, 23, 24] transfers knowledge from labeled source domain to unlabeled target domain. [11] enforces the association loss between the source and target data for domain alignment. CDS [17] performs self-supervised learning (SSL) within a single domain and across two domains. PCS [33] further extends the instance-wise SSL in CDS to prototypical SSL.

Unsupervised Learning for DG. Recently, Zhang et al. [34] present the UDG task and focus on negative selection by reweighting contrastive loss based on domain similarity. However, negative samples mainly serve as noise to avoid the model collapse in contrastive learning. Excessive focus on negative selection suffers from the limited performance gain. BrAD [13] intentionally generates edge-like images as positives, which is a strong human prior and the model fails to learn non-edge features like color and texture. Besides, an additional module is required to be trained for edge mapping. Comparably, we employ strong augmentations to suppress domain information, and find cross domain double-lock nearest neighbors as positives (not imaginary or pre-defined, i.e., representative of actual semantic samples in the dataset) to impose the domain invariance for boosting performance.



(a) Augmentation graph of vanilla method and our method. (b) Acc. vs. Connectivity

Figure 2. (a) Vanilla augmentations generate positive samples with large intra-domain connectivity C_α and small intra-class connectivity C_β. Our method decreases C_α by strong augmentations and increases C_β by using dual nearest neighbors as the positives. (b) Larger semantic connectivity C_s, *i.e.*, larger intra-class and smaller intra-domain connectivity, leads to better generalization accuracy.

3. Methodology

3.1. Problem Formulation

Notations. For a dataset S of N_S samples $\{X_i, y_i, d_i\}_{i=1}^{N_S}$ from a joint distribution P^S on $\mathcal{X} \times \mathcal{Y} \times \mathcal{D}$. $\mathcal{X}, \mathcal{Y}, \mathcal{D}$ are the input, category and domain label space respectively. X, Y, D denotes the corresponding random variables. Let P_X^S, P_Y^S and P_D^S denote the marginal distribution of P^S on X, Y, D respectively. $\mathrm{Supp}(\cdot)$ denotes the support of a distribution.

Unsupervised Domain Generalization (UDG). Let $S_{\mathrm{UL}} = \{X_i, d_i\}_{i=1}^{N_{\mathrm{UL}}}$ be the unlabeled dataset from $P^{S_{\mathrm{UL}}}$ and $S_{\mathrm{L}} = \{X_i, y_i\}_{i=1}^{N_{\mathrm{L}}}$ be the labeled dataset from P^{S_L}. The unknown testing distribution $P^{S_{\mathrm{test}}}$ has no domain overlap with all training data, i.e., $\mathrm{Supp}(P_D^{S_{\mathrm{test}}}) \cap (\mathrm{Supp}(P_D^{S_{\mathrm{UL}}}) \cup \mathrm{Supp}(P_D^{S_{\mathrm{L}}})) = \varnothing$. UDG aims to learn a model with parameters θ that achieves a minimum error on unseen $P^{S_{\mathrm{test}}}$:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(X,Y,D)\sim P^{S_{\mathrm{test}}}} [\ell(X, Y; \theta)] \tag{1}$$

3.2. Preliminary: Vanilla Contrastive Learning

Contrastive learning aims at mapping positive pairs to similar representations while pushing away negative pairs in the embedding space. For any embedded sample z_i, we have the positive embedding z_i^+ (often a random augmentation), many negative embeddings $z^- \in \mathcal{N}_i$, and the InfoNCE loss:

$$\mathcal{L}_{\mathrm{Info}}^i = -\log \frac{\exp\left(z_i \cdot z_i^+/\tau\right)}{\exp\left(z_i \cdot z_i^+/\tau\right) + \sum_{z-\in\mathcal{N}_i} \exp\left(z_i \cdot z^-/\tau\right)} \tag{2}$$

where τ is the temperature. In specific, SimCLR [2] uses random augmentations to generate two views of the image, which are fed into encoder ϕ to obtain $z_i = \phi(aug(x_i))$ and $z_i^+ = \phi(aug(x_i))$. Negative embeddings z^- are formed as all the other embeddings in the mini-batch. Following [34], encoder ϕ is a ResNet-18 with a non-linear projection head.

Vanilla contrastive learning (CL) fails in UDG. We empirically train the model with Eq. (2) on three domains (*Art.*, *Cartoon* and *Sketch*) of PACS [20]. As shown in Fig. 1, the t-sne exhibits vanilla CL fails to learn domain-invariant semantic features but learns domain-biased features. Samples from different domains are clustered and separable, while samples from different classes are indistinguishable. Thus, the learned representation is not domain-invariant, which fails to generalize well on unseen target domains.

