PROMOTING SEMANTIC CONNECTIVITY: DUAL NEAREST NEIGHBORS CONTRASTIVE LEARNING FOR UNSUPERVISED DOMAIN GENERALIZATION

Anonymous authors
Paper under double-blind review

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. Thus, 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^2A). Our DN^2A 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^2A 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 test data share the same distribution. When transferred to practical applications, these methods perform poorly on out-of-distribution data due to domain shift (Shen et al., 2021; Wang et al., 2021). To tackle this issue, Domain Generalization (DG) methods (Muandet et al., 2013; Zhou et al., 2020) learn transferable knowledge from multiple source domains to generalize on unseen target domains. Despite promising results of DG, the assumption of large-scale labeled source data hinders practical use due to the expensive and cumbersome annotation capture. Thus, we study the more practical unsupervised domain generalization (UDG) (Zhang et al., 2022) problem to learn domain-invariant features in an unsupervised manner.

Recent advances in unsupervised learning prefer contrastive learning (CL) (Wu et al., 2018; Chen et al., 2020; He et al., 2020; Tian et al., 2020), which learns semantic representation by exploiting invariance (similarity over different views of the same image) to various data transformations. 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, current CL methods fail to learn domain-invariant semantic features but learn 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 (HaoChen et al., 2021; Wang et al., 2022), semantic connectivity is the support overlap of augmented samples within the same semantic class. We further point out that the degraded semantic connectivity is responsible for the failure of vanilla CL in UDG, which is reflected in two folds: large intra-domain connectivity and small intra-class connectivity. For one thing, positive samples generated by standard augmentations under the i.i.d. hypothesis share too much domain-relevant information. The domain-relevant information induces the model to employ domain-related features for alignment, resulting in large intra-domain connectivity. For another, domain invariance is hard to capture via handcrafted transformations due to significant distribution shifts across domains (*e.g.*, one can hardly transform a cat in sketch to photo). Thus, small intra-class connectivity occurs in cross-domain scenarios.

To address these problems, we propose to suppress the intra-domain connectivity while enlarging 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 leads to the decreased intra-domain connectivity, and the learned unsupervised representation can achieve a greater 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 over cross domain samples potentially belonging to the same category, which can increase the cross-domain intra-class connectivity. Additionally, we improve cross domain NN by a dual NN strategy which 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 (w/o distribution shift) is more accurate than across domain, we propose a novel Cross Domain Double-lock NN (CD^{2}NN) 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 directly 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^{2}NN 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 problem of contrastive learning in UDG, and propose a novel approach, namely Dual Nearest Neighbors contrastive learning with strong Augmentation (DN^{2}A) to increase the semantic connectivity with theoretical proofs.
- We propose to leverage strong augmentations to suppress the intra-domain connectivity and use cross domain neighbors as positive samples to increase the intra-class connectivity by enforcing the similarity over cross domain samples potentially belonging to 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.

Experimentally, our DN^{2}A outperforms state-of-the-art methods 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. Besides, with less than 4% samples compared with 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 (Muandet et al., 2013; Ghifary et al., 2016) or domain-adversarial learning (Li et al., 2018b;c). Many works propose data augmentation (Volpi et al., 2018; Zhou et al., 2020) to generate samples from fictitious domains. There are several methods that leverage optimization-based methods, *e.g.*, meta-learning (Li et al., 2018a) and Invariant Risk Minimization (Arjovsky et al., 2019). Despite promising results, assumption of sufficient labeled data may hinder DG from real applications. Similar to (Zhang et al., 2022), we focus on a new task of Unsupervised DG (UDG) that trains with unlabeled source data.

Unsupervised Learning. Recent progress focuses on contrastive learning, which maximizes the mutual information across different augmentations of the same image (Wu et al., 2018; Chen et al., 2020; He et al., 2020). 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 positives to achieve promising results (Dwibedi et al., 2021; Koohpayegani et al., 2021).

Unsupervised Domain Adaptation (UDA). UDA aims to transfer the knowledge from a labeled source domain to an unlabeled target domain. Haeusser et al. (2017) enforce the association loss between source and target data. Li et al. (2021) propose domain consensus clustering to learn the intrinsic structure of target data. CDS (Kim et al., 2021) performs self-supervised learning (SSL) within a single domain and across two domains. PCS (Yue et al., 2021) further extends the instance-wise SSL in CDS to prototypical SSL, and proposes a powerful end-to-end UDA framework.

Unsupervised Learning for DG. Recently, Zhang et al. (2022) present the UDG task and focus on negative selection by reweighting the 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 has limited gain in improving the performance. Harary et al. (2021) propose to intentionally generate edge-like images as positive samples, which is a strong human prior and the model fails to learn non-edge related features such as 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 propose to find the cross domain double-lock nearest neighbors as positive samples (that are not imaginary and pre-defined, *i.e.*, they are representative of actual semantic samples in the given dataset) to impose the domain invariance for boosting the performance.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

Notations. Given a dataset S of N_S samples $\{X_i, y_i, d_i\}_{i=1}^{N_S}$ from a joint distribution P^S on $\mathcal{X} \times \mathcal{Y} \times \mathcal{D}$, where $\mathcal{X}$ denotes the input space, $\mathcal{Y}$ is the category label space, and $\mathcal{D}$ is the domain label space. 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. Let $\mathrm{Supp}(\cdot)$ denote 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 $P^{S_{\mathrm{L}}}$. The unknown testing distribution $P^{S_{\text{test}}}$ has no domain overlap with all training data, *i.e.*, $\mathrm{Supp}(P_D^{S_{\text{test}}}) \cap (\mathrm{Supp}(P_D^{S_{\mathrm{UL}}}) \cup \mathrm{Supp}(P_D^{S_{\mathrm{L}}})) = \varnothing$. UDG aims to learn a model with parameters θ that achieves a minimum error on unseen $P^{S_{\text{test}}}$:

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

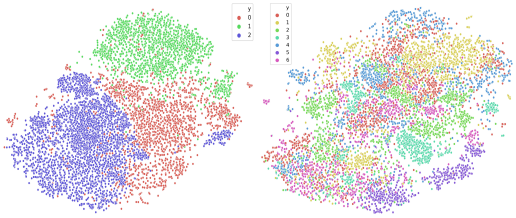
3.2 PRELIMINARY: VANILLA CONTRASTIVE LEARNING

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 particular, SimCLR (Chen et al., 2020) uses random data augmentations to generate two views of the image, which are fed into an 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 (Zhang et al., 2022), the 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 (Li et al., 2017). As shown in Fig. 1, the t-sne of learned features shows that 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 and marginal distribution alignment is not satisfied. Consequently, vanilla CL fails to generalize well on target domains.



(a) Different domains (b) Different classes

Figure 1: The t-sne visualization of unsupervised features learned by SimCLR on PACS.

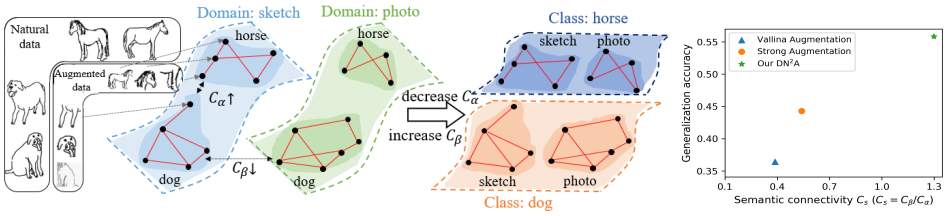
Degraded semantic connectivity is responsible for the failure.

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.

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

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

The key to the success of CL lies in the assumption that intra-class samples could form a connected graph with proper augmentations (HaoChen et al., 2021; Wang et al., 2022), which we point as good semantic connectivity. However, this assumption is not satisfied in UDG, with the degraded semantic connectivity reflected in two ways:



(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 increase 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.

