InCo: Intermediate Prototype Contrast for Unsupervised Domain Adaptation

Yuntao Du[1]*, Hongtao Luo[1]*, Haiyang Yang[1], Juan Jiang[1], and Chongjun Wang[1]**

State Key Laboratory for Novel Software Technology at Nanjing University, 210023,
Nanjing, China
{duyuntao,mf20330054,hyyang,mf20330037}@smail.nju.edu.cn,
chjwang@nju.edu.cn

Abstract. Unsupervised domain adaptation aims to transfer knowledge from the labeled source domain to the unlabeled target domain. Recently, self-supervised learning (e.g. contrastive learning) has been extended to cross-domain scenarios for reducing domain discrepancy in either instance-to-instance or instance-to-prototype manner. Although achieving remarkable progress, when the domain discrepancy is large, these methods would not perform well as a large shift leads to incorrect initial pseudo labels. To mitigate the performance degradation caused by large domain shifts, we propose to construct multiple intermediate prototypes for each class and perform cross-domain instance-to-prototype based contrastive learning with these constructed intermediate prototypes. Compared with direct cross-domain self-supervised learning, the intermediate prototypes could contain more accurate label information and achieve better performance. Besides, to learn discriminative features and perform domain-level distribution alignment, we perform intra-domain contrastive learning and domain adversarial training. Thus, the model could learn both discriminative and invariant features. Extensive experiments are conducted on three public benchmarks (ImageCLEF, Office-31, and Office-Home), and the results show that the proposed method outperforms baseline methods.

Keywords: Unsupervised domain adaptation · Transfer learning · Contrastive learning · Intermediate prototypes.

1 Introduction

Deep learning has achieved remarkable performance in various computer vision tasks, such as image classification [18,13], semantic segmentation [21], and object detection [12]. Despite achieving remarkable progress, deep neural networks trained on a specific domain often fail to generalize to new domains because of the domain shift problem [28]. Unsupervised domain adaptation (UDA) could

* The first two authors contributed equally.

** Corresponding author

overcome this challenge by transferring knowledge from a fully-labeled source domain to an unlabeled target domain.

Most existing deep domain adaptations fall into two strategies: moment matching and adversarial domain adaptation. The former aims to reduce the domain discrepancy by optimizing the statistical distribution discrepancy, such as Maximum Mean Discrepancy (MMD) distance [22], Joint Maximum Mean Discrepancy (JMMD) distance [24], and Wasserstein distance [32]. The latter reduces the domain discrepancy by the adversarial training across domains where a domain discriminator is introduced to distinguish the source domain from the target domain [10]. As the domain adversarial loss only achieves domain-level alignment and may lead to class mismatch, the following methods focus on class-level alignment [23,3] to achieve better adaptation.

Recently, some works have attempted to bridge the domain gap by extending traditional self-supervised learning (SSL, e.g., contrastive learning (CL)), which is learned from a single domain, to performing SSL across domains [17,40,2,36]. Early methods focus on instance-to-instance contrastive learning [17,2,36]. These methods differ in how to construct positive pairs and negative pairs. For example, CDS [17] proposes a two-stage pipeline (i.e., SSL followed by domain adaptation) and cross-domain instance-based contrastive loss is adopted for learning domain-invariant features across domains. It selects a sample in the other domain as a positive pair and other samples as negative pairs. However, such a method regards each instance as a class, the semantic structure of the data (class information) is not encoded by the learned representations. To overcome this challenge, the following methods focus on semantic aware contrastive learning. TCL [2] and CDCL [36] perform instance-to-instance contrastive learning and they select the samples from the same class of the other domain as positive pairs and the samples from other classes as negative pairs. Besides, different from instance-to-instance manner, PCS [40] proposes prototypical cross-domain self-supervised learning, where cross-domain instance-to-prototype matching is designed to transfer knowledge from source to the target in a more robust manner. The semantic information is encoded in the class prototypes and they regard the corresponding class prototype in the other domain as positive pair and other prototypes as negative pairs. Although achieving remarkable progress, when there is a large shift across domains, these methods would not perform well. As these methods rely on pseudo labels for training, the initial pseudo labels are largely affected by the distribution discrepancy. The larger the shift is, the worse pseudo labels we will get. In such cases, the learned model would be negatively affected by mislabeled positive and negative pairs, leading to poor performance.

In this paper, we would explore how to better perform contrastive learning to bridge the domain gap under a large domain shift. Firstly, we follow the line of instance-to-prototype contrastive learning, as cross-domain instance-to-instance matching is very sensitive to abnormal samples [40]. Secondly, recent progress in UDA [26,7] reveals that intermediate domains across domains could effectively deal with large domain shifts and achieve better performance. These intermediate domain based methods focus on the sample-level intermediate domain. In this

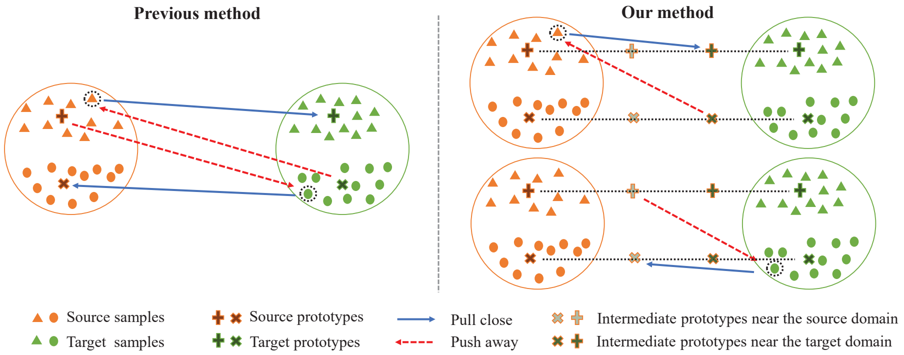


Fig. 1. Illustration of our method. **Left:** previous method performs direct cross-domain instance-to-prototype contrastive learning to bridge domain shift. But it could not perform well when domain shift is large. **Right:** To overcome this problem, we construct multiple intermediate prototypes and perform bidirectional cross-domain instance-to-prototype contrastive learning based on these intermediate prototypes.

