A Prototype-Oriented Contrastive Adaption Network For Cross-domain Facial Expression Recognition

Chao Wang[1], Jundi Ding[1](✉), Hui Yan[1], and Si Shen[1]

Nan Jing University of Science & Technology

Abstract. Numerous well-performing facial expression recognition algorithms suffer from severe slippage when trained on one dataset and tested on another, due to inconsistencies in facial expression datasets caused by different acquisition conditions and subjective biases of annotators. In order to improve the generalization ability of the model, in this paper we propose a simple but effective Prototype-Oriented Contrastive Adaptation Network (POCAN) unified contrastive learning and prototype networks for cross-domain facial expression recognition. We employ a two-stage training pipeline. Specifically, in the first stage, we pre-train on the source domain to obtain semantically meaningful features and obtain good initial conditions for the target domain. In the second stage, we perform intra-domain feature learning and inter-domain feature fusion by narrowing the distance between samples and their corresponding prototypes and widening the distance with other prototypes, and we also use an adversarial loss function for domain-level alignment. In addition, we also consider the problem of data category imbalance, and category weights are introduced into our method so that the categories of the two domains are in a uniform distribution. Extensive experiments show that our method can yield competitive performance on both lab-controlled and in-the-wild datasets.

Keywords: Cross-domain facial expression recognition · Prototype network · Contrastive learning

1 Introduction

Facial expressions(referring to the emotions shown on the face) are one of the most direct and natural ways to convey human emotions. How understanding human beings' emotion is a key problem that needs to be solved in the fields of human-computer interaction, perceptual psychology, intelligent control, medical care, security, etc. Therefore, the research of facial expression recognition(FER) is getting growing attention. Facial expressions are a complex signal, humans have many expressions, and the types of expressions in different regions and cultures are different. However, some facial expressions are universal across cultures. The researchers summarize these expressions as basic expressions. In the 1990s, Ekman et al. [6] proposed six basic expressions through cross-cultural research:

fear, sadness, anger, disgust, surprise, and happiness. Later researchers added neutral to the basic expression.

Early studies of facial expression recognition were conducted on lab-controlled datasets using hand-crafted features (*e.g.*, local binary patterns (LBP [22])) and shallow learning (*e.g.*, sparse learning [34]). Lab-controlled datasets with acted facial expressions collect in artificial environments and do not correspond to the spontaneous, unconstrained situation in practice. Moreover, Shallow learning can only extract shallow features and cannot extract deep features, resulting in poor algorithm performance. As deep learning has achieved remarkable results in many fields, deep learning technology, especially CNN, has been applied to facial expression recognition, and has achieved remarkable progress. At the same time, many facial expression datasets from unconstrained scenarios were collected. Both contributed to the development of facial expression recognition.

Although today's excellent facial expression algorithms can achieve satisfactory performance, these algorithms are trained and tested on a single dataset. When using these algorithms to test on other datasets, they suffer from dramatic performance deterioration [15]. The reason for this phenomenon is due to the existence of domain shift: different facial expression datasets are collected in different contexts, and different annotators perceive expressions differently. Therefore, cross-domain facial expression recognition is a more meaningful and challenging task, which is closely related to the field of domain adaptation in the field of transfer learning. Recently, many studies have begun to focus on cross-domain facial expression recognition and have proposed many effective methods. They either use MMD [13,15] to minimize domain distribution differences, or borrow the idea of adversarial learning [2] to learn cross-domain invariant features, but there are no methods using contrastive learning.

In this paper, motivated by the success of contrastive learning in representation learning, we use the idea of contrastive learning to solve the problem of cross-domain facial expression recognition and propose POCAN[1], a two-stage algorithm.

- In the first stage, we use supervised contrastive learning to perform pretraining on the source domain. By making the samples of the same category close and the samples of different categories separate, we can learn the category information of the source domain and obtain a better initial conditions for the target domain. Inspired by [23], we parameterize the category prototype and learn the category prototypes by bringing the prototype close to the samples belonging to the class it represents. Compared with computing prototypes by clustering, this learnable prototype acquisition method can reduce computational consumption and improve training speed.

- In the second stage, we learn the features of the source and target domains separately. For the target domain features, we use self-supervised contrastive learning and prototype contrastive learning to learn, which can compensate

[1] The source code and the trained model are publicly available at https://github.com/Winter-is-coming-wow/ACCV2022_POCAN.git.

for the lack of self-supervised contrastive learning ability to obtain high-level semantic knowledge, while maintaining the local smoothness of the learned representations. We use supervised contrastive learning and prototype contrastive learning to learn the features of the source domain to keep the model's low error rate in the source domain. At the same time, we carry out a two-level domain fusion strategy, one is to make the samples of one domain close to the prototypes of the same category in the other domain to perform domain alignment at the category level, and the other cleverly uses classifiers to form a domain discriminator and then uses the adversarial method for domain-level alignment.

2 Related works

In this section, we briefly review two related topics: cross-domain facial expression recognition and contrastive learning.