- 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 under the i.i.d hypothesis, positive pairs share too much domain-relevant information, which induces the model to leverage domain-related 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 in sketch to photo), which leads 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 (shown in Fig. 1). We empirically evaluate the connectivity measures 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 learned models. To address the degraded semantic connectivity in UDG, we propose to destroy intra-domain connectivity and construct intra-class connectivity, respectively.

3.3 Destroying Intra-domain Connectivity via Strong Data Augmentation

As explored in previous works (Cubuk et al., 2019; 2020), strong augmentations usually have two 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 the appendix.

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

Proof. Please refer to the appendix. □

3.4 Constructing Intra-class Connectivity by Dual Nearest Neighbors

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

Proposition 2. *Dual nearest neighbors (NN) can increase the intra-class connectivity as $\hat{C}_\beta = \mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_y^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(\mathcal{NN}(x_j)^+|\mathcal{NN}(x_j))]$, 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 appendix. □

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.

$$id_j^{nn} = \underset{k\in\{1,...,|Q_z|\}}{\arg\min} \|z_j - z_k^q\|_2, z_j^{nn} = N(z_j, Q_z) = Q_z[id_j^{nn}] \tag{4}$$

Cross Domain Double-lock Nearest Neighbors (CD^{2}NN). Due to huge distribution shifts, directly searching nearest neighbor (NN) in the cross domain support set may lead to false matches, *i.e.*, the query and its NN have different category labels. As a result, directly using the 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 domain, 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.

Specifically, given the query embedding z, the *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 the *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 the 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 proposed CD^{2}NN $\mathcal{R}(z, Q_z^{cr}, k)$ is defined as

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

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

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

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 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.

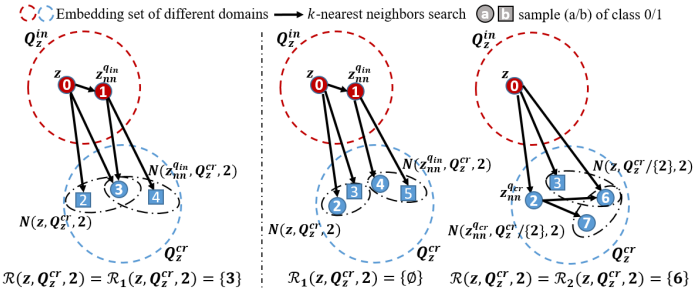


Figure 3: An illustrate example for our proposed CD^{2}NN.

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

Proof. Please refer to the appendix. □

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 the 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.

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. At the beginning of training, the discovered

$$\mathcal{L}_{\text{ours}}^i = \mathcal{L}_{\text{Info}}^i - \lambda \cdot |\mathcal{R}| \log \frac{\exp\left(\mathcal{R}(z_i, Q_{z_i}^{cr}) \cdot z_i^+ / \tau\right)}{\sum_{j=1}^n \exp\left(\mathcal{R}(z_i, Q_{z_i}^{cr}) \cdot z_j^+ / \tau\right)} - \lambda \cdot |\mathcal{C}| \log \frac{\exp\left(\mathcal{C}(z_i, Q_{z_i}^{in}) \cdot z_i^+ / \tau\right)}{\sum_{j=1}^n \exp\left(\mathcal{C}(z_i, Q_{z_i}^{in}) \cdot z_j^+ / \tau\right)} \tag{8}$$

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 $t > T$.

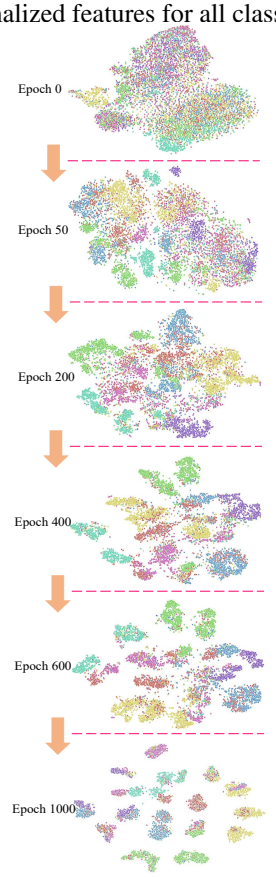
4 Experiments

4.1 Experimental Settings

Settings and Datasets. Following (Zhang et al., 2022), we conduct two real-world UDG settings on benchmark datasets **DomainNet** (Peng et al., 2019) and **PACS** (Li et al., 2017), namely ***all correlated***

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 (Zhang et al., 2022). The best results are in bold. **Right: Epochwise t-SNE for our method**. T-SNE of ℓ_2-normalized features for all classes.

Target domain	Photo	Art.	Cartoon	Sketch	Avg.
Label Fraction 1%					
ERM	10.90	11.21	14.33	18.83	13.82
MoCo V2	22.97	15.58	23.65	25.27	21.87
AdCo	26.13	17.11	22.96	23.37	22.39
SimCLR V2	30.94	17.43	30.16	25.20	25.93
DIUL	27.78	19.82	27.51	29.54	26.16
BrAD (KNN)	55.00	35.54	38.12	34.14	40.70
BrAD (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	37.39	25.57	28.11	31.16	30.56
AdCo	37.65	28.21	28.52	30.35	31.18
SimCLR V2	54.67	35.92	35.31	36.84	40.68
DIUL	44.61	39.25	36.41	36.53	39.20
BrAD (KNN)	58.66	39.11	45.37	46.11	47.31
BrAD (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	44.19	25.85	33.53	24.97	32.14
AdCo	46.51	30.21	31.45	22.96	32.78
SimCLR V2	54.65	37.65	46.00	28.25	41.64
DIUL	53.37	39.91	46.41	30.17	42.47
BrAD (KNN)	67.20	41.99	45.32	50.04	51.14
BrAD (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**



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 appendix.

Implementation Details. For unsupervised training, based on SimCLR (Chen et al., 2020), 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 (Harary et al., 2021; Zhang et al., 2022), 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 after finetuning 30 epochs with lr $1e^{-3}$. Please refer to the appendix for other implementation details.

4.2 Experimental Results

All correlated UDG. Following (Zhang et al., 2022), 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 the 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, which 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.

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

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	18.85	10.57	6.32	11.38	14.97	15.28	12.12	12.90
AdCo	16.16	12.26	5.65	11.13	16.53	17.19	12.47	13.15
SimCLR V2	23.51	15.42	5.29	20.25	17.84	18.85	15.46	16.55
DIUL	18.53	10.62	12.65	14.45	21.68	21.30	16.56	16.53
BrAD (KNN)	40.65	14.00	21.28	16.80	22.29	25.72	22.35	23.46
BrAD (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	28.13	13.79	9.67	20.80	24.91	21.44	18.99	19.79
AdCo	30.77	18.65	7.75	19.97	24.31	24.19	19.42	20.94
SimCLR V2	34.03	17.17	10.88	21.35	24.34	27.46	20.89	22.54
DIUL	39.32	19.09	10.50	21.09	30.51	28.49	23.31	24.83
BrAD (KNN)	55.75	18.15	26.93	24.29	33.33	37.54	31.12	32.66
BrAD (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	32.46	18.54	8.05	25.35	29.91	23.71	21.87	23.05
AdCo	32.25	17.96	11.56	23.35	29.98	27.57	22.79	23.78
SimCLR V2	37.11	19.87	12.33	24.01	30.17	31.58	24.28	25.84
DIUL	35.15	20.88	15.69	25.90	33.29	30.77	26.09	26.95
BrAD (KNN)	60.78	19.76	31.56	26.06	37.43	41.38	34.77	36.16
BrAD (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**

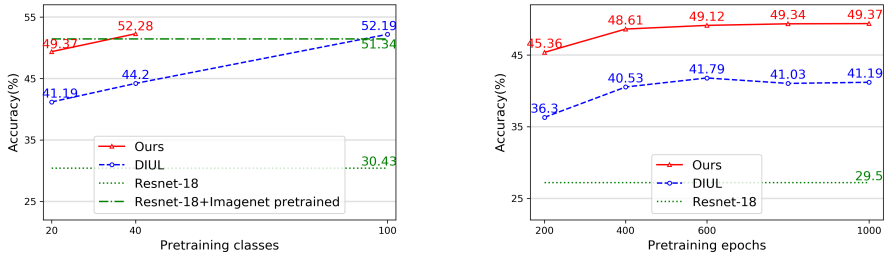
Besides, compared with the UDG method DIUL (Zhang et al., 2022), we achieve 23.69% and 22.23% performance gain on PACS and DomainNet with 5% labeled data, respectively. DIUL focuses on negative sample selection with domain-specific images, but suffers limited performance gain, since negative samples 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 (Harary et al., 2021) 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 related features (*e.g.,* color, texture), which could also contain semantic information (*e.g.,* yellow spot patterns for giraffe in photo). While 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 (Zhang et al., 2022), 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 semantic representation is not directly helpful, our method can achieve promising results by excluding domain-related features and maintaining domain-invariant representation space. Compared with DIUL, we achieve 7.97% performance gain, showing the effectiveness of our proposed UDG method.