paper, we turn to the prototype-level intermediate domain and construct multiple intermediate prototypes for each class to perform contrastive learning. To achieve this, multiple intermediate class prototypes are constructed by a fixed ratio mixup [41] between the source prototypes and the target prototypes [26]. Compare with the sample-level intermediate domain, the intermediate prototypes would be more robust to outliers in the source domain and could contain the sample relations in each domain. Moreover, as shown in Figure 1 and similar to the sample-level intermediate domain, an augmented class prototype close to the source domain has more reliable label information but is less similar to the target domain. By contrast, The class prototype close to the target domain has more relevant information about the target domain, but the label information is less accurate.

To this end, we propose **In**termediate prototype **Co**ntrast (**InCo**), a novel UDA method that constructs multiple intermediate prototypes for performing cross-domain instance-to-prototype contrastive learning. InCo contains three major components to learn semantic, domain-invariant, and discriminative features. As the core component of our method, InCo performs the inter-domain contrastive learning based on intermediate prototypes to mitigate the key challenge of large domain shift. Specially, we construct multiple intermediate class prototypes to bidirectionally apply instance-to-prototype contrastive learning. Before constructing intermediate prototypes, we firstly construct the prototypes in both domains. The source prototypes are computed as the mean representations of each class with true labels, while the target prototypes are computed with pseudo labels. The pseudo labels are obtained by clustering where the class center is initiated by source prototypes. Then, we apply a fixed ratio mixup between the source prototypes and target prototypes to construct intermediate prototypes. In inter-domain contrastive learning, for a given sample (either source domain or target domain), the corresponding class prototype near the other domain is selected as the positive pair and other prototypes are selected as negative pairs. Compared with direct cross-domain matching, the label information of intermediate class prototypes is more accurate as the source domain contains true labels. Thus, it could better deal with the large shift. Besides, similar to previous methods, we also adopt contrastive learning in each domain to learn discriminative features. As the instance-to-prototype contrastive learning, to some content, could be regarded as class-level alignment, we also introduce the domain adversarial loss [10] to further decreases the domain-level distribution discrepancy. Combining these losses together, we could learn both invariant and discriminative features.

We conduct extensive experiments to validate the effectiveness of the proposed method on standard DA benchmarks such as ImageCLEF, Office-31, and Office-Home. We also conduct lots of ablation studies to analyze the proposed method. To sum up, the main contributions of this paper are summarized as follows,

- We propose to construct intermediate prototypes by fixed-ratio mixup to perform contrastive learning for adaptation, which could deal with the large domain shift.
- We follow the instance-to-prototype manner and design bidirectional inter-domain contrastive learning to learn invariant features. Besides, with the intra-domain contrastive learning loss and domain adversarial loss, the model could learn both invariant and discriminative semantic features.
- We conduct extensive experiments on three real-world datasets, the results show the effectiveness of the proposed method.

2 Related Work

2.1 Unsupervised Domain Adaptation

A classical domain adaptation theory [2] indicates that it is crucial to reduce the distribution discrepancy across domains to achieve better adaptation. Based on this theory, many domain adaptation methods have been proposed and they are divided into moment matching and adversarial domain adaptation. The goal of the former is to reduce the statistical distribution discrepancy across domains. The widely used statistical measurements include the first-order moment [22], the second-order moment [33], and other statistical measurements [32]. Adversarial domain adaptation reduces the distribution discrepancy in an adversarial manner [10,4]. DANN [10] introduces a domain discriminator which plays a min-max game with the feature extractor by the domain adversarial loss. MCD [31] introduces two classifiers as a discriminator to play a min-max game with the feature extractor. Considering the practical multi-class problem, MDD [45] proposes a margin-based theory, and a new method based on this theory is proposed. As these methods focus on domain-level alignment, following methods [3] adopt the multi-class discriminator and considers both the domain and class information to achieve class-level alignment.

2.2 Contrastive Learning

Contrastive learning is a promising part in unsupervised learning [1,11,27]. The standard manner of contrastive learning is to learn discriminative representations by pulling the query together with positive pairs and pushing apart from negative pairs. Most methods focus on instance-based methods where each sample is regarded as a class. In these methods, the positive pairs are generated by creating different augmentations of each sample and the negative pairs are randomly chosen from different samples. However, the standard contrastive learning [16] methods have not considered task-specific semantic information. To overcome this problem, supervised contrastive learning has been proposed to leverage category labels to select positive and negative pairs. Furthermore, prototype contrastive learning [19] considered the semantic information in an unsupervised setting by clustering the samples to leverage the semantic information.

2.3 Contrastive Learning for Domain Adaptation

Although achieving remarkable progress, existing contrastive learning approaches can not be directly used in the standard UDA setting as they are performed in a single domain. While some methods have attempted to attend standard contrastive learning to cross-domain scenarios and have achieved satisfactory results. CDS [17] proposes a two-stage pipeline (i.e., SSL followed by domain adaptation), and cross-domain instance-based supervised loss is adopted for learning domain-invariant features across domains. However, the semantic structure of the data (class information) is not encoded by the learned representations. To overcome this challenge, the following methods focus on semantic aware contrastive learning. TCL [2] and CDCL [36] perform instance-to-instance contrastive learning where the positive pairs are selected from the same class in the other domain and the negative pairs are from other classes. Besides, different from instance-to-instance based manner, PCS [40] proposes prototypical cross-domain self-supervised learning, where cross-domain instance-to-prototype matching is designed to transfer knowledge from source to the target in a more robust manner. The semantic information is encoded in the class prototypes and they regard the corresponding class prototype in the other domain as positive pair and other prototypes as negative pairs. Although achieving remarkable progress, these methods would not perform well under large shifts.