Definition 1. *(Semantic Connectivity)* For any input $x \in \mathcal{X}$, let $\mathcal{A}(\cdot|x)$ be the distribution of its augmentations A. Let C be the joint distribution on $\mathcal{X} \times \mathcal{X}$ of augmented views of images x_i, x_j as $C(x_i^+, x_j^+) = \mathcal{A}(x_i^+|x_i)\mathcal{A}(x_j^+|x_j)$. Then we have intra-domain C_α and intra-class C_β connectivity as

$$\begin{aligned} C_\alpha &:= \mathbb{E}_{d\sim P_D^S} \mathbb{E}_{x_i,x_j\sim P_d^{S_{\mathrm{UL}}}} C(x_i^+, x_j^+), \\ C_\beta &:= \mathbb{E}_{y\sim P_Y^S} \mathbb{E}_{x_i,x_j\sim P_y^{S_{\mathrm{UL}}}} C(x_i^+, x_j^+) \end{aligned} \tag{3}$$

Then, *semantic connectivity* is defined as $C_s := C_\beta / C_\alpha$.

Degraded semantic connectivity is responsible for the failure. The key to the success of CL is the assumption that intra-class samples could form a connected graph with proper augmentations [12,31], which we point out as good semantic connectivity. However, this assumption is not satisfied in UDG, with the degraded semantic connectivity in two ways:

- Intra-domain connectivity C_α is too large since pre-defined transformations under the i.i.d hypothesis reserve too much domain-related information.
- Intra-class connectivity C_β is too small since pre-defined transformations cannot overcome significant distribution shifts across domains.

Specifically, as shown in Fig. 2 (a), generated by pre-defined transformations in the i.i.d hypothesis, positive pairs share much domain-relevant information, which induces the model to learn domain-biased features for alignment and results in large intra-domain connectivity. Besides, pre-defined transformations cannot overcome significant distribution shifts across domains (e.g., one can hardly transform a cat from sketch to photo), which leads to small intra-class connectivity in cross domain scenarios. Consequently, a connected graph is more likely formed among intra-domain instead of intra-class samples, which leads to learning domain-clustered features rather than class-clustered (Fig. 1). We empirically evaluate the connectivity on PACS to verify our statements. Fig. 2 (b) shows that smaller C_α and larger C_β (i.e., larger C_s) lead to better generalization accuracy of the unsupervised model. To address the degraded semantic connectivity in UDG, we propose to destroy intra-domain connectivity C_α and construct intra-class connectivity C_β, respectively.

3.3. Destroying C_α via Strong Data Augmentation

As explored in previous works [5, 6], strong augmentations have two common types, i.e., geometric and non-geometric. Specifically, we consider 14 types of augmentations with significant magnitude to produce as strong augmentations as possible, detailed in supplementary material.

Proposition 1. *For stronger augmentations $\hat{A}$, i.e., $A \subseteq \hat{A}$, augmented views have smaller intra-domain connectivity as* $\hat{C}_\alpha := \mathbb{E}_{d\sim P_D^S}\mathbb{E}_{x_i,x_j\sim P_d^{S_{\text{UL}}}}[\hat{\mathcal{A}}(x_i^+|x_i)\hat{\mathcal{A}}(x_j^+|x_j)]$.

Proof. Please refer to the supplementary material. □

3.4. Constructing C_β by Dual Nearest Neighbors

Transformations cannot overcome significant distribution shifts across different domains, e.g., one can hardly transform a cat from sketch to photo. Thus, we search for cross domain nearest neighbors (NN) in the embedding space as positive samples. In this way, we can link multiple cross domain samples potentially belonging to the same semantic class to increase the cross domain intra-class connectivity. In addition, the intra-domain gap also exists due to some degree of semantic variation, e.g., different shapes and backgrounds. Thus, we further improve it by employing in-domain NN as positive samples to increase intra-domain intra-class connectivity. Totally, our dual nearest neighbors, i.e., cross domain and in-domain NN, increase intra-class connectivity in UDG.

Proposition 2. *Dual nearest neighbors can increase the intra-class connectivity as* $\hat{C}_\beta := \mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_y^{S_{\text{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(\mathcal{NN}(x_j)^+|\mathcal{NN}(x_j))]$, *where* $\hat{C}_\beta > C_\beta$. *More accurate cross domain NN and more diverse in-domain NN can further increase the intra-class connectivity.*

Proof. Please refer to the supplementary material. □

Specifically, for a given sample x_j and its embedding z_j, we have a *cross domain support set* of embeddings belonging to different domains $Q_z = \{z_1^q, ..., z_k^q, ..., z_{|Q_z|}^q\}$, where $d_j \neq d_k^q$. We propose to search z_j's NN in the support set Q_z as the positive sample.

$$z_j^{nn} = N(z_j, Q_z) = \arg\min_{z_k^q \in Q_z} \|z_j - z_k^q\|_2 \tag{4}$$

Cross Domain Double-lock Nearest Neighbors (CD^{2}NN). Due to huge distribution shifts, directly searching the nearest neighbor (NN) in the cross domain support set may lead to false matches, i.e., query and its NN have different category labels. As a result, directly using cross domain NN as positives may introduce noise in unsupervised learning and compromise the final result. Since searching NN within a domain (w/o distribution shift) is more accurate than searching across domains, we propose a novel cross domain double-lock NN search strategy to leverage more accurate in-domain NN as a mediator to find more trustworthy cross domain NN.