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 the 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.

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

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	58.39	23.99	28.56	33.73	45.63	25.48	35.89	35.96
MoCo V2	72.84	33.40	34.20	45.83	60.75	43.98	47.78	48.50
AdCo	76.61	31.55	33.42	43.77	64.58	47.76	48.85	49.62
SimCLR V2	75.58	35.52	37.08	47.94	62.40	54.47	50.91	52.16
DIUL	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**



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

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

4.3 Ablation Study

We conduct experiments on PACS for *all correlated* UDG. Unless specified, all models are unsupervisedly pretrained for 600 epochs, and we report KNN accuracy with 5% labeled data.

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

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**

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, demonstrating the effectiveness of our proposed CD^{2}NN to find more accurate neighbors for boosting the performance.

Table 5: Ablation on NN Selection in CD^{2}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 6: Ablation on In-domain Cycle 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**

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

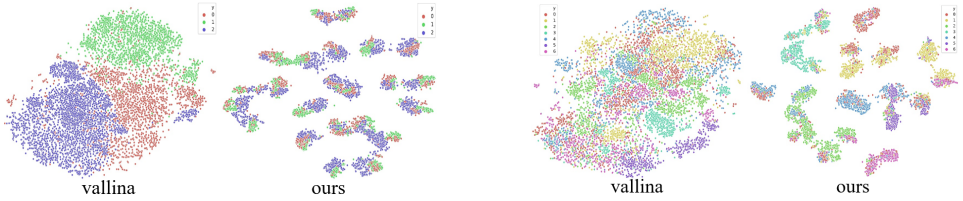
Effects of Epochs, Batch Size and Temperature. As shown in Fig. 4 (b), with a small number of 200 epochs, our method outperforms DIUL by a large margin of 9.06% accuracy. As the epoch increases, our method exceeds DIUL by 8.18% at 1000 epochs. As shown in Table 8, larger batch

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

	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

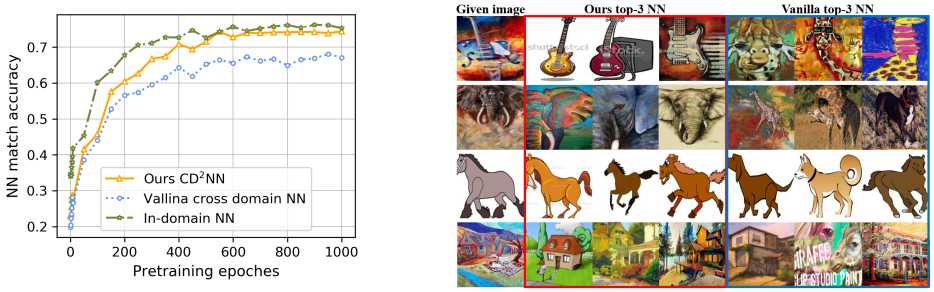
Table 8: Ablation on batch size and temperature.

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



(a) Different colors for different domains (b) Different colors for different classes

Figure 5: T-sne visualization of the learned feature space by vanilla SimCLR and our method.



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

Figure 6: (a) NN match accuracy. (b) NN searched by vanilla SimCLR and our method.

sizes improve the performance with the increased diversity of negative samples. Smaller temperature that indicates stronger penalty for compactness and separability is more effective for classification.

4.4 Discussion

Visualization of Feature Space. Fig. 5 (a) shows that vanilla method learns a embedding 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. 5, vanilla method fails to learn semantic features and samples from different classes are inseparable. By contrast, semantic clusters are clearly formed in our learned feature space.

Nearest Neighbor Match Accuracy. Fig. 6 (a) shows how the accuracy of the 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. 6 (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 connectivity 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-related 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. In addition, in-domain cycle nearest neighbors are incorporated to further overcome intra-domain variances. Experimentally, our DN^{2}A achieves state-of-the-art performance on the UDG task.

REFERENCES

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

Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. In *European Conference on Computer Vision*, pp. 301–318. Springer, 2020.

Liang Chen, Qianjin Du, Yihang Lou, Jianzhong He, Tao Bai, and Minghua Deng. Mutual nearest neighbor contrast and hybrid prototype self-training for universal domain adaptation. 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

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*, pp. 113–123, 2019.

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*, pp. 702–703, 2020.

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*, pp. 9588–9597, 2021.

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.

Philip Haeusser, Thomas Frerix, Alexander Mordvintsev, and Daniel Cremers. Associative domain adaptation. In *Proceedings of the IEEE international conference on computer vision*, pp. 2765–2773, 2017.

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, 2021.

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. *arXiv preprint arXiv:2112.02300*, 2021.

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*, pp. 9729–9738, 2020.

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*, pp. 9123–9132, 2021.

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*, pp. 10326–10335, 2021.

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*, pp. 5542–5550, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018a.

Guangrui Li, Guoliang Kang, Yi Zhu, Yunchao Wei, and Yi Yang. Domain consensus clustering for universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9757–9766, 2021.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5400–5409, 2018b.

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 (ECCV)*, pp. 624–639, 2018c.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pp. 10–18. PMLR, 2013.

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*, pp. 1406–1415, 2019.

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.

Yang Shu, Zhangjie Cao, Chenyu Wang, Jianmin Wang, and Mingsheng Long. Open domain generalization with domain-augmented meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9624–9633, 2021.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pp. 776–794. Springer, 2020.

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.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Wenjun Zeng, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. In *IJCAI*, 2021.

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.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

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*, pp. 13834–13844, 2021.

Xingxuan Zhang, Linjun Zhou, Renzhe Xu, Peng Cui, Zheyan Shen, and Haoxin Liu. Domain-irrelevant representation learning for unsupervised domain generalization. In *CVPR*, 2022.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European conference on computer vision*, pp. 561–578. Springer, 2020.