3 Method

3.1 Problem Definition and Overall Idea

In UDA, we are given a labeled source domain $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{N_s}$ and an unlabeled target domain $\mathcal{D}_t = \{(\mathbf{x}_j^t)\}_{j=1}^{N_t}$. The source samples and target samples are from different distributions $P_s(\mathbf{x}, y)$ and $P_t(\mathbf{x}, y)$. $\mathcal{D}_s$ and $\mathcal{D}_t$ contain the shared K categories, i.e., $\mathcal{Y}_s = \mathcal{Y}_t = \{1, ...K\}$. The goal of UDA is to learn a generalized model with $\mathcal{D}_s$ and $\mathcal{D}_t$ that could classify the target samples correctly.

In this section, we describe **InCo** in detail. As shown in Figure 2, our model consists of four basic modules, a feature extractor g that maps the samples into feature embeddings, a project head h where the contrastive learning is performed, a domain discriminator D that performs domain adversarial training, and a classifier f that classifies the features into K categories. InCo follows the line of instance-to-prototype manner and uses intermediate prototypes to perform cross-domain contrastive learning so that it could effectively bridge the large discrepancy domains. Specially, we construct intermediate prototypes by fixed ratio mixup and design the bidirectional inter-domain contrastive learning based on these intermediate prototypes. Besides, we also perform contrastive learning within each domain to learn discriminative features. Moreover, inter-domain contrastive learning could achieve class-level alignment, we further introduce domain adversarial training to achieve domain-level alignment. Combing these losses together, the model could learn both invariant and discriminative features. In the next subsections, we introduce each loss in detail.

3.2 Revisit of Contrastive Learning

Contrastive learning [27,11,1] aims to learn discriminative features from unlabeled data in the form of positive/negative pairs by a contrastive loss. We denote the query and key vector as q, k, and k^+ and k^- as the positive and negative key for the query q. The goal of contrastive learning is to learn representations such that the query and positive key vector is as close as possible, meanwhile, the query and the negative key vector is far away from each other. A popular framework to achieve this goal is to formulate the contrastive learning as a 'two-class' classification problem, and the loss is formulated as,

$$\mathcal{L}(q, k^+, k^-) = -log \frac{\exp(q \cdot k^+/T)}{\exp(q \cdot k^+/T) + \sum_{k^-} \exp(q \cdot k^-/T)} \tag{1}$$

Here T is the temperature parameter, and $q \cdot k^+$ denotes the inner product between q and k^+.

3.3 Intra-domain Contrastive Learning

As shown in the above subsection, contrastive learning could learn discriminative features for downstream visual tasks by a contrastive loss. Following that, we perform contrastive learning within each domain to learn discriminative representation for every single domain. As instance-to-instance based CL methods treat each sample as a single class, regardless of the semantic information, we follow the previous method [40] and adopt instance-to-prototype based CL, where the prototypes for all classes are set as the key vectors. By the intra-domain CL loss, representations with intra-class compactness and inter-class discrimination could be learned.

To start with, we define every prototype as the mean representation of each class to convey high-level class information. And we maintain two memory banks $\mathcal{Q}^s$ and $\mathcal{Q}^t$ for source and target prototypes respectively:

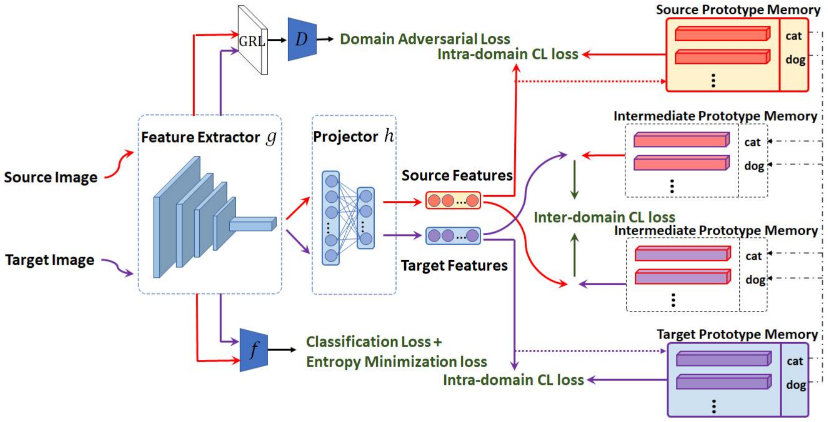


Fig. 2. An overview of InCo. In addition to conventional classification loss on labeled source samples and entropy minimization loss on unlabeled target samples, inter-domain contrastive learning is proposed to bridge the domain gap, where the corresponding prototype near the other domain is selected as the positive pair and other prototypes are selected as negative pairs. We also perform intra-domain contrastive learning within each domain and domain adversarial training, such that the model could learn both invariant and discriminative features.

$$\mathcal{Q}^s = \left[\mu_1^s, \ldots, \mu_K^s\right], \quad \mathcal{Q}^t = \left[\mu_1^t, \ldots, \mu_K^t\right], \tag{2}$$

where μ_k stores the prototype of class k for each domain. After initialization, the memory banks are updated with a momentum m in every batch during training:

$$\mu_k^s \leftarrow m\mu_k^s + (1-m)\frac{1}{|\mathcal{D}_s^k|}\sum_{\mathbf{x}_i^s \in \mathcal{D}_s^k} \mathbf{v}_i^s, \quad \mu_k^t \leftarrow m\mu_k^t + (1-m)\frac{1}{|\mathcal{D}_t^k|}\sum_{\mathbf{x}_i^t \in \mathcal{D}_t^k} \mathbf{v}_i^t \tag{3}$$

where $\mathbf{v}_i = h\left(g\left(\mathbf{x}_i\right)\right)$ is the L_2-normalized feature embedding of $\mathbf{x}_i$ extracted by the feature extractor g and project head h, and $\mathcal{D}_s^k/\mathcal{D}_t^k$ denote the set of source/target samples whose labels/pseudo labels (described in the later subsection) are k in the current mini-batch.

