Probabilistic Contrastive Learning for Domain Adaptation

Junjie Li[1,2] , **Yixin Zhang**[2] , **Zilei Wang**[2∗] , **Saihui Hou**[3] , **Keyu Tu**[2] , **Man Zhang**[1]
[1]Beijing University of Posts and Telecommunications
[2]University of Science and Technology of China
[3]Beijing Normal University
hnljj93@gmail.com,zhyx12@mail.ustc.edu.cn, zlwang@ustc.edu.cn, housaihui@bnu.edu.cn,
tky2017ustc_dx@mail.ustc.edu.cn, zhangman@bupt.edu.cn

Abstract

Contrastive learning has shown impressive success in enhancing feature discriminability for various visual tasks in a self-supervised manner, but the standard contrastive paradigm (features+ℓ_2 normalization) has limited benefits when applied in domain adaptation. We find that this is mainly because the class weights (weights of the final fully connected layer) are ignored in the domain adaptation optimization process, which makes it difficult for features to cluster around the corresponding class weights. To solve this problem, we propose the *simple but powerful* Probabilistic Contrastive Learning (PCL), which moves beyond the standard paradigm by removing ℓ_2 normalization and replacing the features with probabilities. PCL can guide the probability distribution towards a one-hot configuration, thus minimizing the discrepancy between features and class weights. We conduct extensive experiments to validate the effectiveness of PCL and observe consistent performance gains on five tasks, i.e., Unsupervised/Semi-Supervised Domain Adaptation (UDA/SSDA), Semi-Supervised Learning (SSL), UDA Detection and Semantic Segmentation. Notably, for UDA Semantic Segmentation on SYNTHIA, PCL surpasses the sophisticated CPSL-D by $>2\%$ in terms of mean IoU with a much lower training cost (PCL: 1*3090, 5 days v.s. CPSL-D: 4*V100, 11 days). Code is available at https://github.com/ljjcoder/Probabilistic-Contrastive-Learning.

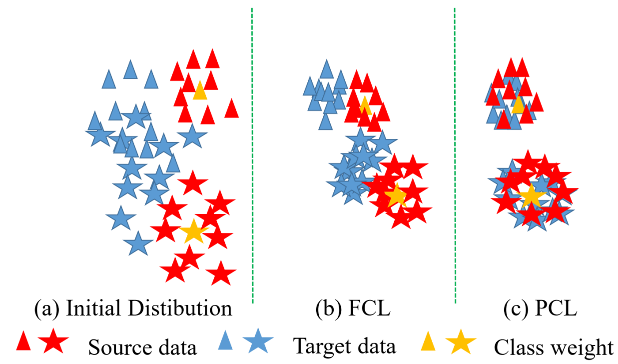


Figure 1: Feature Contrastive Learning (**FSL**) *v.s.*Probabilistic Contrastive Learning (**PCL**). With PCL, the features on target domain can be clustered around the corresponding class weights.

1 Introduction

Deep learning models are usually trained on a specific dataset (source domain) and perform well on similar dataset. However, when these models are applied to data from a different domain (target domain), their performance often degrades significantly. As illustrated in Figure 1(a), this is mainly because there exists domain shift or dataset bias between the source domain and the target domain. Domain adaptation [Yan *et al.*, 2017; Na *et al.*, 2021] offers a solution to this problem by allowing a model trained on a labeled source domain to adapt to an unlabeled or sparsely labeled target domain.

For many visual tasks, the feature discriminability is the basis to obtain satisfying performance. However, in domain adaptation, the learned features for each class on target domain are usually diffuse rather than discriminative as illustrated in Figure 1(a) since target domain lacks the ground-truth labels. Fortunately, contrastive learning is proposed to learn semantically similar features in a self-supervised manner [Chen *et al.*, 2020; Khosla *et al.*, 2020]. Inspired by its great success for representation learning, we hope to perform the standard contrastive learning (features+ℓ_2 normalization) to assist feature extraction on unlabeled target domain. However, we find that naively applying the standard contrastive learning in domain adaptation only brings very limited improvement (*e.g.*, $64.3\% \rightarrow 64.5\%$ as shown in Figure 2). A natural question arises and motivates this work: *Why does contrastive learning perform poorly in domain adaptation?* In the following, we first analyze the possible reasons and then propose a simple but powerful solution to this question.

For recognition tasks to achieve good performance, the learned features are not only required to be discriminative themselves, but also should be close to the class weights (*i.e.*, the weights of the last fully connected layer). However, standard contrastive learning usually typically utilizes features before the classifier to calculate the loss (here we

∗Corresponding Author

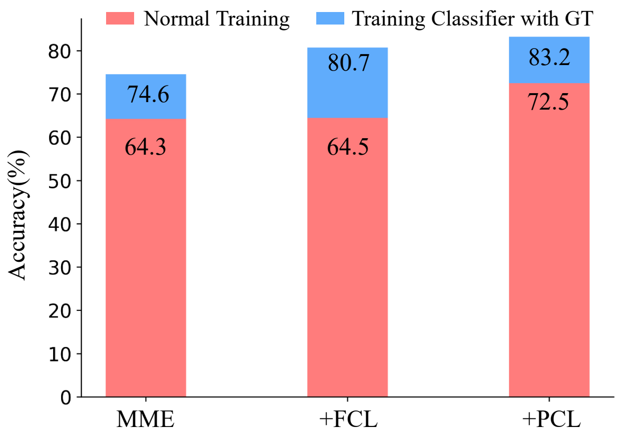


Figure 2: An explorative study under the SSDA setting on DomainNet (R→S) with 3-shot and ResNet34. We use MME as a baseline model.

term this method as Feature Contrastive Learning (FCL)). This approach does not involve the crucial class weights during optimization. While FCL can significantly enhance feature discriminability in the target domain, the issue of domain shift may still cause the features to deviate from the corresponding class weights learned from source data, as depicted in Figure 1(b). We validate this point using an experimental study. Specifically, we freeze the feature extractors and train a classifier borrowing the ground-truth labels on target domain (not available in the normal training). As shown in Figure 2, the accuracy of FCL has increased by 6.1% ($74.6\% \rightarrow 80.7\%$), but the actual accuracy has only increased by 0.2% ($64.3\% \rightarrow 64.5\%$). These findings lead us to propose that *the deviation between features and class weights is the main reason that causes the poor performance of feature-based contrastive learning on domain adaptation*.