APPENDIX

A PROOF OF PROPOSITION

A.1 PROOF OF PROPOSITION 1

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

Proof. Without loss of generality, we consider two given samples x_i and x_j belonging to the same domain, *i.e.*, $d_i = d_j$. For a given data augmentation set A, we first define the augmented distance between two samples as the maximum distance between their augmented views as

$$d_A\left(x_i^+, x_j^+\right) = \max_{x_i^+\in A(x_i), x_j^+\in A(x_j)} \left\|x_i^+ - x_j^+\right\| \tag{9}$$

Since vanilla augmentations are included in strong augmentations, *i.e.*, $A \subseteq \hat{A}$, we have the inequality as $d_{\hat{A}}\left(x_i^+, x_j^+\right) \geq d_A\left(x_i^+, x_j^+\right)$. Correspondingly, we have the supremum of the distance of two augmented view sets as

$$\mathcal{V}_A \triangleq \sup \rho(A(x_i), A(x_j)) = d_A(x_i^+, x_j^+) \tag{10}$$

$$\mathcal{V}_{\hat{A}} \triangleq \sup \rho(\hat{A}(x_i), \hat{A}(x_j)) = d_{\hat{A}}(x_i^+, x_j^+) \tag{11}$$

Since $d_{\hat{A}}\left(x_i^+, x_j^+\right) \geq d_A\left(x_i^+, x_j^+\right)$, we have $\mathcal{V}_{\hat{A}} \geq \mathcal{V}_A$. Then we define the overlap of two distributions as

$$\phi_{\mathcal{A}} \triangleq \mathrm{Supp}(\mathcal{A}(x_i^+ \mid x_i)) \bigcap \mathrm{Supp}(\mathcal{A}(x_j^+ \mid x_j)) \tag{12}$$

$$\phi_{\hat{\mathcal{A}}} \triangleq \mathrm{Supp}(\hat{\mathcal{A}}(x_i^+ \mid x_i)) \bigcap \mathrm{Supp}(\hat{\mathcal{A}}(x_j^+ \mid x_j)) \tag{13}$$

Since $\mathcal{V}_{\hat{A}} \geq \mathcal{V}_A$, we have $\phi_{\hat{\mathcal{A}}} \subseteq \phi_{\mathcal{A}}$. Then for a given data augmentation set A, we define the minimum product of two augmented samples as

$$e_A\left(x_i^+, x_j^+\right) = \min_{x_i^+\in A(x_i), x_j^+\in A(x_j)} x_i^+ x_j^+ \tag{14}$$

Since vanilla augmentations are included in strong augmentations, *i.e.*, $A \subseteq \hat{A}$, we have the inequality as $e_{\hat{A}}\left(x_i^+, x_j^+\right) \leq e_A\left(x_i^+, x_j^+\right)$. We assume the mean of the product value in the overlap part of distributions as a constant multiple of the minimum product. Then we have $\mathcal{A}(x_i^+|x_i)\mathcal{A}(x_j^+|x_j)$ and $\hat{\mathcal{A}}(x_i^+|x_i)\hat{\mathcal{A}}(x_j^+|x_j)$ as

$$\mathcal{A}(x_i^+|x_i)\mathcal{A}(x_j^+|x_j) = \begin{cases} 0 & x_i^+, x_j^+ \notin \phi_{\mathcal{A}} \\ C \cdot e_A\left(x_i^+, x_j^+\right) & x_i^+, x_j^+ \in \phi_{\mathcal{A}} \end{cases} \tag{15}$$

$$\hat{\mathcal{A}}(x_i^+|x_i)\hat{\mathcal{A}}(x_j^+|x_j) = \begin{cases} 0 & x_i^+, x_j^+ \notin \phi_{\hat{\mathcal{A}}} \\ C \cdot e_{\hat{A}}\left(x_i^+, x_j^+\right) & x_i^+, x_j^+ \in \phi_{\hat{\mathcal{A}}} \end{cases} \tag{16}$$

Since $e_{\hat{A}}\left(x_i^+, x_j^+\right) \leq e_A\left(x_i^+, x_j^+\right)$ and $\phi_{\hat{\mathcal{A}}} \subseteq \phi_{\mathcal{A}}$, $\mathcal{A}(x_i^+|x_i)\mathcal{A}(x_j^+|x_j) \geq \hat{\mathcal{A}}(x_i^+|x_i)\hat{\mathcal{A}}(x_j^+|x_j)$. Consequently, we have

$$\mathbb{E}_{d\sim P_D^S}\mathbb{E}_{x_i,x_j\sim P_d^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(x_j^+|x_j)] \geq \mathbb{E}_{d\sim P_D^S}\mathbb{E}_{x_i,x_j\sim P_d^{S_{\mathrm{UL}}}}[\hat{\mathcal{A}}(x_i^+|x_i)\hat{\mathcal{A}}(x_j^+|x_j)] \tag{17}$$

Thus, we draw the conclusion $\hat{C_\alpha} < C_\alpha$. □

A.2 PROOF OF PROPOSITION 2

Proposition 2. *Dual nearest neighbors (NN) can increase the intra-class connectivity as $\hat{C_\beta} = \mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_y^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(\mathcal{NN}(x_j)^+|\mathcal{NN}(x_j))]$, 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. To calculate the intra-class connectivity, we firstly divide all the samples into two parts: intra-domain intra-class samples and cross-domain intra-class samples. Correspondingly, the intra-class connectivity can be calculated as the sum of cross-domain intra-class connectivity and intra-domain intra-class connectivity.

$$\begin{aligned} C_\beta &= \mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_y^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(x_j^+|x_j)] \\ &= \mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_{y,d_i\neq d_j}^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(x_j^+|x_j)] + \mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_{y,d_i=d_j}^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(x_j^+|x_j)] \end{aligned} \tag{18}$$

Without loss of generality, we consider two given samples x_i and x_j belonging to the different domains with the same semantic class, *i.e.*, $d_i \neq d_j$ and $y_i = y_j$. Give a data augmentation set A, we define the overlap of two distributions as

$$\phi_{d_i\neq d_j} \triangleq \mathrm{Supp}(\mathcal{A}(x_i^+ \mid x_i)) \bigcap \mathrm{Supp}(\mathcal{A}(x_j^+ \mid x_j)) \tag{19}$$

For a given data augmentation set A, transformations cannot overcome significant distribution shifts across different domains, *e.g.*, one can hardly transform a cat in sketch to photo. Thus, we have $\phi_{d_i\neq d_j} \simeq \varnothing$.

While we search for cross domain nearest neighbors (NN) in the latent embedding space as the positive sample. Denote the nearest neighbors of x_j in domain i as $N_i(x_j)$. We have the overlap of distributions as

$$\hat{\phi}_{d_i\neq d_j}^N \triangleq \mathrm{Supp}(\mathcal{A}(x_i^+ \mid x_i)) \bigcap \mathrm{Supp}(\mathcal{A}(N(x_j)^+ \mid N(x_j))) \tag{20}$$

Since $N_i(x_j)$ is in the same domain with x_i with similar semantic information, the augmentation overlap exists. Then, we have $\hat{\phi}_{d_i\neq d_j}^N > \varnothing$ and $\hat{\phi}_{d_i\neq d_j}^N > \phi_{d_i\neq d_j}$. Thus, we have

$$\begin{aligned} &\mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_{y,d_i\neq d_j}^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(\mathcal{NN}(x_j)^+|\mathcal{NN}(x_j))] \\ &\qquad > \mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_{y,d_i\neq d_j}^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(x_j^+|x_j)] \end{aligned} \tag{21}$$

Besides, we consider two given samples x_i and x_j belonging to the same domains with the same semantic class, i.e., $d_i = d_j$ and $y_i = y_j$. Similarly, we have the distribution overlap as $\hat{\phi}_{d_i=d_j}$. Though $\phi_{d_i=d_j} > \varnothing$, the overlap is limited by some intra-domain intra-class semantic variances. Comparably, our intra-domain nearest neighbors (NN) can overcome intra-domain variances with the increased overlap as $\hat{\phi}_{d_i=d_j}^N > \phi_{d_i=d_j}$. Thus, we have

$$\begin{aligned} &\mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_{y,d_i=d_j}^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(\mathcal{NN}(x_j)^+|\mathcal{NN}(x_j))] \\ &\qquad > \mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_{y,d_i=d_j}^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(x_j^+|x_j)] \end{aligned} \tag{22}$$

Combined with Eq. 21, we have

$$\begin{aligned} &\mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_{y,d_i\neq d_j}^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(\mathcal{NN}(x_j)^+|\mathcal{NN}(x_j))] \\ &+ \mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_{y,d_i=d_j}^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(\mathcal{NN}(x_j)^+|\mathcal{NN}(x_j))] \\ &> \mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_{y,d_i\neq d_j}^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(x_j^+|x_j)] + \mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_{y,d_i=d_j}^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(x_j^+|x_j)] \end{aligned} \tag{23}$$

Totally, we draw the conclusion $\hat{C_\beta} > C_\beta$.

