Multi-Source Unsupervised Domain Adaptation via Feature Disentanglement and Fusion

Chengrong Yang
School of Information Science
and Engineering,Yunnan University,
Yunnan Key Laboratory of Software
Engineering, Kunming, China
yangchengrong@mail.ynu.edu.cn

Qiwen Jin
The National Pilot School
of Software,Yunnan University,
Yunnan Key Laboratory of Software
Engineering, Kunming, China
12022219114@mail.ynu.edu.cn

Xiaoguo Zhang
Institute of International Rivers
and Eco - Security,Yunnan University,
Kunming, China
zhangxiaoguo1019@hotmail.com

Yujue Zhou
The National Pilot School
of Software,Yunnan University,
Yunnan Key Laboratory of Software
Engineering,Kunming, China
yujue_zhou@163.com

Abstract

Multi-source domain adaptation aims to transfer knowledge from multiple source domains with varying data distributions to a target domain. However, most previous methods have overlooked the unique characteristics of the target domain, leading to a decline in the distinguishability of target domain features. In this paper, we propose a Feature Disentanglement and Fusion Model, termed as FDFM, which effectively leverages target-domain-specific features. Specifically, FDFM comprises a Feature Disentanglement block (FD) and a Feature Fusion block (FF). FD disentangles features into domain-invariant components and target-domain-specific components. FF integrates these two components to obtain more discriminative feature representations in the target domain. Furthermore, we design two classifiers: one for learning from domain-invariant features and another for learning from fused features with pseudo-labels. The final prediction results are obtained by integrating the outputs of all classifiers. We conduct extensive experiments on four popular transfer learning benchmark datasets, demonstrating that FDFM outperforms other state-of-the-art methods. For instance, we achieve an average accuracy improvement from 74.4% to 76.6% on the OFFICE-HOME dataset.

Keywords: Mutil-Source Domain Adaptation, Domain-Specific Feature, Feature Fusion, Feature Disentanglement.

1. Introduction

The inherent variability in data distribution poses a significant obstacle to the pursuit of general artificial intelligence. Specifically, artificial intelligence models struggle to achieve consistent high performance, particularly when confronted in scenarios involving data dispersed across two domains: the source domain and the target domain, each characterized by distinct distributions. Here, the source domain represents the training set, while the test set is denoted as the target domain[1]. While circumventing this challenge is possible when the data in the target domain is well-labeled by the data provider, real-world scenarios often present the limitation of having only annotated data in the source domain. Consequently, the training process across these two domains becomes intractable. In response, Domain Adaptation emerges as a viable solution, proposed by pioneering researchers to enhance the model's performance on the target domain[2]. However, as the number of data sources increases, single-domain adaptation reveals its limitations in effectively addressing these challenges[3]. Consequently, researchers have introduced multi-source adaptation as a strategy to bolster the model's generalization capability across diverse data sources with varying distributions[4, 5, 6].

To address the challenges arising from differences in data distribution across multiple sources, several multi-source domain adaptation methods have been introduced, achieving notable performance milestones[7]. The fundamental concept of domain adaptation involves aligning the distinct data distributions of source and target domains, thereby imbuing the data with domain-invariant representations. These representations, as suggested by Zhao et al.[8], Amosy et al.[9], and Hoffman et al.[10], encapsulate domain-invariant features in both the target and source domains. Researchers anticipate that these domain-invariant features, enriched with discernible characteristics from diverse sources, will enhance model performance. In pursuit of this objective, researchers have proposed a range of methods for aligning features in multi-source domains. Zhu et al.[11] introduced the Multi-Feature Space Alignment Network (MFSAN), a two-stage alignment method. Recognizing the difficulty of establishing a common feature space for all domains as the number of source domains increases, MFSAN first aligns the target domain with each individual source domain in the initial stage. Subsequently, in the second stage, MFSAN aligns the results from all classifiers. Another noteworthy approach is the Multi-Source Contribution Learning for Domain Adaptation (MSCLDA) presented by Li et al.[12] MSCLDA takes into account the varying contributions of model performance across different source domains, weighting the classifier based on the dissimilarities between the source and target domains.