To deal with the deviation, we must introduce class weights information. Therefore, a naïve idea is to use the logits (*i.e.*, the output of the last fully connected layer) to calculate the contrastive loss. However, we experimentally find that simply introducing class weights information without explicit constraints cannot effectively alleviate the deviation problem.

This raises an important question: how can we develop a new type of contrastive loss that effectively alleviates the deviation between features and class weights? The key point is to dig out what kind of signal can effectively indicate that a feature vector is close to its corresponding class weights. For convenience, we define a set of class weights as $W = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_C)$, a feature vector as $\mathbf{f}_i$, and its classification probability as $\mathbf{p}_i$. Then the classification probability $p_{i,c}$ of c-th ($c \in \{1, 2, \ldots, C\}$) class for feature $\mathbf{f}_i$ is

$$p_{i,c} = \frac{\exp(\mathbf{w}_c^\top \mathbf{f}_i)}{\sum_{j \neq c} \exp(\mathbf{w}_j^\top \mathbf{f}_i) + \exp(\mathbf{w}_c^\top \mathbf{f}_i)}.$$

To ensure that feature vector $\mathbf{f}_i$ is closely aligned with its corresponding class weight $\mathbf{w}_c$, such that $\mathbf{w}_c^\top \mathbf{f}_i$ is large, $p_{i,c}$ will be close to 1 while $\{p_{i,j}\}_{j \neq c}$ approaches 0. This alignment suggests that as $\mathbf{f}_i$ gets closer to its corresponding class weights, the probability vector $\mathbf{p}_i$ will increasingly resemble a one-hot vector.

To leverage this observation, we propose a novel contrastive learning framework, Probabilistic Contrastive Learning (PCL). PCL is different from standard methods by substituting features with probabilities and removing the ℓ_2 normalization. These two straightforward yet impactful modifications enable PCL to impose a constraint on the probability vectors, guiding them towards a one-hot vector. This approach significantly mitigates deviation issues. PCL is proposed as a simple but powerful method on domain adaptation, which is easily adaptable across various tasks and seamlessly integrable with different methodologies. Remarkably, PCL surpasses many complex alternatives, such as the meta-optimization in MetaAlign [Wei *et al.*, 2021] and prototypical+triplet loss in ECACL-P [Li *et al.*, 2021c].

Our main contributions are summarized as follows:

1) To the best of our knowledge, this is the first work to clearly point out that the problem of feature deviation from class weights is a core reason for the poor performance of standard FCL in domain adaptation tasks.
2) Based on our analysis, we propose a new self-supervised paradigm called Probabilistic Contrastive Learning (PCL) for domain adaptation, which is simple in implementation and powerful in the generalization to different settings and various methods.
3) Extensive experiments demonstrate that PCL can bring consistent performance improvements on different settings and various methods for domain adaptation.

2 Related Work

2.1 Contrastive Representation Learning

Contrastive learning is a mainstream representation learning method, which aims to learn a compact and transferable feature. Currently, extensive works [Chen *et al.*, 2020; Dwibedi *et al.*, 2021; Khosla *et al.*, 2020; Khosla *et al.*, 2020] have demonstrated the effectiveness of this technique in a variety of vision tasks. Some of these works consider conceiving elaborate model architectures to improve performance, such as memory bank [Khosla *et al.*, 2020] or projection head [Chen *et al.*, 2020]. However, these methods usually ignore class information, leading to the false negative problem. Therefore, another part of the work focuses on how to select samples to alleviate the problem of false negative samples. For example, SFCL [Khosla *et al.*, 2020] uses label information to eliminate wrong negative samples, effectively improving the performance. TCL [Singh *et al.*, 2021] extends this idea from instance-level to group-level and proposes group constrastive loss for semi-supervised action detection tasks. Different from previous methods, we argue that in domain adaptation tasks, the core bottleneck of contrastive learning is the deviation of features from class weights. Based on this perspective, we design a simple yet efficient PCL, which does not have to rely on the techniques mentioned above, such as carefully designed positive and negative sample selection strategies and memory banks.

It is worth noting that, TCL enhances traditional architecture by extending the projection head to the classifier and softmax, the design that bears a resemblance to the form of PCL.

However, in principle, PCL and TCL are still fundamentally different. Specifically, TCL is aimed at enhancing the selection strategy of samples, rather than addressing the issue of deviation between features and class weights. More importantly, it does not clarify the importance of probability, or even mention it.[1] It merely treats probability as a special feature, and naturally preserves ℓ_2 normalization, that is, PCL-ℓ_2. Therefore, whether in form or principle, TCL (PCL-ℓ_2) is still a standard contrastive paradigm (features+ℓ_2 normalization). Experiments show that TCL (PCL-ℓ_2) are obviously inferior to PCL and we will discuss it in Sec. 4.2.

2.2 Domain Adaptation

Domain adaptation mostly focuses on the recognition field (*e.g.*, image classification, object detection, semantic segmentation) and aims to transfer the knowledge from a labeled source domain to an unlabeled target domain. The literature can be roughly categorized into two categories.

The first category is to use domain alignment and domain invariant feature learning. For example, [Yan *et al.*, 2017] measures the domain similarity in terms of Maximum Mean Discrepancy (MMD), while [Peng *et al.*, 2019] introduces the metrics based on second-order or higher-order statistics. In addition, there are some methods [Liu *et al.*, 2019; Cui *et al.*, 2020] to learn domain-invariant features through adversarial training.

The second category is to learn discriminative representation using the pseudo-label technique [Li *et al.*, 2021c; Zhang *et al.*, 2021b]. Particularly, in domain adaptive semantic segmentation, recent high-performing methods [Zhang *et al.*, 2021b; Li *et al.*, 2022] commonly use the distillation techniques. Although distillation can greatly improve the accuracy, it is time consuming and complex.

In this paper, we try to fully exploit the potential of contrastive learning for domain adaptation. Our method can be well generalized to classification, detection, and segmentation tasks. Particularly, in the domain adaptive semantic segmentation, PCL based on the non-distilled BAPA [Liu *et al.*, 2021b] can surpass CPSL-D [Li *et al.*, 2022] which uses complex distillation techniques as well as special initialization strategies.