For more accurate cross domain NN, since the searched neighbors are more likely to belong to the same semantic class, the searched $N_i'(x_j)$ share more similar semantic information with x_i, which results in a larger augmentation overlap as $\hat{\phi}_{d_i\neq d_j}^{N'} > \hat{\phi}_{d_i\neq d_j}^N$. Thus, we have

$$\begin{aligned} &\mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_{y,d_i\neq d_j}^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(\mathcal{NN}'(x_j)^+|\mathcal{NN}'(x_j))] \\ &\qquad > \mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P_{y,d_i\neq d_j}^{S_{\mathrm{UL}}}}[\mathcal{A}(x_i^+|x_i)\mathcal{A}(\mathcal{NN}(x_j)^+|\mathcal{NN}(x_j))] \end{aligned} \tag{24}$$

For more diverse in-domain NN, since the searched neighbors are more likely to overcome more severe intra-domain variances, the searched $N_i'(x_j)$ can lead to a larger augmentation overlap as $\hat{\phi}^{N'}_{d_i=d_j} > \hat{\phi}^{N}_{d_i=d_j}$. Thus, we have

$$\begin{aligned}&\mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P^{S_{\text{UL}}}_{y,d_i=d_j}}\left[\mathcal{A}(x_i^+|x_i)\mathcal{A}(\mathcal{NN}'(x_j)^+|\mathcal{NN}'(x_j))\right]\\ &\qquad > \mathbb{E}_{y\sim P_Y^S}\mathbb{E}_{x_i,x_j\sim P^{S_{\text{UL}}}_{y,d_i=d_j}}\left[\mathcal{A}(x_i^+|x_i)\mathcal{A}(\mathcal{NN}(x_j)^+|\mathcal{NN}(x_j))\right] \end{aligned} \tag{25}$$

Combined with Eq. 24, we can draw the conclusion that more accurate cross domain NN and more diverse in-domain NN can further increase the intra-class connectivity. □

A.3 Proof of Proposition 3

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

Proof. Denote e_{cr} as the error rate of the cross domain nearest neighbor and e_{in} the error rate of the in-domain nearest neighbor. For simplicity, we assume the error rate of the second nearest neighbor is also e_{cr} and e_{in} for cross domain and in-domain, respectively. We assume when the nearest neighbor is wrong, it is equally likely to match to anyone of the remaining $C-1$ classes, where C is the total number of classes.

Considering a given query z, the error rate of the vanilla cross domain NN is $P_{\text{vanilla}} = e_{cr}$.

For our proposed CD^{2}NN strategy shown in Figure 3, if $\mathcal{R}_1 \neq \varnothing$, *i.e.*, our CD^{2}NN selects the NN in $\mathcal{R}_1$. The selected NN is wrong only if the following two conditions are met: 1) The cross domain NN of z is wrong; 2) The in-domain NN $z_{nn}^{q_{in}}$ of z is right and cross domain NN of $z_{nn}^{q_{in}}$ is wrong **or** the in-domain NN $z_{nn}^{q_{in}}$ of z is wrong and the cross domain NN of $z_{nn}^{q_{in}}$ is not in the same class as z. Thus, the error rate of our CD^{2}NN is

$$\begin{aligned}P^{\mathcal{R}_1}_{\text{CD}^2\text{NN}} &= e_{cr}\cdot\left((1-e_{in})\cdot e_{cr} + e_{in}\cdot(1-e_{cr}+e_{cr}\cdot\frac{C-2}{C-1})\right)\\ &= e_{cr}\cdot\left((1-e_{in})\cdot e_{cr} + e_{in}\cdot(1-e_{cr}\cdot\frac{1}{C-1})\right)\\ &< e_{cr}\cdot\left((1-e_{in})\cdot e_{cr} + e_{in}\right).\end{aligned} \tag{26}$$

Then, we have

$$P^{\mathcal{R}_1}_{\text{CD}^2\text{NN}} < e_{cr}\cdot\left(e_{cr}+e_{in}-e_{cr}\cdot e_{in}\right). \tag{27}$$

Since $0 < e_{cr} < 1$ and $0 < e_{in} < 1$, we have

$$e_{cr}+e_{in}-e_{cr}\cdot e_{in}-1 = (e_{cr}-1)\cdot(1-e_{in}) < 0. \tag{28}$$

Thus, $e_{cr}+e_{in}-e_{cr}\cdot e_{in} < 1$ and $e_{cr}\cdot(e_{cr}+e_{in}-e_{cr}\cdot e_{in}) < e_{cr}$. Since $P_{\text{vanilla}} = e_{cr}$, from Equation equation 27, we have:

$$P^{\mathcal{R}_1}_{\text{CD}^2\text{NN}} < e_{cr}\cdot\left((1-e_{in})\cdot e_{cr}+e_{in}\right) < P_{\text{vanilla}}. \tag{29}$$

As shown in Figure 3, if $\mathcal{R}_1 = \varnothing$ and $\mathcal{R}_2 \neq \varnothing$, *i.e.*, our CD^{2}NN selects the NN in $\mathcal{R}_2$. The selected NN is wrong only if the following two conditions are met: 1) The cross domain NN of z is wrong; 2) The cross domain NN $z_{nn}^{q_{cr}}$ of z is right and in-domain NN of $z_{nn}^{q_{cr}}$ is wrong **or** the cross domain NN $z_{nn}^{q_{cr}}$ of z is wrong and the in-domain NN of $z_{nn}^{q_{cr}}$ is not in the same class as z. Thus, the error rate of our CD^{2}NN is

$$\begin{aligned}P^{\mathcal{R}_2}_{\text{CD}^2\text{NN}} &= e_{cr}\cdot\left((1-e_{cr})\cdot e_{in} + e_{cr}\cdot(1-e_{in}+e_{in}\cdot\frac{C-2}{C-1})\right)\\ &= e_{cr}\cdot\left((1-e_{cr})\cdot e_{in} + e_{cr}\cdot(1-e_{in}\cdot\frac{1}{C-1})\right)\\ &< e_{cr}\cdot\left((1-e_{cr})\cdot e_{in} + e_{cr}\right)\end{aligned} \tag{30}$$

Similarly, we have

$$P^{\mathcal{R}_2}_{\text{CD}^2\text{NN}} < e_{cr}\cdot\left(e_{in}+e_{cr}-e_{cr}\cdot e_{in}\right), \tag{31}$$

Since $0 < e_{cr} < 1$ and $0 < e_{in} < 1$, we have

$$e_{in} + e_{cr} - e_{cr} \cdot e_{in} - 1 = (e_{cr} - 1) \cdot (1 - e_{in}) < 0 \tag{32}$$

Thus, $e_{in} + e_{cr} - e_{cr} \cdot e_{in} < 1$ and $e_{cr} \cdot (e_{in} + e_{cr} - e_{cr} \cdot e_{in}) < e_{cr}$. Since $P_{\text{vanilla}} = e_{cr}$, from Equation equation 31, we have: and

$$P^{\mathcal{R}_2}_{\text{CD}^2\text{NN}} < e_{cr} \cdot ((1 - e_{cr}) \cdot e_{in} + e_{cr}) < P_{\text{vanilla}}. \tag{33}$$

Totally, since $P^{\mathcal{R}_1}_{\text{CD}^2\text{NN}} < P_{\text{vanilla}}$ and $P^{\mathcal{R}_2}_{\text{CD}^2\text{NN}} < P_{\text{vanilla}}$, we have $P_{\text{CD}^2\text{NN}} < P_{\text{vanilla}}$. Thus, we show theoretically that Our proposed CD^{2}NN is more accurate than cross domain nearest neighbor in this specific domain generalization setting. □

B More Visualizations

B.1 Intra-class connectivity of our method.