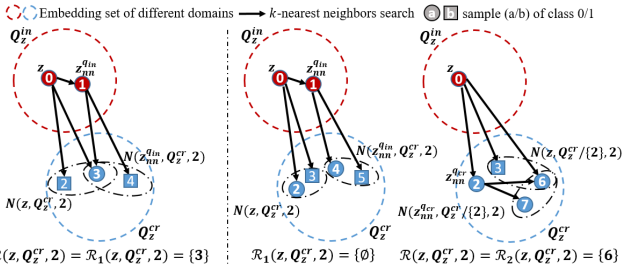


Figure 3. An illustrative example for our proposed CD^{2}NN.

Specifically, given the query embedding z, *in-domain support set* of embeddings within a mini-batch from the same domain Q_z^{in} (for each $z_k^{q_{in}} \in Q_z^{in}$, $d_k^{q_{in}} = d_i$), and *cross domain support set* from different domains Q_z^{cr} (for each $z_k^{q_{cr}} \in Q_z^{cr}$, $d_k^{q_{cr}} \neq d_i$), we define $N(z, Q, k)$ as the k-nearest neighbors (k-NN) of z in Q. We have in-domain NN of z as $N(z, Q_z^{in}, 1) = z_{nn}^{q_{in}}$ and cross domain NN as $N(z, Q_z^{cr}, 1) = z_{nn}^{q_{cr}}$. Our CD^{2}NN $\mathcal{R}(z, Q_z^{cr}, k)$ is defined

$$\mathcal{R}_1(z, Q_z^{cr}, k) = \{z_i^{cr} \mid (z_i^{cr} \in N(z, Q_z^{cr}, k)) \wedge (z_i^{cr} \in N(z_{nn}^{q_{in}}, Q_z^{cr}, k))\} \tag{5}$$

$$\mathcal{R}_2(z, Q_z^{cr}, k) = \{z_j^{cr} \mid \left(z_j^{cr} \in N(z, Q_z^{cr}\backslash z_{nn}^{q_{cr}}, k)\right) \wedge \left(z_j^{cr} \in N(z_{nn}^{q_{cr}}, Q_z^{cr}\backslash z_{nn}^{q_{cr}}, k)\right)\} \tag{6}$$

$$\mathcal{R}(z, Q_z^{cr}, k) = \begin{cases} \mathcal{R}_1(z, Q_z^{cr}, k), & \mathcal{R}_1(z, Q_z^{cr}, k) \neq \varnothing \\ \mathcal{R}_2(z, Q_z^{cr}, k), & \mathcal{R}_1(z, Q_z^{cr}, k) = \varnothing \end{cases} \tag{7}$$

where $\mathcal{R}_1$ and $\mathcal{R}_2$ leverage the in-domain neighbor in Q_z^{in} and Q_z^{cr} to improve the accuracy of cross domain neighbors, respectively. As an illustrative example shown in Fig. 3, for the given query z, directly searching NN results in $\{2\}$, which is a wrong match with the different class label. By CD^{2}NN based on in-domain neighbor $\{1\}$, an accurate neighbor $\{3\}$ is found (left part in Fig. 3). When there are no matches meeting the rule, i.e., $\mathcal{R}_1 = \varnothing$, we further leverage CD^{2}NN based on in-domain neighbor $\{6, 7\}$ to recall the accurate neighbor $\{6\}$ (right part in Fig. 3). Notice that if $\mathcal{R} = \varnothing$, the cross domain neighbor of current query is not trustworthy enough to be used as the positive sample. When $|\mathcal{R}| \geq 1$, we just select the top-1 ranked neighbor as the positive sample, denoted as $\mathcal{R}(z, Q_z^{cr})$ in short.

Proposition 3. *Our proposed CD^{2}NN is more accurate than cross domain NN in the UDG setting.*

Proof. Please refer to the supplementary material. □

In-domain Cycle Nearest Neighbors (ICNN). Directly searching NN in the in-domain support set may fail to find sufficiently diverse samples to overcome intra-domain semantic variances. Thus, we resort to more distinct cross domain NN as a mediator to find more diverse in-domain NN. To ensure the reliability, we employ our proposed CD^{2}NN as the cross domain NN $\mathcal{R}(z, Q_z^{cr})$. Then, we search for the cross domain NN of $\mathcal{R}(z, Q_z^{cr})$ as in-domain cycle NN of z as $N(\mathcal{R}(z, Q_z^{cr}), Q_z^{in}, 1)$, denoted as $\mathcal{C}(z, Q_z^{in})$ in short.