3 Methods

In this section, we first review feature contrastive learning (FCL), and then elaborate on our proposed probabilistic contrastive learning (PCL). Generally speaking, we can split a model for classification, detection, and semantic segmentation into two parts: the encoder E and the classifier F. Here F has the parameters $W = (\mathbf{w}_1, ..., \mathbf{w}_C)$, where C is the number of classes. Particularly, each vector in W is equivalent to the embedding center of a class which we denote as class weights.

[1] In the paper of TCL, the defination of loss is based on the traditional contrastive paradigm (feature+ℓ_2 normalization) and it does not specify the type of feature involved, whether it refers to the feature preceding the classifier, the logits, or the probabilities. The exact form it takes can only be known by examining the code.

3.1 Feature Contrastive Learning

For domain adaptation [Tzeng *et al.*, 2014; Long *et al.*, 2017], the source domain images already have clear supervision signals, and the self-supervised contrastive learning is not urgently required. Thus, we only calculate the contrastive loss (*a.k.a.*, InfoNCE [Oord *et al.*, 2018]) for the target domain data. Specifically, let $\mathcal{B} = \{(x_i, \tilde{x}_i)\}_{i=1}^N$ be a batch of data pairs sampled from target domain, where N is the batch size, and x_i and $\tilde{x}_i$ are two random transformations of a sample. Then, we use E to extract the features, and get $\mathcal{F} = \{(\mathbf{f}_i, \tilde{\mathbf{f}}_i)\}_{i=1}^N$. For a query feature $\mathbf{f}_i$, the feature $\tilde{\mathbf{f}}_i$ is the positive and all other samples are regarded as the negative. Then the InfoNCE loss has the following form:

$$\ell_{\mathbf{f}_i} = -\log \frac{\exp(sg(\mathbf{f}_i)^\top g(\tilde{\mathbf{f}}_i))}{\sum\limits_{j\neq i}\exp(sg(\mathbf{f}_i)^\top g(\mathbf{f}_j)) + \sum\limits_{k}\exp(sg(\mathbf{f}_i)^\top g(\tilde{\mathbf{f}}_k))}, \quad (1)$$

where $g(\mathbf{f}) = \frac{\mathbf{f}}{||\mathbf{f}||_2}$ is a standard ℓ_2 normalization operation, and s is the scaling factor.

From Eq (1), we can observe that there is no class weight information involved in $\ell_{\mathbf{f}_i}$. As a result, in the optimization process of contrastive loss, it is hardly possible to constrain the features to locate around the class weights.

3.2 A Naïve Solution

Now, a natural question is: is it possible to effectively alleviate the deviation problem as long as the class weight information is introduced? We choose logits (*i.e.*, the class scores output by the classifier F) to calculate the contrastive loss as a naive method to introduce class weights information, and refer to it as Logits Contrastive Learning (LCL). However, this approach does not explicitly cluster the features to be close to the class weights and thus does not work experimentally. It shows that simply introducing class weight information without explicit constraints cannot achieve the goal. Therefore, we need to design a new type of contrastive loss to explicitly reduce the deviation between the features and class weights. More discussion about LCL will be provided in Section 4.2.

3.3 Probabilistic Contrastive Learning

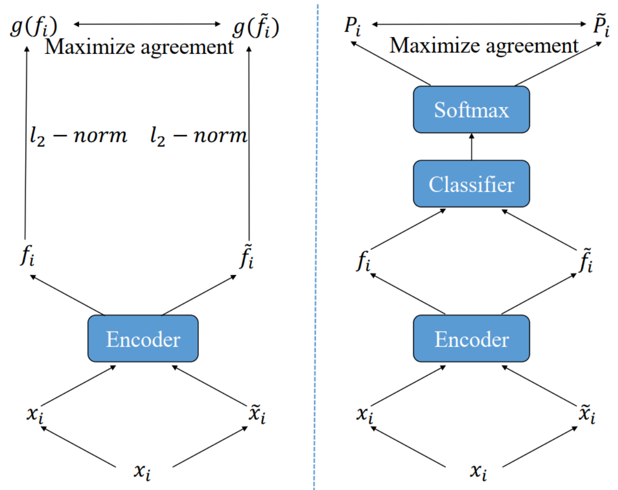
The generalization ability of InfoNCE [Oord *et al.*, 2018] has been fully verified in previous literature. In this work, instead of designing a totally different loss function, we focus on constructing a new input $\mathbf{f}'_i$ to calculate the contrastive loss for the sake of making the feature $\mathbf{f}_i$ close to class weights. Formally, the loss about the new input $\mathbf{f}'_i$ can be written as:

$$\ell_{\mathbf{f}'_i} = -\log \frac{\exp(s{\mathbf{f}'_i}^\top \tilde{\mathbf{f}}'_i)}{\sum_{j\neq i}\exp(s{\mathbf{f}'_i}^\top \mathbf{f}'_j) + \sum_k \exp(s{\mathbf{f}'_i}^\top \tilde{\mathbf{f}}'_k)}. \quad (2)$$

Then our goal is to design a suitable $\mathbf{f}'_i$ so that *the smaller* $\ell_{\mathbf{f}'_i}$ *is, the closer* $\mathbf{f}_i$ *is to the class weights.*

From Eq (2), a smaller $\ell_{\mathbf{f}'_i}$ means a larger ${\mathbf{f}'_i}^\top \tilde{\mathbf{f}}'_i$. Thus the above problem can be roughly simplified to *the larger* ${\mathbf{f}'_i}^\top \tilde{\mathbf{f}}'_i$ *is, the closer* $\mathbf{f}_i$ *is to the class weights.* On the other hand, as elaborated in Section 1, if $\mathbf{f}_i$ is close to the class weight, the corresponding probability $\mathbf{p}_i$ is approximate to the one-hot form:

$$\mathbf{p}_i = (0, .., 1, .., 0). \quad (3)$$



(a) FCL (b) PCL

Figure 3: Framework of FCL and PCL. Different from FCL, PCL uses the output of softmax to perform contrastive learning and removes the ℓ_2 normalization.

