Probabilistic Contrastive Learning for Domain Adaptation

Junjie Li\textsuperscript{1,2}, Yixin Zhang\textsuperscript{2}, Zilei Wang\textsuperscript{2*}, Saihui Hou\textsuperscript{3}, Keyu Tu\textsuperscript{2}, Man Zhang\textsuperscript{1}

\textsuperscript{1}Beijing University of Posts and Telecommunications
\textsuperscript{2}University of Science and Technology of China
\textsuperscript{3}Beijing Normal University

hnjj93@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+\ell_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 \textit{simple but powerful} Probabilistic Contrastive Learning (PCL), which moves beyond the standard paradigm by removing \ell_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 \textgreater \textasciitilde2\% 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.

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 \textit{et al.}, 2017; Na \textit{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 \textit{et al.}, 2020; Khosla \textit{et al.}, 2020]. Inspired by its great success for representation learning, we hope to perform the standard contrastive learning (features+\ell_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 (\textit{e.g.}, 64.3\% \rightarrow 64.5\% as shown in Figure 2). A natural question arises and motivates this work: \textit{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 (\textit{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
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 $\ell_2$ 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 negatives, effectively improving the performance. TCL [Singh et al., 2021] extends this idea from instance-level to group-level and proposes group contrastive 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. It merely treats probability as a special feature, and naturally preserves $\ell_2$ normalization, that is, PCL-$\ell_2$. Therefore, whether in form or principle, TCL (PCL-$\ell_2$) is still a standard contrastive paradigm (features+$\ell_2$ normalization). Experiments show that TCL (PCL-$\ell_2$) 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 use 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 $W = (w_1, ..., w_C)$, 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.

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 $B = \{(x_i, x'_i)\}_{i=1}^N$ be a batch of data pairs sampled from target domain, where $N$ is the batch size, and $x_i$ and $x'_i$ are two random transformations of a sample. Then, we use $E$ to extract the features, and get $F = \{(f_i, f'_i)\}_{i=1}^N$. For a query feature $f_i$, the feature $f'_i$ is the positive and all other samples are regarded as the negative. Then the InfoNCE loss has the following form:

$$\ell_{\text{FCL}} = -\log \frac{\exp(s g(f_i)\top g(f'_i))}{\sum_{j \neq i} \exp(s g(f_j)\top g(f'_i)) + \sum_k \exp(s g(f'_i)\top g(f'_k))}, \quad (1)$$

where $g(f) = \frac{f}{||f||_2}$ is a standard $\ell_2$ normalization operation, and $s$ is the scaling factor.

From Eq (1), we can observe that there is no class weight information involved in $\ell_{\text{FCL}}$. 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 naÃ¯ve 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 $f'_i$ to calculate the contrastive loss for the sake of making the feature $f_i$ close to class weights. Formally, the loss about the new input $f'_i$ can be written as:

$$\ell_{\text{FCL}} = -\log \frac{\exp(s f'_i\top f_i)}{\sum_{j \neq i} \exp(s f'_i\top f'_j) + \sum_k \exp(s f'_j\top f'_k)} \quad (2)$$

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

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

$$p_i = (0, ..., 1, ..., 0) \quad (3)$$
Therefore, our goal can be reformulated as how to design a suitable $f'_i$ so that the larger $f'_i^\top \tilde{f}'_i$, the closer $p_i$ is to the one-hot form.

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

$$0 \leq p_{i,c} \leq 1, \; \forall c \in \{1, ..., C\}. \quad (4)$$

In addition, the $\ell_1$-norm of $p_i$ and $\tilde{p}_i$ equals one, i.e., $||p_i||_1 = \sum_c p_{i,c} = 1$ and $||\tilde{p}_i||_1 = \sum_c \tilde{p}_{i,c} = 1$. Obviously, we have

$$p_i^\top \tilde{p}_i = \sum_c p_{i,c} \tilde{p}_{i,c} \leq 1. \quad (5)$$

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

Importantly, from the above derivation process, we can see that the property that the $\ell_1$-norm of probability equals one is very important. This property guarantees that the maximum value of $p_i^\top \tilde{p}_i$ can only be reached when $p_i$ and $\tilde{p}_i$ satisfy the one-hot form at the same time. Evidently, we cannot perform $\ell_2$ normalization operation on probabilities like the traditional FCL. Finally, our new contrastive loss is defined by