2.1 Cross-Domain Facial Expression Recognition

The serious decline of traditional facial expression recognition algorithms in cross-dataset experiments has made researchers realize the importance of improving the generalization ability of algorithms. Many cross-domain facial expression recognition algorithms have been proposed in the last few years. In [29], Transfer Subspace Learning is introduced into cross-domain facial expression recognition, this method learns a subspace in which the knowledge learned from the source domain is transferred to the target domain. In [30], the author proposed unsupervised domain adaptive dictionary learning (UDADL) to learn a reliable synthesis dictionary for both source and target samples such that the distribution mismatch between two different domains could be alleviated. In [35], a discriminative feature adaptation method is proposed to minimize the mismatch between source and target distribution by learning a new feature space. In [13], the author embeds maximum mean inconsistency (MMD) into deep networks to reduce inconsistency between datasets, while introducing a learnable parameter to reduce the impact of the source domain and target and class distribution differences. In [26], a generative adversarial network (GAN) is introduced to generate more target domain data, and a distributed pseudo-label method is applied to achieve domain adaptation with limited target data without ground truth labels. In [15], the author found that the conditional probabilities between different datasets are different by measuring the deviation between facial expression datasets. Therefore, ECAN is proposed to learn domain-invariant features, align the marginal probability distribution and conditional probability distribution of the source and target domains at the same time, and consider the class imbalance problem between domains. Recently, In [2] joint Adversarial Learning and Graph Representation Learning for simultaneous domain adaptation of global and local Features.

2.2 Contrastive Learning

Unsupervised/self-supervised contrastive learning has great success in representation learning. The basic idea of contrastive learning is to make pairs of positive samples more similar and pairs of negative samples less similar via a contrastive loss in the latent space. The common contrastive loss function is InfoNEC loss [20], the loss for a sample is defined as:

$$L_{InfoNEC}(z_i) = -log(\frac{exp(sim(z_i, z_i')/\tau)}{\sum_{j=1}^{n} exp(sim(z_i, z_j)/\tau)}), \tag{1}$$

where sim() is the similarity function (cosine similarity is commonly used), n is number of sample pairs, τ denotes a temperature parameter, it is an important parameter that controls the strength of the penalty for hard negative samples [25].

In instance-based self-supervised contrastive learning, samples from the same instance are regarded as pairs of positive samples, and samples from different instances are regarded as pairs of negative samples. Positive samples always come from another views of a instance, negative samples come from a memory bank [27], or a queue [9], or a minibatch [3]. Although instance-based self-supervised contrastive learning has good performance, this method has an inherent disadvantage: it treats two samples as a negative sample pair as long as they come from different instances, which unavoidably determines some samples with the same semantics as the anchor as negative samples. This negatively affects the performance of the algorithm. To solve this problem, [4]develop a debiased contrastive objective that corrects for the sampling of same-label datapoints without knowledge of the true labels. [12]propose prototypical contrastive learning (PCL) which encodes the semantic structure of data into the embedding space through make samples similar with its prototype.

3 Method

In the context of unsupervised cross-domain facial expression recognition, we have a labeled source dataset $D_s = \{x_i^s, y_i^s\}_{i=1}^{N_s}$ with marginal probability distribution $P^s(X)$ and an unlabeled target dataset $D_t = \{x_i^t\}_{i=1}^{N_t}$ with marginal probability distribution $P^t(X)$, where N_s and N_t are the number of images in source and target domains respectively. Both x_i^s and x_i^t belong to one of the 7 predefined expressions: anger, disgust, fear, happiness, sadness, surprise, and neutral. The two datasets are drawn from different distributions (i.e., $P^s(X)! = P^t(X)$). Our task is to train a model using the labeled source dataset and the unlabeled target dataset, reducing the domain offset between the source and target domains, and minimizing the error rate of the model on the target domain.

As show in Fig.1, our model consists of a encoder $f(\cdot)$, a projection head $g(\cdot)$, a learnable source domain prototype matrix M_s and a learnable target domain prototype matrix M_t.

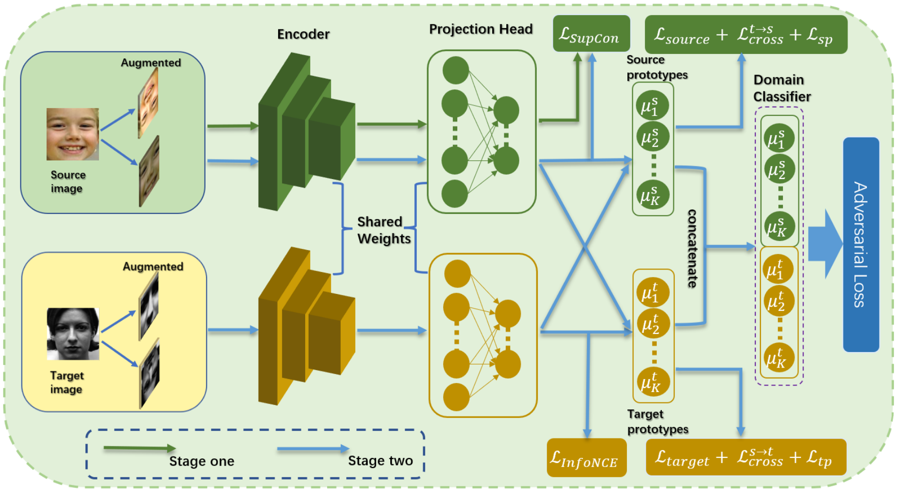


Fig. 1. Overview of the proposed Prototype-Oriented Contrastive Adaptation Network (POCAN) structure. POCAN contains two stages: **(1)pre-train on source domain**: using supervised contrastive learning, an encoder and a projection head are trained to learn semantically meaningful features and obtain good initial conditions for the target domain. **(2)transfer to target domain**: learning domain specific feature via L_{in}, while performing category-level alignment via L_{cross} and domain-level alignment via L_{ad}.