Therefore, our goal can be reformulated as how to design a suitable $\mathbf{f}'_i$ so that *the larger* ${\mathbf{f}'_i}^\top \tilde{\mathbf{f}'}_i$ *is, the closer* $\mathbf{p}_i$ *is to the one-hot form.*

Fortunately, we found that the probability $\mathbf{p}_i$ itself can meet such a requirement. Here we explain the mathematical details. Note that $\mathbf{p}_i = (p_{i,1}, ..., p_{i,C})$ and $\tilde{\mathbf{p}}_i = (\tilde{p}_{i,1}, ..., \tilde{p}_{i,C})$ are both the probability distributions. Then we have

$$0 \le p_{i,c} \le 1, 0 \le \tilde{p}_{i,c} \le 1, \forall c \in \{1, ..., C\}. \tag{4}$$

In addition, the ℓ_1-norm of $\mathbf{p}_i$ and $\tilde{\mathbf{p}}_i$ equals one, *i.e.*, $||\mathbf{p}_i||_1 = \sum_c p_{i,c} = 1$ and $||\tilde{\mathbf{p}}_i||_1 = \sum_c \tilde{p}_{i,c} = 1$. Obviously, we have

$$\mathbf{p}_i{}^\top \tilde{\mathbf{p}}_i = \sum_c p_{i,c}\tilde{p}_{i,c} \le 1. \tag{5}$$

The equality is held if and only if $\mathbf{p}_i = \tilde{\mathbf{p}}_i$ and both of them have a one-hot form as in Eq (3). In other words, in order to maximize $\mathbf{p}_i^\top \tilde{\mathbf{p}}_i$, the $\mathbf{p}_i$ and $\tilde{\mathbf{p}}_i$ need to satisfy the one-hot form at the same time. Therefore, $\mathbf{p}_i$ can be served as the new input $\mathbf{f}'_i$ in Eq (2).

Importantly, from the above derivation process, we can see that *the property that the* ℓ_1*-norm of probability equals one is very important*. This property guarantees that the maximum value of $\mathbf{p}_i^\top \tilde{\mathbf{p}}_i$ can only be reached when $\mathbf{p}_i$ and $\tilde{\mathbf{p}}_i$ satisfy the one-hot form at the same time. Evidently, we cannot perform ℓ_2 normalization operation on probabilities like the traditional FCL. Finally, our new contrastive loss is defined by

$$\ell_{\mathbf{p}_i} = -\log \frac{\exp(s\mathbf{p}_i{}^\top \tilde{\mathbf{p}}_i)}{\sum_{j \neq i} \exp(s\mathbf{p}_i{}^\top \mathbf{p}_j) + \sum_k \exp(s\mathbf{p}_i{}^\top \tilde{\mathbf{p}}_k)}. \tag{6}$$

Figure 3 gives an intuitive comparison between FCL and PCL and we can see two main differences. First, Eq (6) uses the probability $\mathbf{p}_i$ instead of the extracted features $\mathbf{f}_i$. Second, Eq (6) removes the ℓ_2 normalization g. It is worth emphasizing that, the rationale behind PCL is the core value of this work, which leads to a convenient implementation. Benefiting from the conciseness, PCL can well generalized to different settings and various methods.

4 Discussion

In this work, we re-examine contrastive learning in domain adaptation from a new perspective, not just based on the broad perspective of "contrastive learning can improve the generalization of features or effectively utilize unlabeled data" [Singh, 2021; Singh *et al.*, 2021]. Specifically, we argue that in domain adaptation tasks, the core reason for the poor performance of contrastive learning is that traditional FCL cannot effectively narrow the distance between features and class weights. Based on the above insights, we propose the PCL and surprisingly find that only employing two simple operations (using probabilities and removing ℓ_2 normalization), without any other techniques, can greatly alleviate the deviation problem. Few works have examined the challenges of contrastive learning in domain adaptation from this perspective, making our analysis and identification of this shortcoming a significant and novel contribution of this paper.

On the other hand, PCL bears similarities to some existing works due to its simplicity. For example, PCL can easily be thought of as a special projection head [Chen *et al.*, 2020]; it can also be viewed as a special entropy minimization loss [Chen *et al.*, 2019; Zhong *et al.*, 2021]. This similarity raises the suspicion that the reason why PCL is effective is just the application of these techniques rather than the novel points we claim. In addition, there are many works [Dwibedi *et al.*, 2021; Khosla *et al.*, 2020] that have greatly improved FCL by mitigating the false negative sample problem. It easily makes us wonder whether PCL is still necessary when false negative samples are mitigated.

In this section, we demonstrate through comprehensive quantitative comparisons that our insights and proposed PCL are the key points in resolving deviation issues and enhancing domain adaptation performance. In the quantitative comparison, we use the typical semi-supervised domain adaptation (SSDA) setting on DomainNet [Peng *et al.*, 2019] with 3-shot and ResNet34 as our benchmark. In particular, we choose MME [Saito *et al.*, 2019] as the baseline model. To ensure fairness, we also keep the training strategy and parameters completely consistent. The comparisons are organized as follows: In Sec. 4.1, we verify whether PCL is better than FCL. In Sec. 4.2, Sec. 4.3, Sec. 4.4, we compare a series of techniques similar to PCL. In Sec. 4.5, we discuss the false negative problems and deviation problems. Finally, in Sec. 4.6, we show the visualization results. We believe these analyses can also provide some useful insights for other visual tasks [Mohri *et al.*, 2019; Li *et al.*, 2021b].

4.1 PCL v.s. FCL

In this part, we compare contrastive learning based on features (FCL) and probabilities (PCL), and present the results in Table 1. It can be seen that PCL can greatly improve the gain of FCL (FCL: 1.8% v.s. PCL: 7.4%).

4.2 PCL v.s. FCL with Projection Head

Projection head [Chen *et al.*, 2020] is a very useful technique that changes the paradigm from feature+ℓ_2 normalization to projection head feature+ℓ_2 normalization. Inspired by this,

Method	R→C	R→P	P→C	C→S	S→P	R→S	P→R	Mean
Baseline	71.4	70.0	72.6	62.7	68.2	64.3	77.9	69.5
+ FCL	72.5	71.6	73.1	66.4	70.2	64.5	80.8	71.3
+ NTCL	72.9	71.3	73.3	66.3	71.3	67.1	80.5	71.7
+ LCL	72.8	70.6	72.5	66.4	70.5	64.5	81.3	71.2
+ PCL-ℓ_2	75.1	74.4	76.2	70.3	73.5	69.9	82.5	74.6
+ Our PCL	**78.1**	**76.5**	**78.6**	**72.5**	**75.6**	**72.5**	**84.6**	**76.9**

Table 1: Classification accuracy (%) of different features on DomainNet under the setting of 3-shot and Resnet34.