Target domain	Photo	Art.	Cartoon	Sketch	Avg.
Label Fraction 1%					
ERM	10.90	11.21	14.33	18.83	13.82
MoCo V2 [4, 14]	22.97	15.58	23.65	25.27	21.87
AdCo [15]	26.13	17.11	22.96	23.37	22.39
SimCLR V2 [3]	30.94	17.43	30.16	25.20	25.93
DIUL [34]	27.78	19.82	27.51	29.54	26.16
BrAD [13] (KNN)	55.00	35.54	38.12	34.14	40.70
BrAD [13] (linear cls.)	61.81	33.57	43.47	36.37	43.81
Ours (KNN)	66.37	42.68	49.85	54.37	53.32
Ours (linear cls.)	**69.15**	**46.04**	**51.19**	**56.88**	**55.82**
Label Fraction 5%					
ERM	14.15	18.67	13.37	18.34	16.13
MoCo V2 [4, 14]	37.39	25.57	28.11	31.16	30.56
AdCo [15]	37.65	28.21	28.52	30.35	31.18
SimCLR V2 [3]	54.67	35.92	35.31	36.84	40.68
DIUL [34]	44.61	39.25	36.41	36.53	39.20
BrAD [13] (KNN)	58.66	39.11	45.37	46.11	47.31
BrAD [13] (linear cls.)	65.22	41.35	50.88	50.68	52.03
Ours (KNN)	68.93	46.83	54.40	59.92	57.52
Ours (linear cls.)	**73.16**	**52.20**	**59.75**	**66.43**	**62.89**
Label Fraction 10%					
ERM	16.27	16.62	18.40	12.01	15.82
MoCo V2 [4, 14]	44.19	25.85	33.53	24.97	32.14
AdCo [15]	46.51	30.21	31.45	22.96	32.78
SimCLR V2 [3]	54.65	37.65	46.00	28.25	41.64
DIUL [34]	53.37	39.91	46.41	30.17	42.47
BrAD [13] (KNN)	67.20	41.99	45.32	50.04	51.14
BrAD [13] (linear cls.)	72.17	44.20	50.01	55.66	55.51
Ours (KNN)	69.73	50.29	59.22	64.95	61.05
Ours (linear cls.)	**75.41**	**53.14**	**63.69**	**68.57**	**65.20**

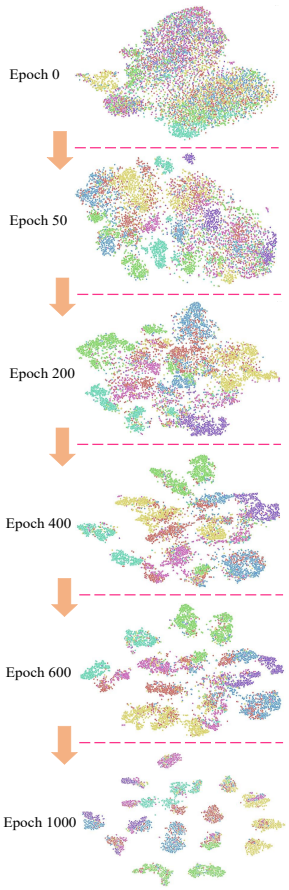


Table 1. **Left: Accuracy (%) results of the *all correlated* setting on PACS.** For each target domain, all other 3 are used as source domains for training. All methods use ResNet18 as the backbone and are pretrained for 1000 epochs before training on few labeled (source only) data. All baselines use a linear classifier (we also include a KNN result w/o any supervised training). ERM indicates the randomly initialized model. Avg. indicates the mean of per-domain accuracies. The reported results are averaged over 3 runs. All baseline results are taken from [34]. The best results are in bold. **Right: Epochwise t-SNE for our method**. T-SNE of ℓ_2-normalized features for all classes.

In summary, with positive samples generated by strong augmentation, cross domain double-lock NN and in-domain cycle NN, we have the total loss as below.

$$\mathcal{L}^i_{\text{ours}} = \mathcal{L}^i_{\text{Info}} - \lambda \cdot |\mathcal{R}| \log \frac{\exp\left(\mathcal{R}(z_i, Q^{cr}_{z_i}) \cdot z^+_i/\tau\right)}{\sum_{j=1}^n \exp\left(\mathcal{R}(z_i, Q^{cr}_{z_i}) \cdot z^+_j/\tau\right)} - \lambda \cdot |\mathcal{C}| \log \frac{\exp\left(\mathcal{C}(z_i, Q^{in}_{z_i}) \cdot z^+_i/\tau\right)}{\sum_{j=1}^n \exp\left(\mathcal{C}(z_i, Q^{in}_{z_i}) \cdot z^+_j/\tau\right)} \quad (8)$$

At the beginning of training, the discovered neighbors are unreliable due to random initialization. As the training proceeds, the searched neighbors are more and more reliable. Thus, λ is set as time-dependent. In practice, a simple binary ramp-up function works well sufficiently, i.e., $\lambda(t) = 0$ in the first T epochs, and $\lambda(t) = 1$ when $t > T$.

4. Experiments

4.1. Experimental Settings

Settings and Datasets. Following [34], we conduct two real-world UDG settings on benchmark datasets **DomainNet** [26] and **PACS** [20], namely ***all correlated*** and ***domain correlated***. ***All correlated*** indicates the unlabeled and labeled data are homologous in the category and domain spaces, i.e., $\text{Supp}(P_D^{S_{UL}}) = \text{Supp}(P_D^{S_L})$ and $\text{Supp}(P_Y^{S_{UL}}) = \text{Supp}(P_Y^{S_L})$. ***Domain correlated*** indicates the unlabeled and labeled data share the same domain space but different categories, i.e., $\text{Supp}(P_D^{S_{UL}}) = \text{Supp}(P_D^{S_L})$ and $\text{Supp}(P_Y^{S_{UL}}) \cap \text{Supp}(P_Y^{S_L}) = \varnothing$. Extensive experiments on ***open-set domain generalization*** and ***few-shot domain adaptation*** are in the supplementary material.