3.1 Learnable Category Prototypes

Most of the methods [12,31,24] using prototypes obtain the prototypes of the categories through the k-means clustering algorithm. However, this method consumes huge computing resources, and when the model parameters change, the prototypes of the categories cannot be updated in time, which affects the convergence speed and accuracy of the algorithm. Inspired by [23], we parameterize the prototypes so that they can be learned using the backpropagation algorithm. Specifically, we define the prototype as a vector $\mu \in \mathbb{R}^m$, where m is the dimension of each prototype that is the same as the embedding dimension of the sample output vector through the encoder and projection head. All prototypes form a prototype matrix $M = \{\mu_1, \mu_2, ..., \mu_k\}$, where k is the number of categories. The prototype matrix is closely connected to the linear classification layer because the neural network weights in the classification layer can be interpreted as class prototypes. Given a sample x_i with label y_i, we compute its feature vector $z_i = g(f(x_i))$, then we compute similarity matirx vector between z_i and M as $P_i = [p_{i,1}, p_{i,2}, ..., p_{i,k}]$, with $P_{i,j} = \frac{exp(\mu_j \cdot z_i)}{\sum_{r=1}^{k} exp(\mu_r \cdot z_i)}$. We learn source and target prototypes by minimize cross entropy between P and true label using:

$$\mathcal{L}_{sp} = \frac{1}{N_s} \sum_{i=1}^{N_s} w_{y_i^s}^s \cdot \mathcal{L}_{CE}(P_i^s, y_i^s), \tag{2}$$

$$\mathcal{L}_{tp} = \frac{1}{N_t^{trust}} \sum_{i=1}^{N_t^{trust}} w_{y_i^t}^t \cdot \mathcal{L}_{CE}(P_i^t, y_i^t), \tag{3}$$

where w is class weight, its calculation method is in Section 3.5. Because the target domain data has no label information, we first generate pseudo-labels(we use hard labels) for the target domain. It is worth noting that we only generate pseudo-labels for samples whose confidence is higher than a certain threshold, denoted as $D_t^{trust} = \{x_i\}_{i=1}^{N_t^{trust}} (x_i \in D_t, confidence(x_i) > threshold_i)$. And the initial prototypes of the target domain are initialized to the prototypes of the source domain. We train the extractor and prototypes in turns, in the same fashion as by performing iterative clustering and representation learning. We freeze the parameters of the prototypes when training the extractor, and freeze the parameters of the extractor when training the prototypes.

3.2 Pre-train On Source Domain

In the traditional field of fully supervised learning, the cross-entropy loss function is usually used to train the model. Recently [11] introduced contrastive learning into the field of supervised learning and proposed supervised contrastive learning, and obtained better results. Compared with traditional supervised learning based on cross-entropy, supervised contrastive learning is more stable to hyperparameters such as optimizers and data augmentation, and has better robustness to data corruption and training data reduction, and is inherently capable of hard positive and negative mining attributes [11]. Based on these advantages, we choose supervised contrastive learning for pre-training, and the experimental results also show the correctness of our choice.

The core idea of supervised contrastive learning is to shorten the distance between samples belonging to the same category and to widen the distance between samples of different categories. Its loss takes the following form:

$$\mathcal{L}_{SupCon} = \sum_{i=1}^{n} \frac{-1}{|P(i)|} \sum_{p \in P(i)} log \frac{exp(sim(v_i \cdot v_p)/\tau')}{\sum_{a \in A(i)} exp(sim(v_i \cdot v_a)/\tau')}, \tag{4}$$

where A(i) denotes the set of samples in a minibatch, $P(i) \equiv \{p \in A(i) : \widetilde{y}_p = \widetilde{y}_i\}$ indicates the set of samples share the same labels with the anchor in a minibatch. After pre-training, we can use the method in Section 3.1 to get the prototype matrix of the source domain M_s.

3.3 Domain-Specific Feature Learning

The goal of this part is to learn the discriminative features of each domain. Inspired by self-supervised contrastive learning using instance discrimination in [3] to learn knowledge from unlabeled data, we use it to learn the features of the target domain. However, instance discrimination-based self-supervised contrastive learning treats samples from different instances as negative sample pairs without considering the semantic similarity of the samples, resulting in the learned features lacking high-level semantic structure. Inspired by some algorithms that learn semantic structure through clustering, we learn the semantic structure of a category by keeping a sample close to its corresponding prototype while staying away from other prototypes. This is exactly what prototype contrastive learning does. So the domain-specific loss in target domain can be written as:

$$\mathcal{L}_{target} = \frac{1}{N_t} \sum_{i=1}^{N_t} L_{InfoNCE}(z_i^t) + \lambda_t \cdot \frac{1}{N_t^{trust}} \sum_{j=1}^{N_t^{trust}} w_{y_j^t}^t \cdot \mathcal{L}_{CE}(P_j^t, y_j^t), \tag{5}$$

where w is class weight, Its calculation method is in Section 3.5. λ_t is a weight from 0 to 1 to suppress the noisy signal from the pseudo-label of target data. Specifically, we set $\lambda_t = \frac{2}{1+exp(-10p)} - 1$, where p is the training progress linearly changing from 0 to 1.