Given a query sample $\mathbf{x}_i$, intra-domain contrastive learning computes the similarity score distribution $\mathbf{P}_i$ over K classes based on the distances to the prototypes, where the k-th element denotes the probability of the sample $\mathbf{x}_i$ belonging to the class k. For the source domain, we have

$$\mathbf{P}_{i,k}^s = \frac{\exp(\mu_k^s \cdot \mathbf{v}_i^s / T)}{\sum_{j=1}^K \exp(\mu_j^s \cdot \mathbf{v}_i^s / T)}, \tag{4}$$

where T is a temperature parameter. Similar operations are performed on target samples and we will get $\mathbf{P}_i^t$. Then we can write intra-domain contrastive loss as:

$$\mathcal{L}_{intra} = \sum_{i=1}^{N_s} \mathcal{L}_{CE}\left(\mathbf{P}_i^s, y_i^s\right) + \sum_{i=1}^{N_t} \mathcal{L}_{CE}\left(\mathbf{P}_i^t, \hat{y}_i^t\right), \tag{5}$$

where $\hat{y}_i^t$ is the pseudo label for target sample $\mathbf{x}_i^t$. As we can see, the intra-domain contrastive loss can push the query feature $\mathbf{v}_i$ close to the prototype indicated by the ground truth label (or pseudo label for target), and keep it away from other prototypes. Thus, we can learn discriminative feature representations for classification in each domain.

3.4 Inter-domain Contrastive Learning

The domain discrepancy across domains posits a unique obstacle for learning effective representations that perform well in the target domain. Recently, some methods have attempted to attend the standard contrastive learning to the cross-domain scenario for bridging domain discrepancy. The main challenge of applying contrasting learning to UDA lies in how to construct positive and negative pairs in a cross-domain scenario. To retain semantic information, some methods select the sample from the same class of the other domain as positive pairs and the samples from other classes as negative pairs according to the true labels or pseudo labels. Besides, some methods adopt cross-domain instance-to-prototype based contrastive learning where the prototype (instead of the samples) of the same class in the other domain is selected as the positive pair and the prototypes of other classes are as the negative pairs. However, both strategies have drawbacks. For the former, the instance-to-instance matching is very sensitive to abnormal samples, especially under domain shift. For the latter, under a large domain shift, direct cross-domain contrastive learning would not perform well as the large domain shift would lead to incorrect initial pseudo labels.

To mitigate these problems, we propose to perform an intermediate domain prototypical contrastive learning. We firstly construct multiple intermediate prototypes by fixed ratio mixup between the source class prototypes and the target prototypes. Then, for the source domain, the intermediate prototypes near the target domain are used to perform cross-domain contrastive learning. The positive pair is the intermediate prototype of the same class close to the target domain and the negative pairs are the intermediate prototypes of the other classes close to the target domain. And the similar strategy is used for the target samples. In this manner, we could not only reduce the domain discrepancy but also prevent the semantic structure of the data. Compared with direct cross-domain matching, the label information of the intermediate prototype is more accurate as the source domain contains true labels and could be less affected by initial pseudo labels.

Specially, we construct a pair of intermediate prototypes, $\{\mu_k^{st}\}_{k=1}^K$ and $\{\mu_k^{ts}\}_{k=1}^K$ using source prototypes and target prototypes with a fixed ratio mixup:

$$\mu_k^{st} = \lambda_{st}\mu_k^s + (1-\lambda_{st})\,\mu_k^t, \quad \mu_k^{ts} = \lambda_{ts}\mu_k^s + (1-\lambda_{ts})\,\mu_k^t \tag{6}$$

where $\lambda_{st} \in (0.5, 1)$ and $\lambda_{ts} \in (0, 0.5)$ are two fixed mixup ratios. We always set $\lambda_{st} + \lambda_{ts} = 1$ to get a pair of domain-symmetric intermediate prototypes. As we can see, the prototypes μ^{st} is close to the source domain and the prototypes μ^{ts} is close to the target domain.

Taking advantage of the intermediate prototypes, we could alleviate the domain shift with instance-to-prototype contrastive learning. The prototypes μ^{st} close to the source domain have more reliable label information because the source prototypes μ^{s} computed with ground truth source label are account for a large proportion. By contrast, the prototypes μ^{ts} close to the target domain have strong target domain relevance but weak label confidence. Thus, we proposed the inter-domain contrastive loss for bidirectional transfer.

Given a query feature $\mathbf{v}_i^s$ in the source domain, and the intermediate prototypes $\{\mu_k^{ts}\}_{k=1}^K$ close to the target domain, inter-domain contrastive loss first computes the similarity distribution $\mathbf{P}_i^{ts}$, which is,

$$\mathbf{P}_{i,k}^{ts} = \frac{\exp(\mu_k^{ts} \cdot \mathbf{v}_i^s / T)}{\sum_{j=1}^{K} \exp(\mu_j^{ts} \cdot \mathbf{v}_i^s / T)} \tag{7}$$

As there are true labels in the source domain, we perform cross-entropy loss on the pair of source instances and target intermediate prototypes to fully use ground truth label information,

$$\mathcal{L}_{ts} = \sum\nolimits_{i=1}^{N_s} \mathcal{L}_{CE}\left(\mathbf{P}_i^{ts}, y_i^s\right) \tag{8}$$

Similarly, we compute $\mathbf{P}_i^{st}$ using the target feature $\mathbf{v}_i^t$ and intermediate prototypes $\{\mu_k^{st}\}_{k=1}^K$ close to the source domain. Then, we perform entropy minimization on the similarity distribution $\mathbf{P}_i^{st}$, which could find the match between the target feature and source intermediate prototypes but rely less on the label information:

$$\mathcal{L}_{st} = -\sum_{i=1}^{N_t} \sum_{k=1}^{K} \mathbf{P}_{i,k}^{st} \log \mathbf{P}_{i,k}^{st} \tag{9}$$