Implementation Details. For unsupervised training, based on SimCLR [2], we adopt ResNet-18 as the backbone, and use the projection head with two MLP layers mapping the features to 128-d and with ℓ_2-norm on top. We strictly follow the protocol of existing UDG methods [13, 34], including same backbone, same number of epochs, and same subset of classes used for training and testing. We use batches of size 128, Adam optimizer with lr $3e^{-4}$ and cosine LR-schedule for 1000 epochs training. For *all correlated*, we evaluate with linear probing and KNN accuracy. For *domain correlated*, due to category shift, we evaluate the model af-

Source domains	{Paint. ∪ Real ∪ Sketch}			{Clipart ∪ Info. ∪ Quick.}				
Target domain	Clipart	Info.	Quick.	Painting	Real	Sketch	Overall	Avg.
Label Fraction 1%								
ERM	6.54	2.96	5.00	6.68	6.97	7.25	5.88	5.89
MoCo V2 [4, 14]	18.85	10.57	6.32	11.38	14.97	15.28	12.12	12.90
AdCo [15]	16.16	12.26	5.65	11.13	16.53	17.19	12.47	13.15
SimCLR V2 [3]	23.51	15.42	5.29	20.25	17.84	18.85	15.46	16.55
DIUL [34]	18.53	10.62	12.65	14.45	21.68	21.30	16.56	16.53
BrAD [13] (KNN)	40.65	14.00	21.28	16.80	22.29	25.72	22.35	23.46
BrAD [13] (linear cls.)	47.26	16.89	23.74	20.03	25.08	31.67	25.85	27.45
Ours (KNN)	62.31	23.84	27.50	29.71	37.07	45.48	35.21	37.65
Ours (linear cls.)	**68.02**	**24.45**	**29.20**	**31.16**	**37.91**	**52.62**	**37.43**	**40.56**
Label Fraction 5%								
ERM	10.21	7.08	5.34	7.45	6.08	5.00	6.50	6.86
MoCo V2 [4, 14]	28.13	13.79	9.67	20.80	24.91	21.44	18.99	19.79
AdCo [15]	30.77	18.65	7.75	19.97	24.31	24.19	19.42	20.94
SimCLR V2 [3]	34.03	17.17	10.88	21.35	24.34	27.46	20.89	22.54
DIUL [34]	39.32	19.09	10.50	21.09	30.51	28.49	23.31	24.83
BrAD [13] (KNN)	55.75	18.15	26.93	24.29	33.33	37.54	31.12	32.66
BrAD [13] (linear cls.)	64.01	25.02	29.64	29.32	34.95	44.09	35.37	37.84
Ours (KNN)	66.54	23.98	34.47	37.89	44.65	54.57	41.64	43.68
Ours (linear cls.)	**70.10**	**27.31**	**36.77**	**40.93**	**47.20**	**60.05**	**44.98**	**47.06**
Label Fraction 10%								
ERM	15.10	9.39	7.11	9.90	9.19	5.12	8.94	9.30
MoCo V2 [4, 14]	32.46	18.54	8.05	25.35	29.91	23.71	21.87	23.05
AdCo [15]	32.25	17.96	11.56	23.35	29.98	27.57	22.79	23.78
SimCLR V2 [3]	37.11	19.87	12.33	24.01	30.17	31.58	24.28	25.84
DIUL [34]	35.15	20.88	15.69	25.90	33.29	30.77	26.09	26.95
BrAD [13] (KNN)	60.78	19.76	31.56	26.06	37.43	41.38	34.77	36.16
BrAD [13] (linear cls.)	68.27	26.60	34.03	31.08	38.48	48.17	38.74	41.10
Ours (KNN)	66.73	22.15	35.93	36.42	46.12	57.14	42.21	44.08
Ours (linear cls.)	**73.04**	**28.23**	**37.80**	**41.77**	**50.94**	**61.69**	**46.72**	**48.91**

Table 2. Accuracy (%) of *all correlated* setting on DomainNet. Overall and Avg. indicate the overall test accuracy and mean of per-domain accuracy respectively. They are different since the test sets of different domains are not of the same size. See Table 1 caption for other details.

ter finetuning 30 epochs with lr $1e^{-3}$. Please refer to the supplementary material for other implementation details.