We add analysis from the connectivity perspective. This experiment is conducted without the in-domain cycle NN to evaluate the performance of CD^{2}NN. Specifically, we train the unsupervised model on PACS with three strategies, *i.e.*, in-domain NN, vanilla cross domain NN and our CD^{2}NN, respectively, and compute the corresponding intra-class connectivity. Fig. 7 shows that our CD^{2}NN can increase the intra-class connectivity as the training proceeds. Vanilla cross domain NN selection strategy suffers the limited intra-class connectivity gain due to many wrong NN matches brought by domain shifts. In-domain NN selection strategy achieves satisfactory intra-class connectivity at the beginning of training by clustering more accurate in-domain neighbors. However, in-domain NN selection strategy fails to overcome distribution shifts across domains and cannot align intra-class samples from different domains, which suffers the degraded intra-class connectivity eventually.

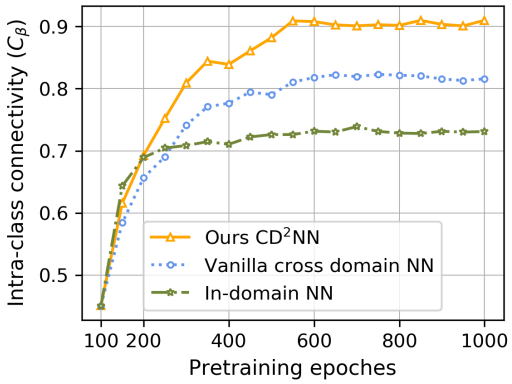


Figure 7: Intra-class connectivity for the model trained with in-domain NN, vanilla cross domain NN and our CD^{2}NN strategy.

B.2 Nearest neighbors searched by our method.

In Fig. 8, we showcase the domain invariant capabilities of the feature representation learned without supervision using our DN^{2}A approach. Each example shows the top-5 nearest neighbors of a random query image (from the PACS dataset) searched in the entire set of images of each of the 4 different PACS domains: *Photo*, *Art painting*, *Cartoon* and *Sketch*. All images are encoded using our self-supervised model trained on three domains (*Art painting*, *Cartoon* and *Sketch*) of PACS dataset.

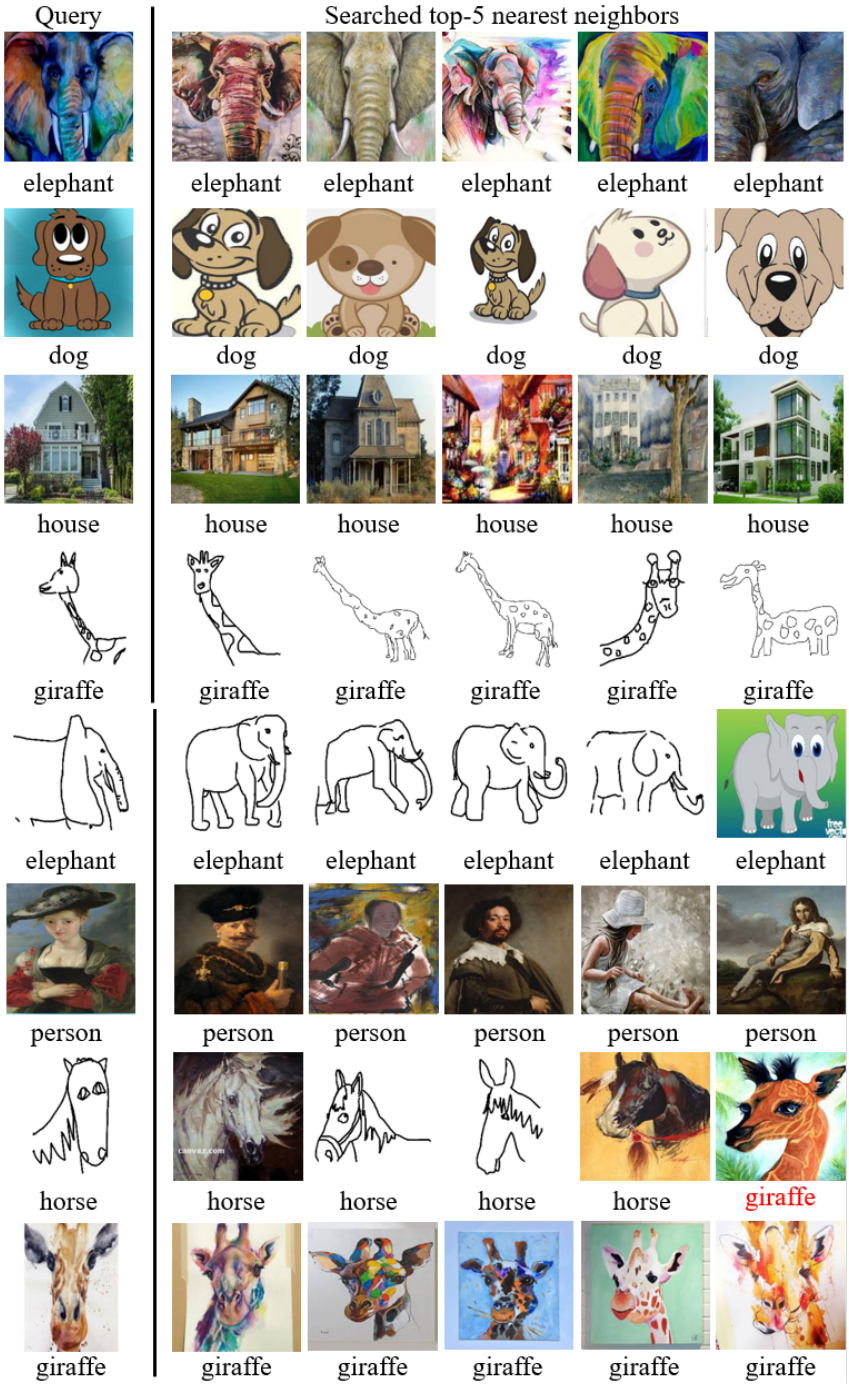


Figure 8: Nearest neighbors searched by our DN^{2}A method.

C More Experiments

C.1 Experiments on the Open Domain Generalization

We follow the open domain generalization setting, *i.e.,* the class split for each domain, in DAML (Shu et al., 2021) to conduct experiments on PACS dataset. We train the model on the unlabeled source data using SimCLR and our DN^{2}A with the same experimental setting in the main text. Besides, we also conduct unsupervised pre-training based on ImageNet initialization (as seen in the bottom half

of Table 9). Then we use the parameters of the trained model as the initialization for the SOTA open domain generalization method DAML (Shu et al., 2021).

Table 9: Results with different initialization methods under the open-domain setting on PACS. † indicates that the unsupervised pre-training is based on ImageNet initialization.

	Art		Sketch		Photo		Cartoon		Avg	
Method	Acc	H-score	Acc	H-score	Acc	H-score	Acc	H-score	Acc	H-score
DAML(random init.) (Shu et al., 2021)	26.20	17.04	24.81	21.04	21.89	16.42	39.92	20.26	28.21	18.69
SimCLR + DAML	35.07	25.91	42.61	31.53	28.33	23.86	48.65	31.66	38.67	28.24
Ours + DAML	**43.86**	**37.30**	**56.42**	**51.09**	**40.67**	**32.37**	**62.81**	**46.54**	**50.94**	**41.83**
DAML(ImageNet init.) (Shu et al., 2021)	54.10	43.02	58.50	56.73	75.69	53.29	73.65	54.47	65.49	51.88
Ours† + DAML	**62.75**	**49.16**	**69.96**	**61.91**	**76.04**	**59.11**	**76.27**	**61.59**	**71.26**	**57.94**

As shown in Table 9, DAML benefits from unsupervised pre-training. Compared with the random initialization and ImageNet initialization, our method can improve DAML for 22.73% and 5.77% average accuracy, respectively. Compared with SimCLR, our DN^{2}A provides a much stronger initialization and boosts the generalization ability of DAML with a 12.27% improvement in average accuracy and a 13.59% improvement in average H-score, demonstrating the effectiveness of our method in the open-set DG setting.