The final inter-domain contrastive loss is:

$$\mathcal{L}_{inter} = \mathcal{L}_{st} + \mathcal{L}_{ts} \tag{10}$$

3.5 Other Losses

Domain adversrial loss. The inter-domain contrastive learning could reduce domain discrepancy and achieve domain alignment. But it only focuses on class-level alignment and does not explicitly reduce the domain-level distribution shift across domains. To deal with this problem, InCo follows the adversarial manner [10], and introduces a domain discriminator D to distinguish the source feature and target feature. While the feature extractor g is trained to confuse the domain discriminator. By this adversarial loss, the feature extractor could learn domain-invariant features. The adversarial object between feature extractor g and domain discriminator D can be written as:

$$\mathcal{L}_{adv} = \mathbb{E}_{\mathbf{x}_i^s \sim \mathcal{D}_s}[\log D(g(\mathbf{x}_i^s))] + \mathbb{E}_{\mathbf{x}_i^t \sim \mathcal{D}_t}[\log(1 - D(g(\mathbf{x}_i^t)))] \tag{11}$$

Classification loss and entropy minimization loss. To capture the source supervised information, the model is trained to minimize the empirical risk on labeled samples as conventional supervised methods. The feature extractor g maps a source sample $\mathbf{x}_i^s$ into the feature. Then, the classifier f would classify the feature into K categories, i.e., $p(y|\mathbf{x}_i^s) = f(g(\mathbf{x}_i^s))$. Then, the cross-entropy loss $\mathcal{L}_{CE}(\cdot, \cdot)$ is adopted to minimize the empirical risk:

$$\mathcal{L}_{cls} = \mathbb{E}_{(\mathbf{x}_i^s, y_i^s) \sim \mathcal{D}_s} \mathcal{L}_{CE}(y_i^s, p(y|\mathbf{x}_i^s)) \tag{12}$$

As there are no labeled samples in the target domain, we adopt the entropy minimization loss to pass through the low-density regions of the target feature space, which is,

$$\mathcal{L}_{ent} = \mathbb{E}_{\mathbf{x}_i^t \sim \mathcal{D}_t} - \sum\nolimits_{k=1}^{K} p_k(y|\mathbf{x}_i^t) \log p_k(y|\mathbf{x}_i^t) \tag{13}$$

where $p_k(y|\mathbf{x}_i^t)$ is the k-th dimension of $p(y|\mathbf{x}_i^t)$ and $p(y|\mathbf{x}_i^t) = f(g(\mathbf{x}_i^t))$ is the prediction of sample $\mathbf{x}_i^t$ by the model. The combined loss is,

$$\mathcal{L}_{cls-ent} = \mathcal{L}_{cls} + \mathcal{L}_{ent} \tag{14}$$

3.6 Overall

Generation of pseudo labels. Since the ground truth labels are not available in the target domain during training, we perform k-means clustering to generate pseudo labels for the target samples. Due to the randomness in clustering, we use class prototypes from the source domain as the initial clustering centers and set the number of clusters as K. In this case, the clustering algorithm can be seen as the distance matching between target features and source prototypes which could better maintain the target data structure and could easily use the cluster label as the target pseudo label $\hat{y}_i^t$.

Training. The InCo learning framework performs intra-domain contrastive loss, inter-domain contrastive loss, domain adversarial loss, and classification loss. Together with the momentum update in the memory bank, the overall learning objective is:

$$\min_{g,h,f} \mathcal{L}_{cls-ent} + \lambda_{intra} \cdot \mathcal{L}_{intra} + \lambda_{inter} \cdot \mathcal{L}_{inter} + \lambda_{adv} \cdot \mathcal{L}_{adv} \tag{15}$$

$$\max_{D} \mathcal{L}_{adv} \tag{16}$$

where λ_{intra}, λ_{inter} and λ_{adv} are hyper-parameters. Following previous method [10], the min-max training procedure in Eq. 15 and 16 is accomplished by applying a Gradient Reversal Layer (GRL). GRL behaves as the identity function during the forward propagation and inverts the gradient sign during the backward propagation, hence driving the parameters to maximize the output loss.

Table 1. Accuracy (%) on the **Office-31** dataset (ResNet-50).

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet-50	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DANN	82.0	96.9	99.1	79.7	68.2	67.4	82.2
MSTN	91.3	98.9	**100.0**	90.4	72.7	65.6	86.5
CDAN+E	94.1	98.6	**100.0**	92.9	71.0	69.3	87.7
DMRL	90.8	99.0	**100.0**	93.4	73.0	71.2	87.9
SymNets	90.8	98.8	**100.0**	93.9	74.6	72.5	88.4
PCS	92.6	96.6	99.4	95.8	76.6	75.8	89.5
GSDA	**95.7**	99.1	**100.0**	94.8	73.5	74.9	89.7
PCT	94.6	98.7	99.9	93.8	77.2	76.0	90.0
InCo	94.0	**99.1**	**100.0**	**95.8**	**77.3**	**77.0**	**90.5**

4 Experiments

4.1 Datasets

We evaluate InCo on three common benchmarks based on previous works [22,23]. **Office-31**[1] is a classical real-world dataset for UDA. It has 4110 images with 31 classes shared with three domains: Amazon (**A**), Webcam (**W**), and DSLR (**D**). In this dataset, six adaptation tasks are constructed. **ImageCLEF**[2] is composed of three domain with 12 classes: Caltech-256 (**C**), ImageNet ILSVRC 2012 (**I**), and Pascal VOC 2012 (**P**). **Office-Home**[3] is a more difficult dataset, which consists of four domains: Artistic (**Ar**), Clipart (**Cl**), Product (**Pr**), and Real-World (**Rw**), containing 15500 images with 65 classes.