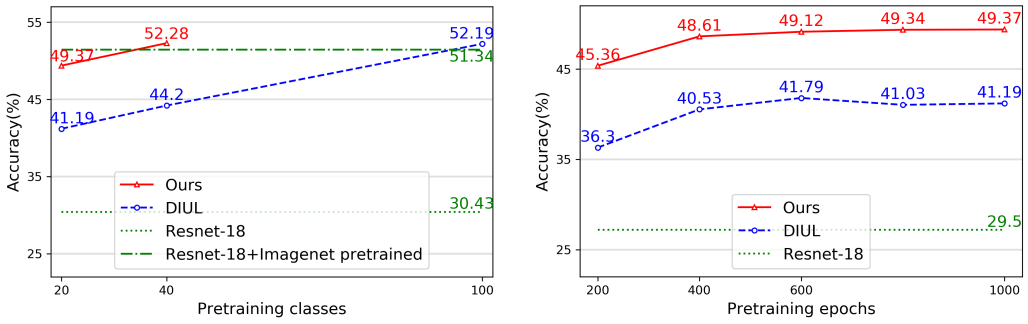
4.2. Experimental Results

All correlated UDG. Following [34], we evaluate the generalization ability under *all correlated* setting, where the proportion of labeled data varies from 1% to 10%. As shown in Table 1 (PACS) and 2 (DomainNet), our method achieves SOTA result. Compared with vanilla CL methods, our DN^{2}A achieves a significant improvement, i.e., 23.56% and 23.07% better than SimCLR V2 on PACS and DomainNet with 10% labeled data, respectively. Vanilla methods learn domain-biased features and fail to generalize well. While our method learns domain-invariant semantic features and forms semantic clusters in the feature space as shown with t-SNE in Table 1. Besides, compared with the UDG method DIUL [34], we achieve 23.69% and 22.23% performance gain on PACS and DomainNet with 5% labeled data, respectively. DIUL focuses on negative sample selection with domain-specific images, but suffers limited performance gain, since negative samples mainly serve as noise to avoid the trivial solution in CL. We argue the key lies in positive samples and achieve better results with the proposed positive selection strategy. Moreover, our method outperforms SOTA UDG method BrAD [13] by 12.01% and 13.11% on PACS and DomainNet with label fractions of 1%, respectively. BrAD generates edge-like images as positive samples with strong human prior, and fails to learn non-edge features (*e.g.,* color, texture), which could also contain semantic information (*e.g.,* yellow spot patterns for giraffe in photo). In contrast, we use cross domain neighbors in the embedding space as positive samples, which are not imaginary and pre-defined, i.e., representative of actual semantic samples in the given dataset.

Domain correlated UDG. *Domain correlated* is a more challenging setting to evaluate the generalization ability of UDG methods in the real world under both domain and category shifts. Following [34], we adopt DomainNet with 20 categories for labeled training and testing and the other 40 categories for unlabeled training. As shown in Table 3, our method achieves the best generalization accuracy on all the domains. We outperform vanilla CL methods by a large margin, i.e., 9.48% and 13.14% better than SimCLR V2 and MoCo V2, respectively. Though categories for unlabeled training are different from those for labeled training and testing, where the learned representation is not directly helpful, our method can achieve promising results by excluding domain-related features and maintaining domain-invariant representation space. Compared to DIUL, we achieve 7.97% performance gain, showing the effectiveness of our DN^{2}A.

Source domains	{Paint. ∪ Real ∪ Sketch}			{Clipart ∪ Info. ∪ Quick.}				
Target domain	Clipart	Info.	Quick.	Painting	Real	Sketch	Overall	Avg.
ERM	55.78	22.40	25.75	31.92	41.58	24.10	33.23	33.59
BYOL [10]	58.39	23.99	28.56	33.73	45.63	25.48	35.89	35.96
MoCo V2 [4, 14]	72.84	33.40	34.20	45.83	60.75	43.98	47.78	48.50
AdCo [15]	76.61	31.55	33.42	43.77	64.58	47.76	48.85	49.62
SimCLR V2 [3]	75.58	35.52	37.08	47.94	62.40	54.47	50.91	52.16
DIUL [34]	78.40	33.98	39.87	47.82	65.07	56.90	52.64	53.67
Ours	**84.13**	**41.61**	**48.12**	**58.61**	**69.09**	**68.28**	**60.23**	**61.64**

Table 3. Accuracy (%) results of the *domain correlated* setting on DomainNet.



(a) Pretraining class. (b) Pretraining epoch.

Figure 4. Accuracy (%) results of our method with different pretraining (a) classes and (b) epochs.

Comparison with ImageNet Pretrained Models. Current DG methods use the models pretrained on ImageNet as initialization. While our method can outperform ImageNet pretrained models by unsupervised training on significantly less unlabeled data. Pretraining on i.i.d. ImageNet fails to be invariant to large intra-class variances caused by strong distribution shifts. Given data with strong heterogeneity, our method can learn domain-invariant representations and generalize well. As shown in Fig. 4 (a), when unlabeled data for pretraining are of 40 classes from DomainNet, our method outperforms ImageNet pretrained initialization by 0.94%. Note that the amount of data used for pretraining is less than 4% of ImageNet. Besides, DN^{2}A outperforms DIUL by a large margin. With only 40 pretraining classes, we achieve comparable accuracy with DIUL using 100 classes.

4.3. Ablation Study

We conduct experiments on PACS for *all correlated* UDG. Unless specified, all models are unsupervisedly pretrained for 600 epochs. KNN accuracy with 5% label is reported.

Effects of Augmentation Strategies. As shown in Table 4, strong augmentations (SA) can improve the baseline by 7.85% accuracy. As aforementioned, SA can generate positive samples with less domain-related information and make the learned model exclude domain-biased features.