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

Figure 3 gives an intuitive comparison between FCL and PCL and we can see two main differences. First, Eq (6) uses the probability $p_i$ instead of the extracted features $f_i$. Second, Eq (6) removes the $\ell_2$ 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 $\ell_2$ 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+$\ell_2$ normalization to projection head feature+$\ell_2$ normalization. Inspired by this,
Asparagus
MME+FCL
Class Weights
Dolphin
MME+PCL

Concise form of PCL (in Sec. 3.3). Based on this, we naturally induce the features and class weights is a key factor affecting contrastive learning. 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 $\ell_2$ normalization. Essentially, it has only one more $\ell_2$ normalization than our PCL, and thus we denote it PCL-$\ell_2$. Like LCL, PCL-$\ell_2$ is also a way to introduce class weight information. However, it is worth noting that PCL-$\ell_2$ 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 + $\ell_2$ 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-$\ell_2$ 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 $\ell_2$ 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 $\ell_2$ 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).

<table>
<thead>
<tr>
<th>Method</th>
<th>R→C</th>
<th>R→P</th>
<th>P→C</th>
<th>C→S</th>
<th>S→P</th>
<th>R→S</th>
<th>P→R</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>71.4</td>
<td>70.0</td>
<td>72.6</td>
<td>62.7</td>
<td>68.2</td>
<td>64.3</td>
<td>77.9</td>
<td>69.5</td>
</tr>
<tr>
<td>+ FCL</td>
<td>72.5</td>
<td>71.0</td>
<td>72.1</td>
<td>66.4</td>
<td>70.2</td>
<td>64.3</td>
<td>80.8</td>
<td>71.3</td>
</tr>
<tr>
<td>+ MME+FCL</td>
<td>72.9</td>
<td>71.3</td>
<td>73.3</td>
<td>66.3</td>
<td>71.3</td>
<td>67.1</td>
<td>80.5</td>
<td>71.7</td>
</tr>
<tr>
<td>+ LCL</td>
<td>72.8</td>
<td>70.6</td>
<td>72.5</td>
<td>66.4</td>
<td>70.5</td>
<td>64.5</td>
<td>81.3</td>
<td>71.2</td>
</tr>
<tr>
<td>+ PCL-$\ell_2$</td>
<td>75.1</td>
<td>74.4</td>
<td>76.2</td>
<td>70.3</td>
<td>73.5</td>
<td>69.9</td>
<td>82.5</td>
<td>74.6</td>
</tr>
<tr>
<td>+ Our PCL</td>
<td>78.1</td>
<td>76.5</td>
<td>78.6</td>
<td>72.5</td>
<td>75.6</td>
<td>72.5</td>
<td>84.6</td>
<td>76.9</td>
</tr>
</tbody>
</table>

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

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.

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

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

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 maxsquares 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

Figure 4: The t-SNE visualization of learned features. Best viewed in color.
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% → 72.2%, BCE: 69.5% → 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 (maxsquare 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.

<table>
<thead>
<tr>
<th>Methods</th>
<th>GTA5 mIoU(%)</th>
<th>SYNTIHA mIoU-13(%)</th>
<th>SYNTIHA mIoU-16(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProDA (CVPR’21) [Zhang et al., 2020]</td>
<td>53.7</td>
<td>61.7</td>
<td>54.4</td>
</tr>
<tr>
<td>CPSL (CVPR’22) [Li et al., 2022]</td>
<td>57.4</td>
<td>62.2</td>
<td>53.3</td>
</tr>
<tr>
<td>BAPA (ICCV’21) [Liu et al., 2021b]</td>
<td>59.3</td>
<td>62.0</td>
<td>55.3</td>
</tr>
<tr>
<td>ProDA-D+CaCo (CVPR’22) [Huang et al., 2022]</td>
<td>58.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CPSL-D (CVPR’22) [Li et al., 2022]</td>
<td>60.8</td>
<td>65.3</td>
<td>57.9</td>
</tr>
<tr>
<td>+ Our PCL</td>
<td>61.7</td>
<td>68.2</td>
<td>60.3</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Methods</th>
<th>mAP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MeGA-CDA (CVPR’21) [YS et al., 2021]</td>
<td>44.8</td>
</tr>
<tr>
<td>UMT (CVPR’21) [Dong et al., 2021]</td>
<td>43.1</td>
</tr>
<tr>
<td>+ Our PCL</td>
<td>47.8</td>
</tr>
</tbody>
</table>

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 SYNTIHA [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 SYNTIHA (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 SYNTIHA. 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
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


[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.


[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.