4.2 Setup

We use PyTorch to implement the proposed method. We use ResNet-50 [13] pre-trained on ImageNet [30] as the backbones for all datasets. To enable a fair comparison with the existing method [40], we remove the last FC layer in ResNet and implement a projection head h with the default nonlinear projection and an additional hidden layer activated by ReLU as same as SimCLR [1]. The output dimension of h is 512 and L2-normalizing is performed on the output features. Following DANN [10], we use the same architecture for the domain discriminator D and the classifier f. We use SGD with a momentum of 0.9 and weight decay $5e^{-4}$ to train the InCo for all the experiments. The initial learning rate is 0.001 for the pre-trained feature extractor and 0.01 for other modules. Besides, we split large batch size into small parts, and use gradient accumulation in Pytorch which could backward gradient after multiple forward iterations to achieve the same

[1] https://www.hemanthdv.org/officeHomeDataset.html
[2] https://www.imageclef.org/2014/adaptation
[3] https://www.hemanthdv.org/officeHomeDataset.html

Table 2. Accuracies (%) on the **ImageCLEF** dataset (ResNet-50).

Method	I → P	P → I	I → C	C → I	C → P	P → C	Avg
ResNet-50	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DAN	74.5	82.2	92.8	86.3	69.2	89.8	82.5
DANN	75.0	86.0	96.2	87.0	74.3	91.5	85.0
MADA	75.0	87.9	96.0	88.8	75.2	92.2	85.8
iCAN	79.5	89.7	94.7	89.9	78.5	92.0	87.4
CDAN	77.7	90.7	97.7	91.3	74.2	94.3	87.7
A^2LP	79.6	92.7	96.7	92.5	78.9	96.0	89.4
CGDM	78.7	93.3	97.5	92.7	79.2	95.7	89.5
ETD	**81.0**	91.7	**97.9**	93.3	79.5	95.0	89.7
SymNets	80.2	93.6	97.0	93.4	78.7	**96.4**	89.9
InCo	79.5	**94.5**	96.5	**94.8**	**80.3**	96.2	**90.3**

effect as large batch size but obtain smoother prototypes with multiple update operations. Specially, we use a batch size of 16 for Office-31 and ImageCLEF and backward loss after four forward iterations. For Office-Home, we use a batch size of 32 and backward after two forward iterations. The temperature parameter T is fixed to 0.3, 0.5, and 0.1 for Office-31, ImageCLEF, and Office-Home. The momentum m is 0.9 for all datasets. The hyper-parameters λ_{intra}, λ_{inter}, and λ_{adv} are all set to 1.0 which is selected from $\{0.5, 1.0, 2.0\}$. We set mixup ration $\lambda_{st} = 0.8$ and $\lambda_{ts} = 0.2$ for all datasets with λ_{st} selected from $\{0.7, 0.8, 0.9\}$.

4.3 Baselines

We compare with InCo with four kinds of baselines:

- **ResNet-50**. This baseline refers to the source-only method, where only the source samples are used for training.
- **Moment matching and adversarial-based methods**, including DAN [22], DANN [10], MADA [29], MCD [31], CDAN [23], MSTN [38], iCAN [42], MDD [45], SymNets [44], GSDA [14], DMRL [37], ETD [20], A^2LP [43], MDD+IA [15], BNM [5], BDG [39], GVB [6], SRDC [34], and CGDM [9].
- **Prototype-based methods**, including PCT [35].
- **Contrastive learning based methods**, including PCS [40].

4.4 Results

Table 1 displays the performances of various models on Office-31. Generally, InCo outperforms the baseline method in most transfer tasks (5/6). It is noticed that InCo is especially effective on harder transfer tasks, e.g. W→A and A→D, where the two domains are substantially different. Moreover, compared with PCS, which adopts direct cross-domain instance-to-prototype contrastive learning, InCo

Table 3. Classification accuracies (%) on the **Office-Home** dataset (ResNet-50).

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg.
ResNet-50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
MCD	48.9	68.3	74.6	61.3	67.6	68.8	57.0	47.1	75.1	69.1	52.2	79.6	64.1
CDAN	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
BNM	52.3	73.9	80.0	63.3	72.9	74.9	61.7	49.5	79.7	70.5	53.6	82.2	67.9
MDD	54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
BDG	51.5	73.4	78.7	65.3	71.5	73.7	65.1	49.7	81.1	74.6	55.1	84.8	68.7
MDD+IA	56.2	77.9	79.2	64.4	73.1	74.4	64.2	54.2	79.9	71.2	58.1	83.1	69.5
GVB	57.0	74.7	79.8	64.6	74.1	74.6	65.2	55.1	81.0	74.6	59.7	84.3	70.4
SRDC	52.3	76.3	81.0	**69.5**	76.2	78.0	**68.7**	53.8	81.7	**76.3**	57.1	85.0	71.3
PCT	57.1	78.3	81.4	67.6	77.0	76.5	68.0	55.0	81.3	74.7	60.0	85.3	71.8
InCo	**59.2**	**78.6**	**82.5**	67.1	**79.8**	**79.8**	67.3	55.4	**82.7**	74.6	59.3	84.8	**72.6**

gets an improvement of 1%. This verifies that the intermediate prototype based contrastive learning method is a legitimate solution in the context of domain adaptation under large domain shifts.

Table 2 illustrates the performance comparisons on the six adaption directions of ImageCLEF. InCo again demonstrates strong superiority over its competitors. Particularly, InCo offers a significant performance boost on tasks C→P, P→I, and C→I. Compared with other moment matching and adversarial domain adaptation methods, InCo achieves better performance and the results show that contrastive learning based methods could learn invariant features and achieve domain alignment. Besides, by intra-domain contrastive learning, the model could learn more discriminative features, leading to better performance.