Effects of Cross Domain Double-lock Nearest Neighbor. This experiment is conducted without the in-domain cycle NN to evaluate the performance of CD^{2}NN. Table 4 shows that our proposed CD^{2}NN achieves 10.46% accuracy gain by overcoming distribution shifts and learning domain-invariant features. Besides, we compare various neighbor selection strategies. In-domain: use in-domain NN $N(z, Q_z^{in}, 1)$ as the positive; Vanilla: directly search for cross domain NN $N(z, Q_z^{cr}, 1)$ as the positive; Ours: use CD^{2}NN $\mathcal{R}(z, Q_z^{cr})$ as the positive; GT labels: use ground-truth labels to construct cross-domain intra-class samples as the positive, which is the upper bound performance. As shown in Table 5, In-domain suffers limited performance, since in-domain NN fails to overcome distribution shifts across domains. Though Vanilla can achieve better performance by using cross domain NN, it is undermined by the noise of NN searched across different distributions. Our method improves Vanilla by 3.13% accuracy, showing the effectiveness of our CD^{2}NN to find more accurate NNs for boosting performance.

SA	CD^{2}NN	ICNN	Photo	Art.	Cartoon	Sketch	Avg.
✗	✗	✗	38.80	30.17	33.61	43.08	36.42
✓	✗	✗	53.77	34.08	40.64	48.58	44.27
✓	✓	✗	67.66	43.48	52.22	55.54	54.73
✓	✓	✓	**67.84**	**44.06**	**53.98**	**57.43**	**55.82**

Table 4. Ablation on strong augmentation, cross domain double-lock NN and in-domain cycle NN.

	Photo	Art.	Cartoon	Sketch	Avg.
GT labels	69.94	51.45	57.38	62.97	60.43
In-domain	62.04	38.99	46.38	47.58	48.75
Vanilla	65.22	40.85	49.88	50.44	51.60
Ours	**67.66**	**43.48**	**52.22**	**55.54**	**54.73**

Table 5. Ablation on Cross Domain Neighbor Selection in CD^{2}NN.

	Photo	Art.	Cartoon	Sketch	Avg.
Vanilla	67.43	43.54	52.99	56.72	55.17
Ours	**67.84**	**44.06**	**53.98**	**57.43**	**55.82**

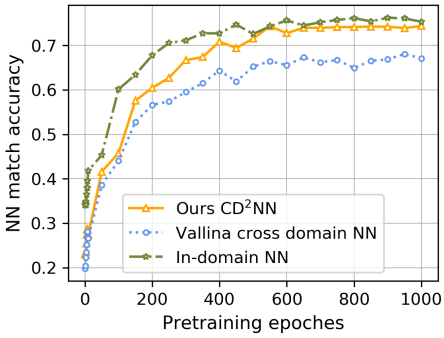
Table 6. Ablation on our In-domain Cycle NN.

	Photo	Art.	Cartoon	Sketch	Avg.
k=1	**67.84**	**44.06**	53.98	**57.43**	**55.82**
k=2	66.62	43.37	**54.21**	56.80	55.25
k=4	65.57	40.79	54.13	54.93	53.86
k=8	63.91	39.04	53.27	54.81	52.76

Table 7. Ablation on k in our proposed dual nearest neighbors (*i.e.,* in both CD^{2}NN and ICNN).

Batch	Temp.	Photo	Art.	Cartoon	Sketch	Avg.
128	0.15	67.43	43.72	53.20	56.19	55.14
128	0.07	67.84	44.06	53.98	57.43	55.82
128	0.03	66.71	44.85	53.81	56.87	55.56
256	0.07	66.73	44.98	54.46	58.15	56.08
64	0.07	65.57	42.29	51.84	55.42	53.78

Table 8. Ablation on batch size and temperature.



(a) NN match accuracy.

(b) NN searched by our method and vanilla SimCLR.

Figure 5. (a) NN match accuracy. (b) NN searched by vanilla SimCLR and our method.

Effects of In-domain Cycle Nearest Neighbor. Table 6 shows that our in-domain cycle nearest neighbor achieves 1.09% accuracy gain by overcoming the intra-domain gap. Besides, compared with vanilla in-domain NN, our ICNN can achieve 0.65% gain by exploring more diverse samples.

Effects of k in Nearest Neighbors. In experiments, we select the top-1 ranked neighbor as the positive. Here we investigate whether increasing the diversity of neighbors (i.e., increasing k) results in improved performance. As shown in Table 7, although our method is somewhat robust to changing the value of k, increasing beyond $k = 1$ always results in slight degradation due to the brought noise.

Effects of Epochs, Batch Size and Temperature. As shown in Fig. 4 (b), with a small number of 200 epochs, our DN^{2}A outperforms DIUL by a large margin of 9.06% accuracy. As the epoch increases, our method exceeds DIUL by 8.18% at 1000 epochs. Table 8 shows larger batch sizes improve performance with the increased diversity of negative samples. A relatively small temperature with stronger penalty for compactness and separability is more effective for classification.