Following the suggestion of work [21], we use both supervised contrastive learning and prototype contrastive learning to learn source domain features to ensure that the model maintains a low error rate in the source domain. So the domain-specific loss in source domain can be written as:

$$\mathcal{L}_{source} = \frac{1}{N_s} \sum_{i=1}^{N_s} (\mathcal{L}_{SupCon}(z_i^s) + w_{y_i^s}^s \cdot \mathcal{L}_{CE}(P_i^s, y_i^s)). \tag{6}$$

And the loss for domain-specific feature learning is:

$$\mathcal{L}_{in} = \mathcal{L}_{target} + \mathcal{L}_{source}. \tag{7}$$

3.4 Cross-Domain Feature Fusion

Category-level Alignment. Many existing unsupervised domain adaptation methods use MMD to minimize domain distribution differences or use adversarial methods to maximize domain fusion to perform domain alignment. Both methods are performed at the domain level, however, domain distribution alignment does not guarantee alignment between categories. Performing category-level alignment can better align the features of the two domains. We consider that if the source domain and target domain are already aligned, the prototypes of the source domain should also have a good classification effect on the target domain, and vice versa, the prototypes of the target domain should also classify the source domain samples well. Based on the above assumptions, we perform category-level domain alignment by forcing samples to be close to the corresponding prototype of another domain and away from the prototypes of other categories.

In particular, given a source domain sample x_i^s and target domain prototype matrix $M_t = \{\mu_j^t\}_{j=1}^k$, we first compute the features of x_i^s Vector $z_i^s = g(f(x_i^s))$, and then calculate the similarity distribution between z_i^s and M_t, with $P_{ij}^{s \to t} = softmax(z_i^s \cdot M_t)$. Then we use the cross-entropy loss function to maximize the similarity of x_i^s with the corresponding prototype and minimize the similarity with other prototypes, which is:

$$\mathcal{L}_{cross}^{s\to t} = \sum_{i=1}^{N_s} w_{y_i^s}^s \cdot \mathcal{L}_{CE}(P_{ij}^{s\to t}, y_i^s). \tag{8}$$

Similarly,we can compute $\mathcal{L}_{cross}^{t\to s}$, and the final loss for category-level alignment is:

$$\mathcal{L}_{cross} = \mathcal{L}_{cross}^{s\to t} + \mathcal{L}_{cross}^{t\to s}. \tag{9}$$

Domain-Level Alignment. Adversarial domain adaptation methods usually include a discriminator to identify which domain the sample comes from. But in our method, we do not set an additional domain discriminator. Inspired by [28], we utilize the trainable prototypes of the source and target domains to form a domain discriminator. Specifically, given a sample x_i, we first obtain its feature vector $z_i = g(f(x_i))$ and then calculate the similarity between z_i and the prototype matrices of the two domains respectively: $sim(z_i, M_s) = z_i \cdot M_s, sim(z_i, M_t) = z_i \cdot M_t$, Then we concatenate $sim(z_i, M_s)$ and $sim(z_i, M_t)$ and apply softmax to form $P^{st} = softmax([sim(z_i, M_s), sim(z_i, M_t)])$. Define $\alpha_i^s = \sum_{i=1}^{k} P^{st}$ as the score of sample belonging to the source domain and $\alpha_i^t = \sum_{i=k+1}^{2k} P^{st}$ as the score of sample belonging to target domain. For a sample of the source domain, α_i^s should be greater than α_i^t. Similarly, For the target domain sample, α_i^t should be greater than α_i^s. The domain discriminator should try to distinguish whether the samples are from the source domain or the target domain. Therefore, we train the domain discriminator with

$$\mathcal{L}_{dis} = -\frac{1}{N_s}\sum_{j=1}^{N_s} log(\sum_{i=1}^{k} P_j^{st}(x_j^s)) - \frac{1}{N_t}\sum_{j=1}^{N_t} log(\sum_{i=k+1}^{2k} P_j^{st}(x_j^t)). \tag{10}$$

According to the idea of an adversarial network, the feature extractor is designed to learn domain-invariant features, which does the opposite to the domain discriminator, so we optimize the feature extractor by the following formula to make it have an adversarial relationship with the domain discriminator.

$$\mathcal{L}_{st} = -\frac{1}{N_t}\sum_{j=1}^{N_t} log(\sum_{i=1}^{k} P_j^{st}(x_j^t)) - \frac{1}{N_s}\sum_{j=1}^{N_s} log(\sum_{i=k+1}^{2k} P_j^{st}(x_j^s)). \tag{11}$$

So the adversarial loss for domain-level alignment is:

$$\mathcal{L}_{ad} = \mathcal{L}_{dis} + \mathcal{L}_{st}. \tag{12}$$

3.5 Obtain class weights and a adaptative threshold

Class imbalance is a common problem in facial expression datasets, and understandably: it is much easier to collect a happiness or sadness expression than an anger or disgust expression. Some previous works [13,15] set a learnable class-wise weighting parameter to explore the class distribution of the target domain to match the distribution of the source domain. We believe that it is inappropriate to simply make the distribution of the target domain match the distribution of the source domain, because the distribution of the source domain may itself be unbalanced, the impact of imbalance is that the classifier tends to discriminate the samples as those expressions that are easy to recognize. If the distribution of the target domain matches the distribution of the source domain, this classifier bias will be transmitted to the target domain. So we set the category weights of the two domains separately so that they are both on a relatively balanced distribution. Specifically, we calculate the class weights in the following way:

$$w_i = 1 - (\frac{n_i}{\sum_{j=1}^{k} n_j})^2, \tag{13}$$

where n_i is the number of category i.