Table 3 reports the classification accuracy of twelve transfer tasks on the Office-Home dataset. We can see that InCo gets the best accuracy in the six categories and obtains comparable results in others. Compared with PCT which is a prototype based method, InCo obtains better results combined with an intermediate prototype based contrastive learning and the improvement is 0.8%. Moreover, Office-Home is a more challenging dataset than the other two datasets, and we get better improvement in this dataset, which shows that InCo could deal with large domain shifts.

4.5 Insight Analysis

Analysis of intermediate prototypes. To better understand the role of intermediate prototypes, we conduct lots of ablation studies to analyze them. We compare InCo with the following variants: 1) **No intermediate prototypes**, where we do not construct any intermediate prototypes and perform direct cross-domain instance-to-prototype contrastive learning. 2) **Only one intermediate prototype for each class**, where only one intermediate prototype is constructed

Table 4. Analysis of intermediate prototypes on **Office-31** dataset.

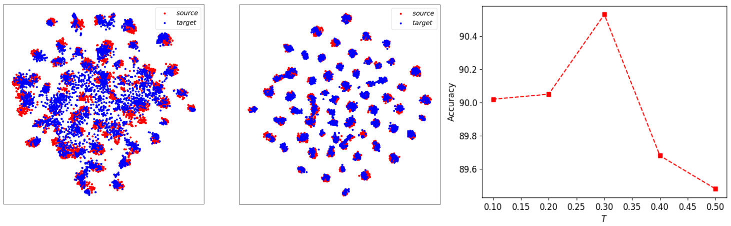
Settings	Average
No intermediate prototypes	89.41
One intermediate prototype (0.2)	89.31
One intermediate prototype (0.5)	89.59
One intermediate prototype (0.8)	89.76
Two intermediate prototypes (0.1+0.9)	89.94
Two intermediate prototypes (0.3+0.7)	89.75
Two intermediate prototypes (0.2+0.8, ours)	**90.53**

for each class and it is used to perform cross-domain instance-to-prototype contrastive learning for samples from both domains. 3) **Two intermediate prototypes for each class**, where two intermediate prototypes are constructed for each class but with a different mixup ratio ($\lambda_{st} = 0.9$,$\lambda_{st} = 0.8$, and $\lambda_{st} = 0.7$). The results are shown in Table 4. As we can see, in most cases, intermediate prototype based contrastive learning methods outperform direct cross-domain contrastive learning methods as the intermediate prototype are more accurate. Besides, two intermediate prototypes could achieve better performance than that of one intermediate prototype for each class ($\lambda_{st} = 0.8$), as two prototypes contain complementary information and could better bridge two domains. Moreover, we obverse that InCo works well with different mixup ratio, and we experimentally find that $\lambda_{st} = 0.8$ and $\lambda_{ts} = 0.2$ is the best value.

Ablation study of losses. In this subsection, we investigate the influence of each component on the overall objective defined in Eq. 15. The results are shown in Table 5. Only adopting the classification loss $\mathcal{L}_{cls}$ and the entropy loss $\mathcal{L}_{ent}$ gets the worst accuracy. After adding domain adversarial loss $\mathcal{L}_{adv}$ to achieve domain-level alignment, the performance is improved to 87.32%. Then, the intra-domain contrastive learning loss $\mathcal{L}_{intra}$ is added to learn discriminative features, the accuracy is improved by 1.9%. Lastly, combining the cross-domain contrastive learning loss $\mathcal{L}_{inter}$, InCo achieves the best performance.

Table 5. Ablation study of losses on **Office-31** dataset.

$\mathcal{L}_{cls-ent}$	$\mathcal{L}_{adv}$	$\mathcal{L}_{intra}$	$\mathcal{L}_{inter}$	Average
√				78.53
√	√			87.32
√	√	√		89.27
√	√	√	√	**90.53**



(a) t-SNE by source-only

(b) t-SNE by InCo

(c) Sensitivity of T

Fig. 3. Visualization of representations learned by source-only model and InCo as well as the parameter sensitivity of T.

Feature visualization. Figure 3(a) and 3(b) show the t-SNE [25] visualization of the features from both domains for task Cl→Rw (65 classes) before (source-only) and after alignment, respectively. Before alignment, there exists a large distribution shift between the source domain and the target domain. While after alignment the domain shift is reduced and the features of target samples have become discriminative. Thus, the samples can be easily classified by the classifier.

Parameter sensitivity of *T*. We perform parameter sensitivity of the temperature parameter T on Office-31. When $T \leq 1$, the model would sharpen the similarity score in contrastive learning to avoid ambiguous predictions, thus, we set $T \leq 1$, and the results under different values are shown in Figure 3(c). As we can see, the performance raises firstly and then drops, as a smaller value would be overconfident in the predictions and a larger value would be less confident. We experimentally find that $T = 0.3$ is the best value.

5 Conclusion

In this paper, we propose a novel UDA method InCo, which performs instance-to-prototype contrastive learning based on intermediate prototypes to deal with large domain shifts. The intermediate prototypes are constructed with a fixed ratio mixup between the source prototypes and target prototypes. Compared with direct cross-domain instance-to-prototype contrastive learning, the intermediate prototypes are more accurate and could mitigate the problem of incorrect initial pseudo labels. Together with intra-domain contrastive learning and domain adversarial training, the model could learn both invariant and discriminative semantic features. The results of three real-world datasets show the effectiveness of the proposed method. In the future, we would like to explore more difficult scenarios such as source-free domain adaptation [8].

6 Acknowledgement

This paper is supported by the National Key Research and Development Program of China (Grant No. 2018YFB1403400), the National Natural Science Foundation of China (Grant No. 61876080), the Key Research and Development Program of Jiangsu(Grant No. BE2019105), the Collaborative Innovation Center of Novel Software Technology and Industrialization at Nanjing University.