Revisiting the concept of domain adaptation, it involves the acquisition of domain-invariant features in both the source and target domains, facilitating the training of classifiers capable of generalizing across both domains[13, 14]. Traditional domain adaptation methods typically achieve domain-invariant features by eliminating domain-specific features[15, 16]. While extracting domain-invariant features is an effective means of enhancing the model's performance on the target domain, this approach comes with the drawback of discarding distinctive features inherent to the target and source domains. Self-supervised learning and contrastive learning methods have demonstrated the value of these variant features by maximizing the dissimilarity between features of different samples[17, 18, 19]. These variant features harbor a wealth of information conducive to categorization. In this paper, we term these variant features specific to the target domain as domain-specific features.

Domain-specific features not only contain valuable information for tasks but also contribute to the discriminatory power of features. Their removal results in a reduction of feature distinctiveness. Importantly, the presence of domain-invariant features diminishes with an increasing number of source domains[11]. Establishing a suitable common feature space is not only highly challenging but also leads to a decrease in the discriminative capability of the extracted features. Consequently, the elimination of domain-specific features in multi-source domain adaptation induces a more pronounced performance decline compared to single-source domain adaptation. The incorporation of domain-specific features can enhance feature distinctiveness to a certain extent. However, prior methods have largely overlooked the significance of domain-specific features, consequently falling short of achieving optimal performance.

This paper introduces a novel feature disentanglement and fusion model designed to effectively incorporate target domain-specific features and enhance feature discriminability. Our model comprises three key components. Firstly, the feature extraction module employs a public feature extractor, initially extracting features. Subsequently, it separates domain-invariant features and domain-specific features by combining the features through both a common feature extraction network and a specific feature extraction network. The second component is the feature fusion module, where domain-invariant features from the source domain are combined with specific features from the target domain to produce a more discriminative feature representation in the target domain. Finally, the classification module employs pseudo-labels to guide the classifier in learning from the fusion features, resulting in robust classification outcomes. Additionally, we integrate all classifiers to derive the final predicted labels.

The primary contributions of this paper can be delineated as follows:

- We introduce an innovative multi-source domain adaptation model that utilizes feature disentanglement to effectively separate domain-specific and domain-invariant features.
- We employ an unsupervised fusion feature mechanism to acquire a more discriminative feature representation. Notably, our model demonstrates effective classifier training even in the absence of target domain labels.
- We have conducted comprehensive experiments on four benchmark datasets, and the simulation results demonstrate that our model achieves state-of-the-art performance.

The structure of this paper is organized as follows. Section 2 provides a review of related work pertinent to the topics discussed herein. In Section 3, the proposed method is elucidated in detail. Section 4 presents and analyzes the simulation experimental results of our model across three benchmark datasets. Finally, in Section 5, we summarize our model and offer a discussion on potential avenues for future research.

2. Related Works

In this section, we delve into related research, specifically focusing on single-source unsupervised domain adaptation methods (SUDA) and multi-source unsupervised domain adaptation methods (MUDA).

2.1. Sigle-source domain adaptation

In recent years, single-source unsupervised domain adaptation has emerged as a prominent solution for mitigating data distribution disparities across domains. Depending on the settings, two predominant approaches characterize SUDA. The first is Max Mean Discrepancy (MMD)-based domain adaptation[20, 21, 22, 23], and the second involves is adversarial network-based domain adaptation[24, 25, 26, 27].