Open-set DG setting assumes different source domains contain private classes and shared classes. With our proposed DN^{2}A, samples in different domains from the shared classes can be aligned. For open-set samples in private classes of some domains (no cross domain neighbors of the same class), our proposed cross domain double-lock NN selection strategy can filter out these untrustworthy noisy neighbors not used as positive samples, *i.e.*, $\mathcal{R} = \varnothing$. With positive samples generated by strong augmentation to suppress the domain information, our method learns class-semantic similarity by separating visually dissimilar images, and eventually separates the private classes from the shared classes. Totally, our method can align shared classes in different source domains, while separating shared classes from private classes. Thus, our proposed method can achieve good performance for the challenging open-set DG setting.

C.2 Experiments on the Few-shot Domain Adaptation

We follow the few shot domain adaptation protocol defined in (Yue et al., 2021) with the same data split, where source domain has a single or three labeled images per-class and the remaining images are provided as unlabeled. Following (Kim et al., 2021; Yue et al., 2021), we use the Resnet-50 pretrained on ImageNet as the backbone, and use 1 or 3 source domain samples per class for the source-only training.

Table 10: Target accuracy (%) on few-shot domain adaptation with source 1-shot and 3-shots labels per class on the Office dataset.

Method	Office: Target Acc. on 1-shot / 3-shots						
	A→D	A→W	D→A	D→W	W→A	W→D	Avg
CDS (Kim et al., 2021)	48.3 / 65.9	49.2 / 65.5	61.4 / 64.4	77.5 / 90.4	57.4 / 64.4	71.5 / 93.0	60.9 / 73.9
PCS (Yue et al., 2021)	60.2 / 78.2	69.8 / 82.9	76.1 / 76.4	90.6 / 94.1	71.2 / 76.3	91.8 / 96.0	76.6 / 84.0
PCS w/o APCU & MIM	47.2 / 71.1	52.7 / 70.6	59.0 / 75.5	76.4 / 90.3	58.5 / 74.1	66.9 / 91.8	60.1 / 78.9
Ours	50.8 / 72.4	54.9 / 71.2	65.1 / 69.7	77.6 / 90.8	62.6 / 71.9	71.5 / 93.1	63.8 / 78.2

As shown in Table 10, our method outperforms CDS by 2.9% average accuracy for 1-shot adaptation. CDS assumes samples of the same class are closer than other samples of different classes across different domains, and directly applies the cross domain matching, which suffers from false matches and introduce the noise to compromise the final performance. As an end-to-end framework proposed for domain adaptation, PCS aims to learn a model that could achieve high accuracy on the target domain. Thus, PCS achieves the highest accuracy with adaptive prototypical classifier learning (consisted of Adaptive Prototype-Classifier Update (APCU) and Mutual Information Maximization (MIM)) for target domain. We also take the result without APCU and MIM from (Yue et al., 2021) for a relatively fair comparison of the cross domain self-supervised learning strategy itself. Our method

outperforms by 3.7% average accuracy for 1-shot adaptation. The proposed instance-prototype cross domain matching (Yue et al., 2021) also suffers from the match noise and degrades the performance.

D Related Works

D.1 Unsupervised Domain Adaptation (UDA).

UDA aims to transfer the knowledge from a labeled source domain to an unlabeled target domain. Haeusser et al. Haeusser et al. (2017) propose the association loss as an discrepancy measure to enforce associations between source and target data for producing statistically domain invariant embeddings. Li et al. Li et al. (2021) propose domain consensus clustering to learn the intrinsic structure of the target domain via encouraging discriminative target clusters. Chen et al. Chen et al. (2022) achieve the feature alignment via mutual nearest neighbors contrast and exploit domain discrimination knowledge by hybrid prototype self-training.

D.2 Self-supervised Learning for UDA.

Recently, self-supervised learning is introduced into domain adaptation. CDS Kim et al. (2021) is proposed to perform self-supervised learning (SSL) not only within a single domain but also across two domains for better domain adaptation performance. PCS Yue et al. (2021) further extends the instance-wise SSL in CDS to prototypical SSL, and proposes a powerful end-to-end framework for domain adaptation. Our DN^{2}A is different from CDS and PCS in the starting point. CDS and PCS are proposed for domain adaptation, where there are two domains and the goal is the target domain alignment. While our method is proposed for domain generalization with multiple domains (more than two domains) to learn domain invariant features. Notice that the cross-domain matching strategy, which is the key component of CDS and PCS for domain alignment, cannot be easily extended to multiple domains. Directly matching each pair of multiple domains may cause the negative transfer, especially for open-set samples. While our method can flexibly find the neighbors in the right domain among multiple domains (also applicable for two domains) for domain-invariant learning. Secondly, CDS assumes that samples of the same class are closer than other samples of different classes across different domains, and uses entropy minimization to implicitly discover and enforce the similarity between cross domain pairs, which suffers from the match noise brought by the domain gap and can be deemed as the vanilla cross domain NN selection counterpart of our method. Though PCS proposes the instance-prototype matching to mitigate the noise, the performance is undermined, especially for open-set samples, where there could be no positive matches from the same class. PCS indiscriminately pushes these negative matches together, while our cross domain double-lock NN (CD^{2}NN) can avoid this situation by excluding the untrustworthy negative matches from training. Thus, our proposed CD^{2}NN strategy is more flexible, effective and robust for cross domain matching, and can be used in CDS and PCS as a superior alternative of their cross domain SSL strategy to boost the performance for domain adaptation tasks. Besides, our CD^{2}NN strategy can extend CDS and PCS to multi-source domain adaptation task, which could be interesting future work.

E Limitations

For the extreme case, where there are no shared classes between any domains, our work fails to use cross domain nearest neighbors for learning the domain invariant feature space. One possible way to address this issue is to use generative-based methods to generate fictitious cross domain samples potentially belonging to the same class as nearest neighbors. Our work can be further improved with adaptive prototypical classifier learning to achieve better performance for domain adaptation task and multi-source domain adaptation task.

F Datasets and Implementation Details

F.1 Datasets.

DomainNet (Peng et al., 2019) is a recently proposed large scale dataset with 0.6 million images of 345 classes distributed on 6 domains, *i.e.*, *Real*, *Clipart*, *Infograph*, *Painting*, *Quickdraw* and *Sketch*.

Table 11: Various augmentations we applied to strongly augment the training images.

Operation	ShearX(Y)	TranslateX(Y)	Rotate	AutoContrast	Identity	Equalize
Mag Range	[-0.3,0.3]	[-0.3,0.3]	[-30,30]	0 or 1	0 or 1	0 or 1
Operation	Solarize	Posterize	Contrast	Color	Brightness	Sharpeness
Mag Range	[0,256]	[4,8]	[0.05,0.95]	[0.05,0.95]	[0.05,0.95]	[0.05,0.95]

We follow the training/testing split released by (Peng et al., 2019) and follow (Chattopadhyay et al., 2020) to partition the training split at a ratio of 9:1 into the training and validation splits for model selection. **PACS** (Li et al., 2017) consists of four domains, *i.e.*, *Photo*, *Art painting*, *Cartoon* and *Sketch*, with diverse image styles. It contains seven classes and 9,991 images totally. We use the original training/validation split provided by (Li et al., 2017).

F.2 Implementation Details.

Specifically, our strong augmentation strategy consists of 14 types of augmentations: ShearX/Y, TranslateX/Y, Rotate, AutoContrast, Identity, Equalize, Solarize, Posterize, Contrast, Color, Brightness, Sharpness. The magnitude of each augmentation is significant enough to produce as strong augmentations as possible. More details of different transformations are listed in Table 11. Specifically, to transform an image, we randomly select 5 augmentations from the above 14 types of transformations, which creates powerful $\hat{A}$ with $\binom{14}{5}$ possible combinations, and apply them to the image sequentially.

The UDG experiments consist of three steps: 1) unsupervised training on the source domains; 2) using a small subset of labeled source domain images to train the unsupervised model (linear probing or fine-tuning); 3) testing the trained model on the target domain, which is unseen during the whole training process.