When generating pseudo-labels for the target domain, we take samples with label confidence greater than a certain threshold as trust samples, and use these samples to train prototypes. How to set an appropriate threshold is a relatively difficult matter, because different datasets have different degrees of difficulty in identification. For the more difficult datasets, we hope to set a lower threshold to ensure that the trust samples can occupy a certain proportion. Moreover, for different categories in a dataset, the samples that are easier to identify will always get a higher confidence level, and the difficult categories have a low confidence level, which leads to a small proportion of difficult samples in the trust sample set, which is not conducive to training prototypes. Considering these we set up an adaptive threshold generation method, specifically, for the category i of target domain D_t, its threshold is set as

$$threshold_i = avgc_i + (maxc_i - avgc_i) * (1 - avgc_i^{\beta}), \tag{14}$$

where $avgc_i$ is average confidence of samples judged to be category i, $maxc_i$ is the maximum confidence of the sample that is judged to be category i. β a is a hyperparameter that trades off pseudo-label noise and the number of trusted samples, a larger value indicates a stricter threshold policy.

3.6 Overall Objective

In the second stage, we update the four modules shown in Fig.1: encoder $f(\cdot)$, profection head $g(\cdot)$, source prototypes matrix M_s and target prototypes matrix M_t. Based on the above, we have the following training objective for POCAN :

$$\underset{f,g}{min} \mathcal{L}_{in} + \mathcal{L}_{cross} + \mathcal{L}_{st}. \tag{15}$$

$$\underset{M_s,M_t}{min} \mathcal{L}_{sp} + \mathcal{L}_{tp} + \mathcal{L}_{dis}. \tag{16}$$

We iteratively take turns training the feature extractor and prototype matrixes. First, we train the encoder and the projection head via Eq. (15), and second, we update the source and target domain prototype matrix via Eq. (16).

4 Experiments

4.1 Databases

CK+ [16]: The Extended CohnKanade (CK+) dataset is a lab-controlled dataset that consists of 593 video sequences from 123 subjects. Only 309 sequences are labeled with six basic expression labels based on the Facial Action Coding System (FACS). Following previous work [2], we select the three frames with peak formation and the first frame(neutral face) from 309 sequences, resulting in 1,236 images. The dataset is divided into a training set of 1,125 and a test set of 129 images.

JAFFE [17]: The Japanese Female Facial Expression (JAFFE) database is a laboratory-controlled dataset that contains 213 samples of posed expressions from 10 Japanese females. Each person has 3-4 images annotated with one of the six basic expressions and one image annotated with a neutral expression. This database is challenging because it is a highly biased dataset in terms of gender and ethnicity. All images were used in our experiments.

Oulu-CASIA [33]: The Oulu-CASIA is a lab-controlled facial expression dataset consisting of six expressions (except neutral) from 80 people between 23 and 58 years old. Similar to CK+, we select the last three frames with peak information and the first frame (neutral face) from the 480 videos with the VIS System under normal indoor illumination, resulting in 1,920 images.

FER2013 [7]: It is a large-scale and unconstrained dataset collected automatically by the Google image search API. It contains 35,887 gray images of size 48×48 pixels, and each image is annotated with seven basic expressions. The dataset is further divided into a training set of 28,709 images, a validation set of 3,589 images, and a test set of 3,589 images.

SFEW2.0 [5]: The Static Facial Expressions in the Wild(SFEW) 2.0 is a real-world facial expression dataset that was created by selecting static frames from different films with spontaneous expressions, various head pose, age range, occlusions, and illuminations. This challenging dataset is divided into three groups: 958 training sets, 436 validation sets, and 372 test sets. In our experiments, only the training set and validation set provided with labels is used.

RAF-DB [14]: The Real-World Affective Face Database(RAF-DB) is a real-world dataset collected from the Internet. With manually crowd-sourced annotation and reliable estimation, seven basic and eleven compound emotion labels are provided for the samples. The dataset contains 15,339 images from thousands of individuals and is divided into two groups (12,271 training samples and 3,068 testing samples) for evaluation.

4.2 Experimental Details

We use ResNet-50 pre-trained on the VGG-FACE2 [1] dataset without fully connected layers as the encoder. A two-layer MLP is used as the projection head to map the output of the encoder into a 256-dimensional vector. Both the encoder and projection head outputs are normalized using L2-normalizing. All images are resized to 112×112. In the first stage, we train for 300 epochs using stochastic gradient descent (SGD) with an initial learning rate of 0.001, a momentum of 0.9, and weight decay of 0.00001. We use cosine descent to adjust the learning rate. In the second stage, we alternately train the encoder, projection head, and prototypes for about 20 epochs. We use SGD with the same momentum and weight decay as the first stage. The learning rates for the feature extractor and projection head are set to 0.00001, and the learning rates for the source and target prototypes are initialized to 0.0001, divided by 10 after about 10 epochs.