CLDA [Singh, 2021] uses the classifier as the projection head to design two contrastive learning losses and achieve better performance. PCL is similar in form to using the classifier as the projection head. In this section, we mainly verify whether the performance gain of our PCL comes from the application of the projection head. To this end, we designed the following three types of projection heads:

(1) Following the SimCLR [Chen *et al.*, 2020], we introduce an additional nonlinear transformation (NT) on the feature. We call it NT-Based Contrastive Learning (NTCL).

(2) We directly use the classifier as the projection head and named it Logits Contrastive Learning (LCL) to introduce class weight information.

(3) We further generalize the projection head to classifier+softmax. For this setting, the contrastive loss paradigm becomes classifier+softmax with ℓ_2 normalization. Essentially, it has only one more ℓ_2 normalization than our PCL, and thus we denote it PCL-ℓ_2. Like LCL, PCL-ℓ_2 is also a way to introduce class weight information. However, it is worth noting that PCL-ℓ_2 is not a natural extension of the projection head technique, since current mainstream contrastive learning methods [Chen *et al.*, 2020; Dwibedi *et al.*, 2021; Chen *et al.*, 2021; Chen and He, 2021] do not include softmax in the projection head. The purpose of this design is to verify such a question: **when probability is used as a special feature, can the traditional contrastive learning paradigm (feature + ℓ_2 normalization) be as effective as PCL?**

Table 1 gives the experimental results and we obtain the following observations. **First**, the above three projection heads all get lower performance than PCL, which indicates that the key reason for the gain of PCL is not from the use of the projection head. **Second**, both LCL and PCL-ℓ_2 are inferior to PCL, which shows that simply introducing class weight information cannot effectively enforce features to gather around class weights. It also means that to reach the goal, the loss function needs to be carefully designed. **Third**, the results experimentally verify the importance of core motivation. Without realizing the problem of features deviating from class weights, we cannot break out of the standard paradigm and induce PCL. Because we have neither reason to use probability nor reason to abandon the widely used ℓ_2 normalization. Even if, like TCL [Singh *et al.*, 2021], happens to extend the projection head to softmax, it still has no enough motivation to remove the ℓ_2 normalization that is widely used in contrastive learning. In this work, however, our core motivation is exactly that the deviation between the features and class weights is a key factor affecting contrastive learning performance. Based on this, we naturally induce the concise form of PCL (in Sec. 3.3).

Method	R→C	R→P	P→C	C→S	S→P	R→S	P→R	Mean
Baseline	71.4	70.0	72.6	62.7	68.2	64.3	77.9	69.5
+ FCL	72.5	71.6	73.1	66.4	70.2	64.5	80.8	71.3
+ SFCL	72.7	72.4	73.2	66.5	70.8	65.5	81.2	71.8
+ BCE	73.3	73.0	74.7	66.7	71.9	67.5	80.4	72.5
+ maxsqures	73.9	72.3	74.0	66.3	71.3	67.6	80.3	72.2
+ FCL+maxsqures	72.9	72.0	73.1	66.7	70.7	66.8	80.7	71.8
+ FCL+BCE	73.9	73.1	74.1	66.5	71.5	67.3	81.4	72.5
+ Our PCL-MSE	76.6	75.3	76.2	69.7	74.2	70.5	83.8	75.1
+ PCL	**78.1**	**76.5**	**78.6**	**72.5**	**75.6**	**72.5**	**84.6**	**76.9**

Table 2: Classification accuracy (%) of different FCL improvement methods on DomainNet under the setting of 3-shot and Resnet34.

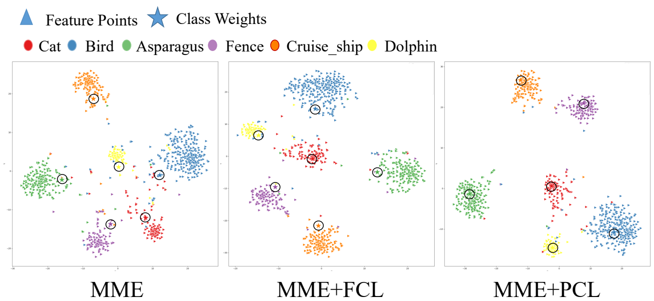


Figure 4: The t-SNE visualization of learned features. Best viewed in color.

4.3 Cosine distance v.s. MSE distance

In domain adaptation, there has been a work [French *et al.*, 2017] that exploits the similarity of the prediction space for consistency constraints, although it does not use the form of contrastive learning. Intuitively, PCL seems to just apply this idea to the contrastive learning loss (transfer feature space to prediction space). Therefore, we need to answer an important question: **is PCL effective only because of the consistency constraint in the prediction space?**

From the previous analysis, PCL naturally requires the use of probabilistic cosine similarity to narrow the distance between features and class weights, rather than just requiring the consistency of the output space. To verify it, we replace the inner product in PCL with the MSE used in [French *et al.*, 2017] to get PCL-MSE. Table 2 gives the results. It can be seen that PCL is better than PCL-MSE. This is because mse can only make the probability similar but not make the probability appear in the one-hot form. This again proves the importance of the motivation of PCL, because this motivation ensures that PCL must use the cosine distance but not the MSE distance.

4.4 PCL v.s. Entropy Minimization

PCL forces probabilities to approximate one-hot form, thereby reducing the entropy of predictions. Therefore, PCL can also be regarded as an entropy minimization loss. Naturally, it also raises an important question: **can we achieve the similar performance by using other classic entropy minimization losses, such as maxsqures loss [Chen *et al.*, 2019] and binary cross entropy loss (BCE) [Zhong *et al.*, 2021]?**

Table 2 gives the results and we have the following observations.

First, like PCL, the optimization goals of maxsquares loss and BCE loss can force the probability to approach the one-hot form. Therefore, they can indeed bring gains based on the baseline (*e.g.*, maxsquare: $69.5\% \rightarrow 72.2\%$, BCE: $69.5\% \rightarrow 72.5\%$).