For unsupervised training, based on SimCLR (Chen et al., 2020), 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 (Harary et al., 2021; Zhang et al., 2022), 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. We set the temperature as $\tau = 0.07$ and warm up epoch as $T = 100$. For **DomainNet**, we train on *Painting*, *Real* and *Sketch* and test on *Clipart*, *Infograph* and *Quickdraw*, and vice versa. For **PACS**, we evaluate our method in the leave-one-domain-out way, *i.e.*, train on three domains and test on the remaining domain.

For *all correlated* setting, we evaluate with linear probing and KNN accuracy. For linear probing, we train a linear classifier with learning rate 30 for 30 epochs and use the source validation set for model selection. Besides, we provide KNN (K=1) accuracy for our method, where we directly use our unsupervised features without any additional training. For *domain correlated* setting, due to category shift, we evaluate the model after finetuning 30 epochs with learning rate $1e^{-3}$, and use the source validation set for model selection.

G Surrogate Metrics for Connectivity.

We propose to define the Overlap Ratio (OR) metric as a surrogate measure for the degree of connectivity. Given an unlabeled dataset S_{UL} with N_{UL} samples, we randomly augment each raw image $x_i \in S_{\text{UL}}$ for C times, and get an augmented set $\tilde{S}_{\text{UL}} = \{x_{ij}, i \in [N_{\text{UL}}], j \in [C]\}$, which is the experimental approximation to the distribution of augmentations $\mathcal{A}(\cdot|x)$. Then, for each $x_{ip} \in \tilde{S}_{\text{UL}}$that is an augmented view of $x_i \in S_{\text{UL}}$, denoting its k-nearest neighbors in $\tilde{S}_{\text{UL}}$ in the embedding space of the encoder f as $N(x_{ip}, \tilde{S}_{\text{UL}} \backslash x_{ip}, k)$, other augmented views from the same domain as $\mathcal{C}_\alpha(x_{ip}) = \{x_{jl}, d_i = d_j, l \in [C]\}$, and other augmented views from the same category as $\mathcal{C}_\beta(x_{ip}) = \{x_{jl}, y_i = y_j, l \in [C]\}$, we can define the intra-domain overlap ratio and intra-class overlap ratio as the ratio of augmented views from the same domain and category in its k-nearest

neighbors, respectively.

$$\mathrm{OR}_{\alpha}\left(x_{ip}\right)=\frac{\#[N(x_{ip},\tilde{S}_{\mathrm{UL}}\backslash x_{ip},k)\cap\mathcal{C}_{\alpha}\left(x_{ip}\right)]}{\#N(x_{ip},\tilde{S}_{\mathrm{UL}}\backslash x_{ip},k)}\in[0,1] \tag{34}$$

$$\mathrm{OR}_{\beta}\left(x_{ip}\right)=\frac{\#[N(x_{ip},\tilde{S}_{\mathrm{UL}}\backslash x_{ip},k)\cap\mathcal{C}_{\beta}\left(x_{ip}\right)]}{\#N(x_{ip},\tilde{S}_{\mathrm{UL}}\backslash x_{ip},k)}\in[0,1] \tag{35}$$

We can define its average as Average Overlap Ratio (AOR) on the whole dataset:

$$\mathrm{AOR}_{\alpha}=\mathbb{E}_{x_{ip}\sim\tilde{S}_{\mathrm{UL}}}\mathrm{OR}_{\alpha}\left(x_{ip}\right),\mathrm{AOR}_{\beta}=\mathbb{E}_{x_{ip}\sim\tilde{S}_{\mathrm{UL}}}\mathrm{OR}_{\beta}\left(x_{ip}\right) \tag{36}$$

Here AOR_{α} and AOR_{β} are used as surrogate metrics for intra-domain and intra-class connectivity, respectively. Specifically, we adopt $C=10$ augmentations for each image and take $k=1$ by default. The encoder f is ResNet-18 trained with 10 epochs for warm up in an unsupervised manner.

H Main Algorithms

Algorithm 1 Cross Domain Double-lock Nearest Neighbor Search

Require: The query embedding z, the in-domain support set of embeddings within a mini-batch from the same domain Q_z^{in} and the cross domain support set from different domains Q_z^{cr}.
Ensure: Cross domain double-lock nearest neighbor of z as $\mathcal{R}(z,Q_z^{cr})$.

Search the in-domain nearest neighbor of z from the in-domain support set Q_z^{in} as $N(z,Q_z^{in},1)=z_{nn}^{q_{in}}$.
Search the top-2 cross domain nearest neighbor of z from the cross domain support set Q_z^{cr} as $N(z,Q_z^{cr},2)$.
Search the top-2 cross domain nearest neighbor of $z_{nn}^{q_{in}}$ from the cross domain support set Q_z^{cr} as $N(z_{nn}^{q_{in}},Q_z^{cr},2)$.
if $N(z,Q_z^{cr},2)\cap N(z_{nn}^{q_{in}},Q_z^{cr},2)\neq\varnothing$ **then**
  return the top-1 ranked neighbor in $N(z,Q_z^{cr},2)\cap N(z_{nn}^{q_{in}},Q_z^{cr},2)$.
else
  Search the top-2 cross domain nearest neighbor of z from the cross domain support set Q_z^{cr} except $z_{nn}^{q_{cr}}$ as $N(z,Q_z^{cr}\backslash z_{nn}^{q_{cr}},2)$.
  Search the top-2 in-domain nearest neighbor of $z_{nn}^{q_{cr}}$ from the cross domain support set Q_z^{cr} except $z_{nn}^{q_{cr}}$ as $N(z_{nn}^{q_{cr}},Q_z^{cr}\backslash z_{nn}^{q_{cr}},2)$.
  if $N(z,Q_z^{cr}\backslash z_{nn}^{q_{cr}},2)\cap N(z_{nn}^{q_{cr}},Q_z^{cr}\backslash z_{nn}^{q_{cr}},2)\neq\varnothing$ **then**
    return the top-1 ranked neighbor in $N(z,Q_z^{cr}\backslash z_{nn}^{q_{cr}},2)\cap N(z_{nn}^{q_{cr}},Q_z^{cr}\backslash z_{nn}^{q_{cr}},2)$.
  else
    return $\varnothing$.
  end if
end if

I The Symbol Table

We add a symbol table as shown in Table 12 to clarify the meaning of the math symbols used in the paper.

Table 12: The meaning of the main symbols defined in the paper.

Symbol	Meaning
X_i, y_i, d_i	image/category label/domain label
$\mathcal{X}$, $\mathcal{Y}$, $\mathcal{D}$	input/category label/domain label space
X, Y, D	random variables defined in $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{D}$.
S, P^S, N^S	dataset/distribution/size of dataset
P_X^S, P_Y^S, P_D^S	marginal distribution of P^S on X, Y, D
$\mathrm{Supp}(\cdot)$	support of a given distribution
S_{UL}, S_{L}	labeled/unlabeled source training data
S_{test}	unseen target testing data
$P^{S_{\mathrm{UL}}}$,$P^{S_{\mathrm{L}}}$,$P^{S_{\text{test}}}$	distribution of S_{UL}, S_{L}, S_{test}
z	embedding for the input x
z_+, z_-	positive/negative embedding for z
$\mathcal{A}(x^+\|x)$	distribution of the augmentations of x
C_α, C_β	intra-domain/intra-class connectivity
Q_z^{in}, Q_z^{cr}	in-domain/cross domain support set for z
$N(z, Q, k)$	k-nearest neighbor of z in Q
$z_{nn}^{q_{in}}$, $z_{nn}^{q_{cr}}$	in-domain/cross domain nearest neighbor of z
$\mathcal{R}(z, Q_z^{cr})$	our proposed CD^{2}NN neighbor of z
$\mathcal{C}(z, Q_z^{in})$	our proposed ICNN neighbor of z
e_{cr}, e_{in}	error rate of cross domain/in-domain NN
$P_{\mathrm{CD^2NN}}$, P_{vanilla}	error rate of our CD^{2}NN/vanilla cross domain NN