References

1. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. ICML (2020)
2. Chen, Y., Pan, Y., Wang, Y., Yao, T., Tian, X., Mei, T.: Transferrable contrastive learning for visual domain adaptation. Proceedings of the 29th ACM International Conference on Multimedia (2021)
3. Cicek, S., Soatto, S.: Unsupervised domain adaptation via regularized conditional alignment. ICCV pp. 1416–1425 (2019)
4. Cui, F., Chen, Y., Du, Y., Cao, Y., Wang, C.: Joint feature and labeling function adaptation for unsupervised domain adaptation. In: PAKDD (2022)
5. Cui, S., Wang, S., Zhuo, J., Li, L., Huang, Q., Tian, Q.: Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. CVPR pp. 3940–3949 (2020)
6. Cui, S., Wang, S., Zhuo, J., Su, C., Huang, Q., Tian, Q.: Gradually vanishing bridge for adversarial domain adaptation. CVPR pp. 12452–12461 (2020)
7. Dai, Y., Liu, J., Sun, Y., Tong, Z., Zhang, C., yu Duan, L.: Idm: An intermediate domain module for domain adaptive person re-id. ICCV pp. 11844–11854 (2021)
8. Du, Y., Yang, H., Chen, M., Jiang, J., Luo, H., Wang, C.: Generation, augmentation, and alignment: A pseudo-source domain based method for source-free domain adaptation. ArXiv **abs/2109.04015** (2021)
9. Du, Z., Li, J., Su, H., Zhu, L., Lu, K.: Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. CVPR pp. 3936–3945 (2021)
10. Ganin, Y., Ustinova, E., et al.: Domain-adversarial training of neural networks. JMLR (2016)
11. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.B.: Momentum contrast for unsupervised visual representation learning. CVPR pp. 9726–9735 (2020)
12. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask r-cnn. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**, 386–397 (2020)
13. He, K., Zhang, X., et al.: Deep residual learning for image recognition. CVPR (2016)
14. Hu, L., Kan, M., Shan, S., Chen, X.: Unsupervised domain adaptation with hierarchical gradient synchronization. CVPR pp. 4042–4051 (2020)
15. Jiang, X., Lao, Q., Matwin, S., Havaei, M.: Implicit class-conditioned domain alignment for unsupervised domain adaptation. ICML (2020)
16. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. NeurIPS (2020)
17. Kim, D., Saito, K., Oh, T.H., Plummer, B.A., Sclaroff, S., Saenko, K.: Cds: Cross-domain self-supervised pre-training. ICCV pp. 9103–9112 (2021)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM **60**, 84 – 90 (2012)
19. Li, J., Zhou, P., Xiong, C., Socher, R., Hoi, S.C.H.: Prototypical contrastive learning of unsupervised representations. ICLR (2021)
20. Li, M., Zhai, Y., Luo, Y.W., Ge, P., Ren, C.X.: Enhanced transport distance for unsupervised domain adaptation. CVPR pp. 13933–13941 (2020)

21. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
22. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. ICML (2015)
23. Long, M., Cao, Z., et al.: Conditional adversarial domain adaptation. In: NeruIPS (2018)
24. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: ICML (2017)
25. van der Maaten, L., Hinton, G.E.: Visualizing data using t-sne. JMLR **9**, 2579–2605 (2008)
26. Na, J., Jung, H., Chang, H., Hwang, W.: Fixbi: Bridging domain spaces for unsupervised domain adaptation. CVPR pp. 1094–1103 (2021)
27. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. ArXiv **abs/1807.03748** (2018)
28. Pan, S.J., Yang, Q.: A survey on transfer learning. TKDE **22**, 1345–1359 (2010)
29. Pei, Z., Cao, Z., Long, M., Wang, J.: Multi-adversarial domain adaptation. In: AAAI (2018)
30. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. IJCV **115**, 211–252 (2015)
31. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. CVPR pp. 3723–3732 (2018)
32. Shen, J., Qu, Y., Zhang, W., Yu, Y.: Wasserstein distance guided representation learning for domain adaptation. In: AAAI (2018)
33. Sun, B., Feng, J., Saenko, K.: Return of frustratingly easy domain adaptation. In: AAAI (2016)
34. Tang, H., Chen, K., Jia, K.: Unsupervised domain adaptation via structurally regularized deep clustering. CVPR pp. 8722–8732 (2020)
35. Tanwisuth, K., Fan, X., Zheng, H., Zhang, S., Zhang, H., Chen, B., Zhou, M.: A prototype-oriented framework for unsupervised domain adaptation. NeurIPS (2021)
36. Wang, R., Wu, Z., Weng, Z., Chen, J., Qi, G.J., Jiang, Y.G.: Cross-domain contrastive learning for unsupervised domain adaptation. ArXiv **abs/2106.05528** (2022)
37. Wu, Y., Inkpen, D., El-Roby, A.: Dual mixup regularized learning for adversarial domain adaptation. ECCV (2020)
38. Xie, S., Zheng, Z., Chen, L., Chen, C.: Learning semantic representations for unsupervised domain adaptation. In: ICML (2018)
39. Yang, G., Xia, H., Ding, M., Ding, Z.: Bi-directional generation for unsupervised domain adaptation. In: AAAI (2020)
40. Yue, X., Zheng, Z., Zhang, S., Gao, Y., Darrell, T., Keutzer, K., Vincentelli, A.S.: Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. CVPR pp. 13829–13839 (2021)
41. Zhang, H., Cissé, M., Dauphin, Y., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. ICLR (2018)
42. Zhang, W., Ouyang, W., Li, W., Xu, D.: Collaborative and adversarial network for unsupervised domain adaptation. CVPR pp. 3801–3809 (2018)
43. Zhang, Y., Deng, B., Jia, K., Zhang, L.: Label propagation with augmented anchors: A simple semi-supervised learning baseline for unsupervised domain adaptation. ECCV (2020)
44. Zhang, Y., Tang, H., Jia, K., Tan, M.: Domain-symmetric networks for adversarial domain adaptation. CVPR pp. 5026–5035 (2019)

45. Zhang, Y., Liu, T., Long, M., Jordan, M.I.: Bridging theory and algorithm for domain adaptation. In: ICML (2019)