4.4. Discussion

Visualization of Feature Space. Fig. 1 (a) shows vanilla method learns a feature space with domain-related information where domains are separable. While in our feature space, samples from different domains are closely entangled, indicating the learning of domain-invariant features. As shown in Fig. 1, vanilla method fails to learn semantic features and samples from different classes are inseparable. By contrast, semantic clusters are clearly formed in our feature space.

Nearest Neighbor Match Accuracy. Fig. 5 (a) shows how the accuracy of searched NN (i.e. from the same class) for three strategies (in-domain NN, vanilla cross domain NN and our CD^{2}NN) varies as training proceeds. In-domain NN has high accuracy due to no distribution shifts. Vanilla cross domain NN leads to many wrong matches due to domain shifts. For our method that leverages in-domain NN to find more trustworthy cross domain NN, though starting with the low accuracy of rough 20%, the accuracy of picking the right neighbor achieves about 74% at the end of training, which approximates the highest in-domain NN accuracy. Besides, we show a random batch of NN retrieved in Fig. 5 (b). The NNs picked by our method are from the same semantic class. For the vanilla method, the retrieval is mainly based on domain-relevant information, *e.g.,* style and texture.

5. Conclusion

In this paper, we first figure out the failure of vanilla contrastive learning in the UDG task is due to large intra-domain and small intra-class connectivity of positive samples generated by pre-defined augmentations under the i.i.d hypothesis. Thus, we leverage strong augmentations to suppress domain-biased information and propose to use a novel cross domain double-lock nearest neighbors as positives, which effectively link different domain samples belonging to the same class. Besides, in-domain cycle nearest neighbors are incorporated to further overcome intra-domain variances. Experimentally, our method achieves state-of-the-art results on the UDG task.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 62250055, Grant 61932022, Grant 61931023, Grant 61971285, Grant 61831018, Grant 62120106007, Grant 61972256, Grant T2122024, Grant 62125109, and in part by the Program of Shanghai Science and Technology Innovation Project under Grant 20511100100.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 2

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020. 1, 2, 3, 5

[3] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33:22243–22255, 2020. 5, 6, 7

[4] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 5, 6, 7

[5] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. AutoAugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019. 4

[6] 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. 4

[7] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3937–3946, 2021. 2

[8] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021. 2

[9] Muhammad Ghifary, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1414–1430, 2016. 2

[10] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 7

[11] Philip Haeusser, Thomas Frerix, Alexander Mordvintsev, and Daniel Cremers. Associative domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2765–2773, 2017. 2

[12] Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34:5000–5011, 2021. 1, 3

[13] Sivan Harary, Eli Schwartz, Assaf Arbelle, Peter Staar, Shady Abu-Hussein, Elad Amrani, Roei Herzig, Amit Alfassy, Raja Giryes, Hilde Kuehne, et al. Unsupervised domain generalization by learning a bridge across domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5280–5290, 2022. 2, 5, 6

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 2, 5, 6, 7

[15] Qianjiang Hu, Xiao Wang, Wei Hu, and Guo-Jun Qi. AdCo: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1074–1083, 2021. 5, 6, 7

[16] Jiaxing Huang, Dayan Guan, Aoran Xiao, Shijian Lu, and Ling Shao. Category contrast for unsupervised domain adaptation in visual tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1203–1214, 2022. 2

[17] Donghyun Kim, Kuniaki Saito, Tae-Hyun Oh, Bryan A Plummer, Stan Sclaroff, and Kate Saenko. CDS: Cross-domain self-supervised pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9123–9132, 2021. 2

[18] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Mean shift for self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10326–10335, 2021. 2

[19] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2

[20] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5542–5550, 2017. 3, 5

[21] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018. 2

[22] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision*, pages 624–639, 2018. 2

[23] Yuchen Liu, Yabo Chen, Wenrui Dai, Mengran Gou, Chun-Ting Huang, and Hongkai Xiong. Source-free domain adaptation with contrastive domain alignment and self-supervised exploration for face anti-spoofing. In *European Conference on Computer Vision*, pages 511–528. Springer, 2022. 2

[24] You-Wei Luo, Chuan-Xian Ren, Dao-Qing Dai, and Hong Yan. Unsupervised domain adaptation via discriminative manifold propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1653–1669, 2020. 2

[25] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013. 1, 2

[26] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019. 5

[27] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021. 1

[28] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision*, pages 776–794. Springer, 2020. 1

[29] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[30] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Wenjun Zeng, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 4627–4635, 2021. 1

[31] Yifei Wang, Qi Zhang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. In *International Conference on Learning Representations*, 2022. 1, 3

[32] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 1, 2

[33] Xiangyu Yue, Zangwei Zheng, Shanghang Zhang, Yang Gao, Trevor Darrell, Kurt Keutzer, and Alberto Sangiovanni Vincentelli. Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13834–13844, 2021. 2

[34] Xingxuan Zhang, Linjun Zhou, Renzhe Xu, Peng Cui, Zheyan Shen, and Haoxin Liu. Towards unsupervised domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4910–4920, 2022. 1, 2, 3, 5, 6, 7

[35] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European Conference on Computer Vision*, pages 561–578. Springer, 2020. 1, 2