4.3 Comparison Results

It is difficult to compare the results of cross-domain facial expression recognition across literature because the feature extractors used in different literature and the selection of source domains are different. Fortunately, the work [2] constructs a unified CD-FER evaluation benchmark by re-implementing some excellent cross-domain recognition algorithms in the literature, ensuring that these algorithms use the same source/target datasets and feature extractors to achieve fair CD-FER evaluation. Therefore, for the fairness of the comparison, according to the previous work [2], we use ResNet-50 as the encoder and RAF-DB as the source domain.

Table.1. report the results of cross-domain recognition on three lab-controlled datasets. Table.2. report the results of cross-domain recognition on two in-the-wild datasets. Note that DT refers to directly transferring the model to the target domain after pre-training on the source domain (that is, after the first stage). As shown in the tables, when using the same encoder and source domain datasets, our method achieves the best accuracy on JAFFE and the second-best results on CK+, FER2013 and SFEW2.0 datasets. There is only one dataset on which our method does not show better results: Oulu-CASIA. Only ECAN [15] conducts cross-domain recognition experiments on Oulu-CASIA, and it uses a different pre-trained model and source domain than ours. It has a DT accuracy of 59.39%, which is much higher than our DT accuracy 53.90%, which may be responsible for the difference in performance. It should be pointed out that AGRA [2] used landmark information as local features, and we do not use any additional feature information. Nonetheless, our method achieves competitive performance on both lab-controlled and in-the-wild datasets.

4.4 Empirical Analysis

Ablation Study. Our POCAN have three important loss : domain-specific feature learning $\mathcal{L}_{in}$, category-level alignment loss $\mathcal{L}_{cross}$ and adversarial loss for domain-level alinment $\mathcal{L}_{ad}$. Therefore , We examine the effectiveness of each component of our method by remove each loss while keeping other loss. We also detect the effect of class imbalance. The experimental results are shown in Table 3. From the results of ablation experiments we can see that both in-domain feature learning and domain alignment are important in our method.

Table 1. Cross-domain accuracy(%) comparison on CK+, JAFFE and Oulu-CASIA dataset.

Method	Source	Backbone	CK+	JAFFE	Oulu-CASIA
STCNN [8]	MMI+FERA	Inception-ResNet	73.91	-	-
GDFER [19]	*Sixdatasets*†	Inception	64.20	-	-
ICID [10]	RAF-DB+MMI+SFEW	DarkNet-19	88.7	-	-
ICID [10]	RAF-DB+MMI	DarkNet-19	84.5	-	-
DFA [35]	CK+	Customized	-	63.38	-
FTDNN [32]	*Sixdatasets*⋆	VGGNet	88.58	44.32	-
DETN [13]	RAF-DB	Customized	78.83	57.75	-
ECAN [15]	RAF-DB 2.0	VGGNet	86.49	61.94	**63.97**
AGRA [2]	RAF-DB	ResNet-50	**85.27**	61.50	-
ICID [10]	RAF-DB	ResNet-50	74.42	50.70	-
DFA [35]	RAF-DB	ResNet-50	64.26	44.44	-
FTDNN [32]	RAF-DB	ResNet-50	79.07	52.11	-
DETN [13]	RAF-DB	ResNet-50	78.22	55.89	-
ECAN [15]	RAF-DB	ResNet-50	79.77	57.28	-
DT	RAF-DB	ResNet-50	76.74	52.11	53.90
POCAN(Ours)	RAF-DB	ResNet-50	84.50	**64.32**	55.83

The upper part is taken from the corresponding papers, the middle part is taken from [2], the bottom part are generated by our implementation.

† MultiPIE, CK+, DISFA, FERA, SFEW, and FER2013.

⋆ CK+, JAFFE, MMI, RaFD, KDEF, BU3DFE, ARFace.

Table 2. Cross-domain accuracy(%) comparison on FER2013 and SFEW2.0 dataset.

Method	Source	Backbone	FER2013	SFEW2.0
GDFER [19]	*Sixdatasets*†	Inception	34.00	39.80
DETN [13]	RAF-DB	Customized	52.37	47.55
ECAN [15]	RAF-DB 2.0	VGGNet	58.21	54.34
AGRA [2]	RAF-DB	ResNet-50	**58.95**	**56.43**
ICID [10]	RAF-DB	ResNet-50	53.70	48.85
DFA [35]	RAF-DB	ResNet-50	45.79	43.07
FTDNN [32]	RAF-DB	ResNet-50	55.98	47.48
DETN [13]	RAF-DB	ResNet-50	52.29	49.40
ECAN [15]	RAF-DB	ResNet-50	56.46	52.29
DT	RAF-DB	ResNet-50	54.73	47.49
POCAN(Ours)	RAF-DB	ResNet-50	58.87	53.44