Second, compared to PCL, the gains of maxsquares loss and BCE loss are very limited, which reflects the fact that maintaining the InfoNCE format is critical to the success of PCL. As we emphasized in Sec.1, for recognition tasks, we not only require the features to be close to the class weights, but also require the features themselves to be compact enough. However, for BCE loss and maxSquares loss, although they can also make the probability approach the one-hot form, they use a pair-wise form of loss to constrain the sample (maxsqure loss even discards negative samples and different data augmentations), which is far less effective than the InfoNCE form in learning compact feature representations [Wang and Liu, 2021].

4.5 PCL v.s. SFCL

In original contrastive learning loss, there may be some negative samples that belong to the same category as the query sample, which are called false negative samples. Many methods [Dwibedi *et al.*, 2021; Khosla *et al.*, 2020] point out that false negative samples are harmful. In this section, we try to answer another important question: **whether can we make FCL work well by reducing the false negative samples without designing PCL?**

An appropriate way to address the false negative samples problem is to use supervised feature contrastive learning (SFCL) [Khosla *et al.*, 2020]. Table 2 shows the experimental comparison. It can be seen that SFCL can indeed improve the performance of FCL. However, compared with PCL, SFCL has a very limited improvement over FCL. Specifically, SFCL can learn better feature representations by alleviating the false negative problem, but it cannot solve the problem of the deviation between the features and class weights. The experimental results reveal that the deviation problem is more critical than the false negative samples for domain adaptation.

4.6 Visualization Analysis

Figure 4 shows the relationship between the unlabeled features and the class weights for the three methods, including MME, MME+FCL, and MME+PCL. Firstly, compared with MME, MME+FCL produces more compact feature clusters for the same category and more separate feature distributions for different categories. However, the learned class weights deviate from the feature centers for both MME+FCL and MME. Secondly, the class weights of MME+PCL are much closer to the feature centers than MME+FCL. It demonstrates that PCL is significantly effective in enforcing the features close to the class weights.

5 Experiments

In this section, we will verify the validity of the PCL on five different tasks. In order to ensure fairness, in each task, we strictly follow the baseline experimental settings and only add additional PCL loss.

Methods	GTA5	SYNTHIA	
	mIoU (%)	mIoU-13 (%)	mIoU-16 (%)
ProDA (CVPR'21) [Zhang *et al.*, 2021b]	53.7	-	-
CPSL(CVPR'22) [Li *et al.*, 2022]	55.7	61.7	54.4
BAPA (ICCV'21) [Liu *et al.*, 2021b]	57.4	61.2	53.3
ProDA-D (CVPR'21) [Zhang *et al.*, 2021b]	57.5	62.0	55.5
ProDA-D+CaCo (CVPR'22) [Huang *et al.*, 2022]	58.0	-	-
CPSL-D (CVPR'22) [Li *et al.*, 2022]	**60.8**	65.3	57.9
BAPA*	57.7	60.1	53.3
+ Our PCL	60.7	**68.2**	**60.3**

Table 3: Result on UDA Semantic Segmentation. -D means to use an additional two-step distillation technique. $*$ means our reimplementation.

Method	AP	Method	AP
MeGA-CDA (CVPR'21) [VS *et al.*, 2021]	44.8	RPA (CVPR'21)* [Zhang *et al.*, 2021c]	45.3
UMT (CVPR'21) [Deng *et al.*, 2021]	43.1	**+ Our PCL**	**47.8**

Table 4: Detection performance (%) on UDA detection task. $*$ means our reimplementation.

5.1 UDA Semantic Segmentation

Setup We evaluate our method on two standard UDA semantic segmentation tasks: GTA5 [Richter *et al.*, 2016]→Cityscapes [Cordts *et al.*, 2016] and SYNTHIA [Ros *et al.*, 2016]→Cityscapes.

The current SOTA methods generally adopt the distillation technique for post-processing. It makes the training process very complicated and requires some special training strategies. Therefore, here we divide these methods into simple non-distilled methods and complex distillation methods. In particular, we take the non-distilled BAPA [Liu *et al.*, 2021b] with ResNet-101 [He *et al.*, 2016] as our baseline due to its simplicity and efficiency. For the hyperparameter in PCL, we set $s = 20$ in all experiments.

Table 3 gives the results. First, our method can achieve very significant gains on the baseline and outperforms all non-distilled methods by a large margin on SYNTHIA (6.5% for mIoU-13 and 5.9% of mIoU-16). Second, even compared to distillation-based methods, our method has only slightly lower performance than CPSL-D on GTA5 and outperforms CPSL-D by more than 2% on SYNTHIA. Notably, the training cost of our method is much lower than CPSL-D (PCL: 1*3090, 5 days v.s. CPSL-D: 4*V100, 11 days).

5.2 UDA Detection

Setup We conduct an experiment on SIM10k [Johnson-Roberson *et al.*, 2017] → Cityscapes [Cordts *et al.*, 2016] scenes to verify effective of our PCL for the object detection task. In particular, we choose the RPA [Zhang *et al.*, 2021c] with Vgg16 [Simonyan and Zisserman, 2014] as the baseline. We add PCL to the classification head to improve the classification results of the RPA model. For the hyperparameter in PCL, we set $s = 20$ in the experiment.

Table 4 gives the results. It can be seen that PCL can significantly improve the performance of the RPA model. This proves the effectiveness of PCL on the UDA detection task.

5.3 UDA Classification

Setup We evaluate our PCL in the following two standard benchmarks: **Office-Home** [Venkateswara *et al.*, 2017] and

Method	Acc	Method	Acc
SDAT (ICML'22) [Rangwani *et al.*, 2022]	72.2	SCDA (ICCV'21) [Li *et al.*, 2021d]	73.1
NWD (CVPR'22) [Chen *et al.*, 2022]	72.6	HMA (ICCV'23) [Zhou *et al.*, 2023]	73.2
GVB* (CVPR'20) [Cui *et al.*, 2020]	70.3	GVB†	73.5
+ MetaAlign (CVPR'21) [Wei *et al.*, 2021]	71.3	-	-
+ Our PCL	**72.3**	**+ Our PCL**	**74.5**

Table 5: Average performance (%) of 12 UDA tasks on Office-Home. * means our reimplementation.