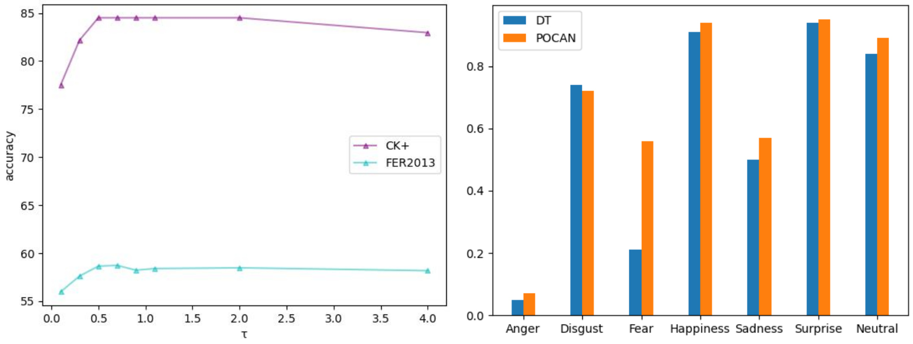
The upper part is taken from the corresponding papers,the middle part is taken from [2],the bottom part are generated by our implementation.

† MultiPIE, CK+, DISFA, FERA, MMI, and FER2013 or SFEW.

Table 3. Ablation Study of the role of each loss function and category weights in our method.

Method	CK+	JAFFE	Oulu	FER2013	SFEW2.0	AVG
POCAN(w/o $\mathcal{L}_{in}$)	80.62	57.75	55.26	57.70	51.65	60.60
POCAN(w/o $\mathcal{L}_{cross}$)	76.74	58.21	52.18	58.34	53.30	59.75
POCAN(w/o $\mathcal{L}_{ad}$)	83.72	61.97	55.78	58.76	53.37	62.72
POCAN(w/o weights)	83.72	60.56	55.68	**58.87**	52.37	62.24
POCAN	**84.50**	**64.32**	**55.83**	58.74	**53.44**	**63.37**

Just doing domain alignment or in-domain feature learning yields mediocre results. Category-level alignment is more important than domain-level alignment, which contributes little to the improvement of our experimental results. We can also find that category-wise weights are beneficial to our approach.



(a) (b)

Fig. 2. (a) Performances w.r.t τ on CK+ and FER2013. (b) CK+ F1 scores after the first stage and the second stage

Parameter Analysis. The hyperparameter τ has an important impact on feature learning in the target domain. To examine the effect of this parameter in our experiments, we show the accuracy of our method on the lab-controlled dataset CK+ and in-the-wild dataset FER2013 under different τ. As shown in Fig.2(a), our method benefits from a relatively large temperature parameter, and 0.5-0.7 is a suitable range. A extremely low temperature will seriously degrade the performance of the model, and too high one will also degrade the performance of the model. The smaller the temperature is, the more the loss function pays attention to the difficult negative samples. The larger the temperature, the more

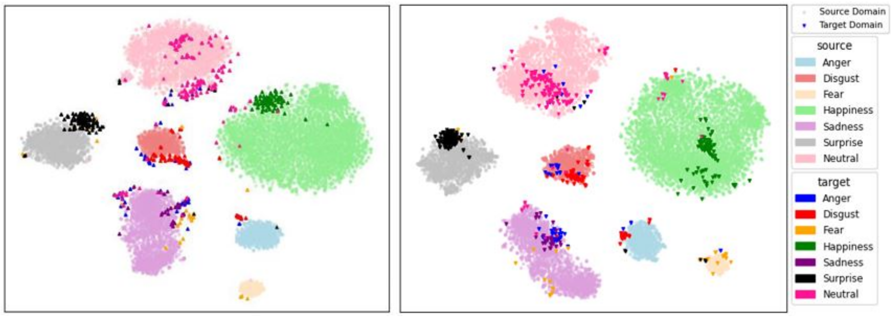


Fig. 3. The t-SNE visualization of the feature representations learned by our proposed method after the first stage (left) and the second stage (right) on CK+.

evenly the loss function pays attention to negative samples. Difficult negative samples are likely to be samples of the same category as the anchor and too much attention to these samples will destroy the high-level class semantics learned by the model. Therefore the temperature is a parameter of a trade-off between learning new features and retaining already learned features.

Visualization. We utilize t-SNE [18] to visualize the feature representations of different domains at different training stages. As shown in Fig.3, after the first stage, there is a difference in the distributions of the source and target domains, and after domain adaptation, the distributions of the same classes in different domains become closer, and the distribution of different categories in the same domain is more separated. And Fig.2(b) shows the specific changes in the accuracy of each category. The effectiveness of our method is demonstrated intuitively.

5 Conclusion

In this paper, we propose POCAN for unsupervised cross-domain facial expression recognition research. We parameterize the prototypes to make them learnable variables, reducing the computational effort of traditional prototype computing methods. We carry out a two-stage training approach, wherein in the first stage we pre-train on the source domain to acquire semantically meaningful features, and in the second stage, we simultaneously perform domain-specific feature learning and two levels of domain alignment. Moreover, the problem of category imbalance in facial expression datasets is considered and we propose an adaptive threshold method to select trusted target samples. We conduct extensive experiments on both lab-controlled and in-the-wild datasets, and our method exhibits competitive results.

Acknowledgements This work was supported in part by the NSF of china (Grant No. 61773215).