Method	Acc	Method	Acc
CST (NeurIPS'21) [Liu *et al.*, 2021a]	80.6	SENTRY (ICCV'21) [Prabhu *et al.*, 2021]	76.7
GVB* (CVPR'21) [Cui *et al.*, 2020]	75.0	GVB†	80.4
+ Our PCL	80.8	**+ Our PCL**	**82.5**

Table 6: Classification accuracy (%) of Synthetic→Real on VisDA-2017 for UDAs. * means our reimplementation.

VisDA-2017 [Peng *et al.*, 2017]. We take GVB-GD [Cui *et al.*, 2020] with ResNet50 [He *et al.*, 2016] as our baseline. For the hyperparameter in PCL, we set $s = 7$ in all experiments.

Table 5 and Table 6 give the results. In particular, inspired by some previous semi-supervised domain adaptation methods [Li *et al.*, 2021a; Li *et al.*, 2021c], we add FixMatch [Sohn *et al.*, 2020] to GVB and get the stronger GVB†. It can be seen that our method can bring considerable gains both on GVB and GVB†. In particular, with GVB as the baseline, PCL outperforms MetaAlign by 1%, which involves more complicated operations than PCL. This further demonstrates the superiority of PCL.

5.4 Semi-Supervised Domain Adaptation

Setup We evaluate the effectiveness of our proposed approach on two SSDA benchmarks, *i.e.*, *DomainNet* [Peng *et al.*, 2019] and *Office-Home*. We choose MME [Saito *et al.*, 2019] with ResNet34 [He *et al.*, 2016] as our baseline model. In particular, inspired by CDAC [Li *et al.*, 2021a] and ECACL-P [Li *et al.*, 2021c], we add FixMatch [Sohn *et al.*, 2020] to MME to build a stronger MME†. For the hyperparameter in PCL, we set $s = 7$ in all experiments.

Table 7 gives the results and we obtain the following observations: 1) PCL outperforms the methods without Fixmatch for most of settings. In particular, CLDA uses a classifier as the projection head for instance-level and class-level contrastive learning. Our PCL can defeat CLDA, although PCL is only equipped with instance contrastive learning. The results indicate the superiority of PCL over the projection head. 2) Comparing with the methods using Fixmatch, PCL has obvious advantages in terms of simplicity and effectiveness. First of all, these methods, in addition to using Fixmatch, also carefully design many complex strategies to enhance performance. For example, ECACL-P designs the prototypical loss and triplet loss. Even so, without relying on Fixmatch, simple PCL can almost equal these complex methods. After equipping Fixmatch, PCL achieved clear advantages in all settings.

Methods	Office-Home	DomainNet	
	3-shot	1-shot	3-shot
CLDA (NeurIPS'21) [Singh, 2021]	**75.5**	71.9	75.3
MME* (ICCV'19) [Saito *et al.*, 2019]	73.5	67.9	69.5
+ Our PCL	**75.5**	**73.5**	**76.9**
ECACL-P† (ICCV'21) [Li *et al.*, 2021c]	-	72.8	76.4
MCL† (IJCAI'22) [Yan *et al.*, 2022]	77.1	74.4	76.5
ProMM† (IJCAI'23) [Huang *et al.*, 2023]	77.8	76.1	77.4
CDAC† (CVPR'21) [Li *et al.*, 2021a]	74.8	73.6	76.0
+SLA (CVPR'23) [Yu and Lin, 2023]	76.3	75.0	76.9
MME†	76.9	72.9	76.1
+ Our PCL	**78.1**	**75.1**	**78.2**

Table 7: Average performance (%) on *DomainNet* (7 tasks) and *Office-Home* (12 tasks). $*$ means our reimplementation.

Method	4-shot Acc	Method	4-shot Acc
Freematch (ICLR'23) [Wang *et al.*, 2023]	62.02±0.42	Softmatch (ICLR'23) [Chen *et al.*, 2023]	62.90±0.77
FixMatch* (NeurIPS'20) [Sohn *et al.*, 2020]	53.58± 2.09	**+ Our PCL**	**57.62±2.52**
CCSSL* (CVPR'22) [Yang *et al.*, 2022]	60.49±0.57	**+ Our PCL**	**62.95**±1.39
FlexMatch* (NeurIPS'21) [Zhang *et al.*, 2021a]	61.78± 1.17	**+ Our PCL**	**64.15±0.53**

Table 8: Classification accuracy (%) of SSL for CIFAR-100 (400 labels). * means our reimplementation.

5.5 Semi-Supervised Learning

In fact, as long as the features of unlabeled data cannot be clustered around the class weights, PCL has good potential to improve the performance. In this section, we consider the case where the source domain and target domain come from the same distribution, *i.e.*, semi-supervised learning. In particular, for the semi-supervised tasks, the unlabeled features will deviate from the class weight when the labeled data is very scarce. Therefore, we consider the case where there are only 4 labeled samples per class.

Setup We conduct the experiments on CIFAR-100 [Krizhevsky *et al.*, 2009] and take three SSL methods, including Fixmatch [Sohn *et al.*, 2020], CCSSL [Yang *et al.*, 2022] and Flexmatch [Zhang *et al.*, 2021a] as our baseline. For the hyperparameter in PCL, we set $s = 7$ in all experiments.

The evaluation results are reported in Table 8. For these three different baselines, PCL can bring significant gains. This further proves our conclusion that PCL has the potential to improve model performance in scenarios where feature and class weights deviate.

6 Discussion and Conclusion

In this paper, we propose a simple yet effective probabilistic contrastive learning to address the problem of feature deviation from weights. Therefore, our method has shown positive results on multiple domain adaptation tasks. An open question worth discussing is: Can PCL be applied into general contrastive learning (GCL) for classification? Due to the absence of a classifier in the unsupervised pre-training stage of GCL, directly applying PCL meets challenges. A feasible solution is to employ a clustering algorithm to construct class centers and then use PCL to learn more robust features. We hope that PCL can bring some useful insights into general unsupervised representation learning tasks.

Acknowledgments

This work is jointly supported by the National Natural Science Foundation of China under Grant (62176246, 62276031) and Anhui Provincial Key Research and Development Project (202304a05020045).

References

[Chen and He, 2021] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021.

[Chen *et al.*, 2019] Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. In *ICCV*, 2019.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[Chen *et al.*, 2021] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[Chen *et al.*, 2022] Lin Chen, Huaian Chen, Zhixiang Wei, Xin Jin, Xiao Tan, Yi Jin, and Enhong Chen. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In *CVPR*, 2022.