References

1. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: Vggface2: A dataset for recognising faces across pose and age. In: 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018). pp. 67–74. IEEE (2018)
2. Chen, T., Pu, T., Wu, H., Xie, Y., Liu, L., Lin, L.: Cross-domain facial expression recognition: A unified evaluation benchmark and adversarial graph learning. IEEE transactions on pattern analysis and machine intelligence (2021)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
4. Chuang, C.Y., Robinson, J., Lin, Y.C., Torralba, A., Jegelka, S.: Debiased contrastive learning. Advances in neural information processing systems **33**, 8765–8775 (2020)
5. Dhall, A., Goecke, R., Lucey, S., Gedeon, T.: Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). pp. 2106–2112. IEEE (2011)
6. Ekman, P., Friesen, W.V.: Constants across cultures in the face and emotion. Journal of personality and social psychology **17**(2), 124 (1971)
7. Goodfellow, I.J., Erhan, D., Carrier, P.L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.H., et al.: Challenges in representation learning: A report on three machine learning contests. In: International conference on neural information processing. pp. 117–124. Springer (2013)
8. Hasani, B., Mahoor, M.H.: Spatio-temporal facial expression recognition using convolutional neural networks and conditional random fields. In: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). pp. 790–795. IEEE (2017)
9. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
10. Ji, Y., Hu, Y., Yang, Y., Shen, F., Shen, H.T.: Cross-domain facial expression recognition via an intra-category common feature and inter-category distinction feature fusion network. Neurocomputing **333**, 231–239 (2019)
11. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. Advances in Neural Information Processing Systems **33**, 18661–18673 (2020)
12. Li, J., Zhou, P., Xiong, C., Hoi, S.C.: Prototypical contrastive learning of unsupervised representations. arXiv preprint arXiv:2005.04966 (2020)
13. Li, S., Deng, W.: Deep emotion transfer network for cross-database facial expression recognition. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 3092–3099. IEEE (2018)
14. Li, S., Deng, W.: Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition. IEEE Transactions on Image Processing **28**(1), 356–370 (2018)
15. Li, S., Deng, W.: A deeper look at facial expression dataset bias. IEEE Transactions on Affective Computing (2020)
16. Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In: 2010 ieee computer society conference on computer vision and pattern recognition-workshops. pp. 94–101. IEEE (2010)

17. Lyons, M.J., Akamatsu, S., Kamachi, M., Gyoba, J., Budynek, J.: The japanese female facial expression (jaffe) database. In: Proceedings of third international conference on automatic face and gesture recognition. pp. 14–16 (1998)
18. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
19. Mollahosseini, A., Chan, D., Mahoor, M.H.: Going deeper in facial expression recognition using deep neural networks. In: 2016 IEEE Winter conference on applications of computer vision (WACV). pp. 1–10. IEEE (2016)
20. Van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv e-prints pp. arXiv–1807 (2018)
21. Qiu, Z., Zhang, Y., Lin, H., Niu, S., Liu, Y., Du, Q., Tan, M.: Source-free domain adaptation via avatar prototype generation and adaptation. arXiv preprint arXiv:2106.15326 (2021)
22. Shan, C., Gong, S., McOwan, P.W.: Facial expression recognition based on local binary patterns: A comprehensive study. Image and vision Computing **27**(6), 803–816 (2009)
23. Tanwisuth, K., Fan, X., Zheng, H., Zhang, S., Zhang, H., Chen, B., Zhou, M.: A prototype-oriented framework for unsupervised domain adaptation. Advances in Neural Information Processing Systems **34**, 17194–17208 (2021)
24. Tian, L., Tang, Y., Hu, L., Ren, Z., Zhang, W.: Domain adaptation by class centroid matching and local manifold self-learning. IEEE Transactions on Image Processing **29**, 9703–9718 (2020)
25. Wang, F., Liu, H.: Understanding the behaviour of contrastive loss. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2495–2504 (2021)
26. Wang, X., Wang, X., Ni, Y.: Unsupervised domain adaptation for facial expression recognition using generative adversarial networks. Computational intelligence and neuroscience **2018** (2018)
27. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3733–3742 (2018)
28. Xia, H., Zhao, H., Ding, Z.: Adaptive adversarial network for source-free domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9010–9019 (2021)
29. Yan, H.: Transfer subspace learning for cross-dataset facial expression recognition. Neurocomputing **208**, 165–173 (2016)
30. Yan, K., Zheng, W., Cui, Z., Zong, Y.: Cross-database facial expression recognition via unsupervised domain adaptive dictionary learning. In: International Conference on Neural Information Processing. pp. 427–434. Springer (2016)
31. 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. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13834–13844 (2021)
32. Zavarez, M.V., Berriel, R.F., Oliveira-Santos, T.: Cross-database facial expression recognition based on fine-tuned deep convolutional network. In: 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). pp. 405–412. IEEE (2017)
33. Zhao, G., Huang, X., Taini, M., Li, S.Z., PietikäInen, M.: Facial expression recognition from near-infrared videos. Image and Vision Computing **29**(9), 607–619 (2011)

34. Zhong, L., Liu, Q., Yang, P., Liu, B., Huang, J., Metaxas, D.N.: Learning active facial patches for expression analysis. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2562–2569. IEEE (2012)
35. Zhu, R., Sang, G., Zhao, Q.: Discriminative feature adaptation for cross-domain facial expression recognition. In: 2016 International Conference on Biometrics (ICB). pp. 1–7. IEEE (2016)