[Chen *et al.*, 2023] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *ICLR*, 2023.

[Cordts *et al.*, 2016] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[Cui *et al.*, 2020] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*, 2020.

[Deng *et al.*, 2021] Jinhong Deng, Wen Li, Yuhua Chen, and Lixin Duan. Unbiased mean teacher for cross-domain object detection. In *CVPR*, 2021.

[Dwibedi *et al.*, 2021] 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. *ICCV*, 2021.

[French *et al.*, 2017] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Huang *et al.*, 2022] Jiaxing Huang, Dayan Guan, Aoran Xiao, Shijian Lu, and Ling Shao. Category contrast for unsupervised domain adaptation in visual tasks. In *CVPR*, 2022.

[Huang *et al.*, 2023] Xinyang Huang, Chuang Zhu, and Wenkai Chen. Semi-supervised domain adaptation via prototype-based multi-level learning. *IJCAI*, 2023.

[Johnson-Roberson *et al.*, 2017] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *ICRA*, 2017.

[Khosla *et al.*, 2020] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *NeurIPS*, 2020.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Citeseer, 2009.

[Li *et al.*, 2021a] Jichang Li, Guanbin Li, Yemin Shi, and Yizhou Yu. Cross-domain adaptive clustering for semi-supervised domain adaptation. In *CVPR*, 2021.

[Li *et al.*, 2021b] Junjie Li, Zilei Wang, and Xiaoming Hu. Learning intact features by erasing-inpainting for few-shot classification. In *AAAI*, 2021.

[Li *et al.*, 2021c] Kai Li, Chang Liu, Handong Zhao, Yulun Zhang, and Yun Fu. Ecacl: A holistic framework for semi-supervised domain adaptation. In *ICCV*, 2021.

[Li *et al.*, 2021d] Shuang Li, Mixue Xie, Fangrui Lv, Chi Harold Liu, Jian Liang, Chen Qin, and Wei Li. Semantic concentration for domain adaptation. In *ICCV*, 2021.

[Li *et al.*, 2022] Ruihuang Li, Shuai Li, Chenhang He, Yabin Zhang, Xu Jia, and Lei Zhang. Class-balanced pixel-level self-labeling for domain adaptive semantic segmentation. In *CVPR*, 2022.

[Liu *et al.*, 2019] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *ICML*, 2019.

[Liu *et al.*, 2021a] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. *Advances in Neural Information Processing Systems*, 34:22968–22981, 2021.

[Liu *et al.*, 2021b] Yahao Liu, Jinhong Deng, Xinchen Gao, Wen Li, and Lixin Duan. Bapa-net: Boundary adaptation and prototype alignment for cross-domain semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8801–8811, 2021.

[Long *et al.*, 2017] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, 2017.

[Mohri *et al.*, 2019] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *ICML*. PMLR, 2019.

[Na *et al.*, 2021] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *CVPR*, 2021.

[Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[Peng *et al.*, 2017] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko.

Visda: The visual domain adaptation challenge. In *arXiv preprint arXiv:1710.06924*, 2017.

[Peng *et al.*, 2019] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

[Prabhu *et al.*, 2021] Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8558–8567, 2021.

[Rangwani *et al.*, 2022] Harsh Rangwani, Sumukh K Aithal, Mayank Mishra, Arihant Jain, and Venkatesh Babu Radhakrishnan. A closer look at smoothness in domain adversarial training. In *ICML*, 2022.

[Richter *et al.*, 2016] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, pages 102–118. Springer, 2016.

[Ros *et al.*, 2016] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, pages 3234–3243, 2016.

[Saito *et al.*, 2019] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *ICCV*, 2019.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Singh *et al.*, 2021] Ankit Singh, Omprakash Chakraborty, Ashutosh Varshney, Rameswar Panda, Rogerio Feris, Kate Saenko, and Abir Das. Semi-supervised action recognition with temporal contrastive learning. In *CVPR*, pages 10389–10399, 2021.

[Singh, 2021] Ankit Singh. Clda: Contrastive learning for semi-supervised domain adaptation. *NeurIPS*, 2021.

[Sohn *et al.*, 2020] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

[Tzeng *et al.*, 2014] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[Venkateswara *et al.*, 2017] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, , and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

[VS *et al.*, 2021] Vibashan VS, Vikram Gupta, Poojan Oza, Vishwanath A. Sindagi, and Vishal M. Patel. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. In *CVPR*, 2021.

[Wang and Liu, 2021] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *CVPR*, 2021.

[Wang *et al.*, 2023] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Zhen Wu, and Jindong Wang. Freematch: Self-adaptive thresholding for semi-supervised learning. *ICLR*, 2023.

[Wei *et al.*, 2021] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation. In *CVPR*, 2021.

[Yan *et al.*, 2017] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *CVPR*, 2017.

[Yan *et al.*, 2022] Zizheng Yan, Yushuang Wu, Guanbin Li, Yipeng Qin, Xiaoguang Han, and Shuguang Cui. Multi-level consistency learning for semi-supervised domain adaptation. *IJCAI*, 2022.

[Yang *et al.*, 2022] Fan Yang, Kai Wu, Shuyi Zhang, Guannan Jiang, Yong Liu, Feng Zheng, Wei Zhang, Chengjie Wang, and Long Zeng. Class-aware contrastive semi-supervised learning. *CVPR*, 2022.

[Yu and Lin, 2023] Yu-Chu Yu and Hsuan-Tien Lin. Semi-supervised domain adaptation with source label adaptation. In *CVPR*, pages 24100–24109, 2023.

[Zhang *et al.*, 2021a] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *NeurIPS*, 2021.

[Zhang *et al.*, 2021b] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *CVPR*, 2021.

[Zhang *et al.*, 2021c] Yixin Zhang, Zilei Wang, and Yushi Mao. Rpn prototype alignment for domain adaptive object detector. In *CVPR*, 2021.

[Zhong *et al.*, 2021] Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *CVPR*, 2021.

[Zhou *et al.*, 2023] Lihua Zhou, Mao Ye, Xiatian Zhu, Siying Xiao, Xu-Qian Fan, and Ferrante Neri. Homeomorphism alignment for unsupervised domain adaptation. In *ICCV*, pages 18699–18710, 2023.