MMD-based approaches extend deep neural networks for domain adaptation by incorporating adaptation layers that align the mean embedding of distributions. The introduction of MMD into domain adaptation was pioneered by Tzeng et al.[28], who computed the MMD loss between source and target domains before the last fully connected layer, co-optimizing it with the classification loss. Long et al.[29] further extended MMD with the introduction of the Deep Adaptation Network (DAN), utilizing Multi-Kernel MMD (MK-MMD). MK-MMD employs multiple kernels on the foundation of MMD to adapt the embedding of source and target domains. Conversely, Zhu et al.[30] considered both global and local domain alignment in the adaptation process, proposing the Deep Subdomain Adaptation Network (DSAN) based on Local Maximum Mean Discrepancy (LMMD). LMMD accommodates class-level distributional alignment without necessitating specific training.

The adversarial network-based domain adaptation approach trains networks by incorporating a discriminator into the model. Precisely, the discriminator guides the feature extractor to extract domain-invariant features by determining whether features originate from the source or target domain, and both components are trained adversarially. Ganin et al.[24] introduced a discriminator in deep networks, employing adversarial training to train the network and the discriminator concurrently. The discriminator assesses whether the distributions of features extracted by the network are equivalent, resulting in features with similar distributions in the source and target domains. Similar to DSAN, Wang et al.[31] simultaneously align global and local feature distributions through an adversarial approach. Chen et al.[32] introduced Batch Spectral Penization (BSP) into adversarial adaptation to mitigate the discriminative reduction of features induced by adversarial training.

Despite the remarkable achievements of previous single-source domain adaptation methods, they exhibit limitations when confronted with multiple data sources.

2.2. Multi-source domain adaptation

As the ability to collect information grows, data emanates from diverse sources. Single-source domain adaptation algorithms fall short of meeting the realistic demands posed by this diversity. Consequently, multi-source domain adaptation algorithms emerge as a promising technique[33, 34, 35].

Multi-source domain adaptation leverages various techniques such as ensemble learning, attention mechanisms, etc., to effectively harness information from multiple source domain data[36, 37, 38]. Deng et al.[39] introduced the attention mechanism into multi-source domain adaptation, directing the model to focus on features with transferability. Xu et al.[40] employed a graphical structure to jointly model relationships between instances/categories for multiple source domains and target domains, proposing both Conditional Random Field (CRF-MSDA) and Markov Random Field (MRF-MSDA). Concerning the aspect of multi-source domain adaptive privacy-preserving policy, Feng et al.[41] achieved domain adaptation through knowledge distillation of models from different source domains. Mancini et al.[42] addressed the limitations and inconsistencies of traditional methods by introducing graphs into multi-source domain adaptation, enabling both predictive domain adaptation and continuous domain adaptation.

Despite the comprehensive exploration of multi-source domain adaptation in the aforementioned methods, there has been a notable neglect of target domain-specific features, which have the potential to significantly enhance the discriminative power of features.

3. Method

In this section, we articulate the problem definition and elucidate the constituents of our model. Our model comprises three key components outlined as follows: 1) Base feature extractor; 2) Domain-invariant/specific feature extractor and feature fusion; and 3) Classifier and ensemble, as depicted in Fig. 1. The role of the Base feature extractor is to extract the base features of samples, encompassing both domain-invariant and domain-specific features, thereby alleviating the challenge of identifying a potential common feature space for source and target domains. In the Domain-invariant/specific feature extractor and feature fusion stage, our model extracts target domain-specific features and all domain-invariant features. The fusion of these two features is geared towards providing an enhanced feature representation for the final classification stage. In the Classification and ensemble stage, we classify samples based on the feature representations obtained in the prior stage and employ ensemble techniques on classifiers trained across multiple source domains to enhance the final classification results.

3.1. Problem Statement

In the MUDA setting, our objective is to integrate information from multiple source domains to enhance the model's performance on the target domain. We are provided with access to data X_t from the target domain D_t, but not to the corresponding labels. Denoting $D_t = \{(X_t)\} = \{(x_i^t)\}_{i=1}^{n_t}$, where n_t is the number of samples in the tar-

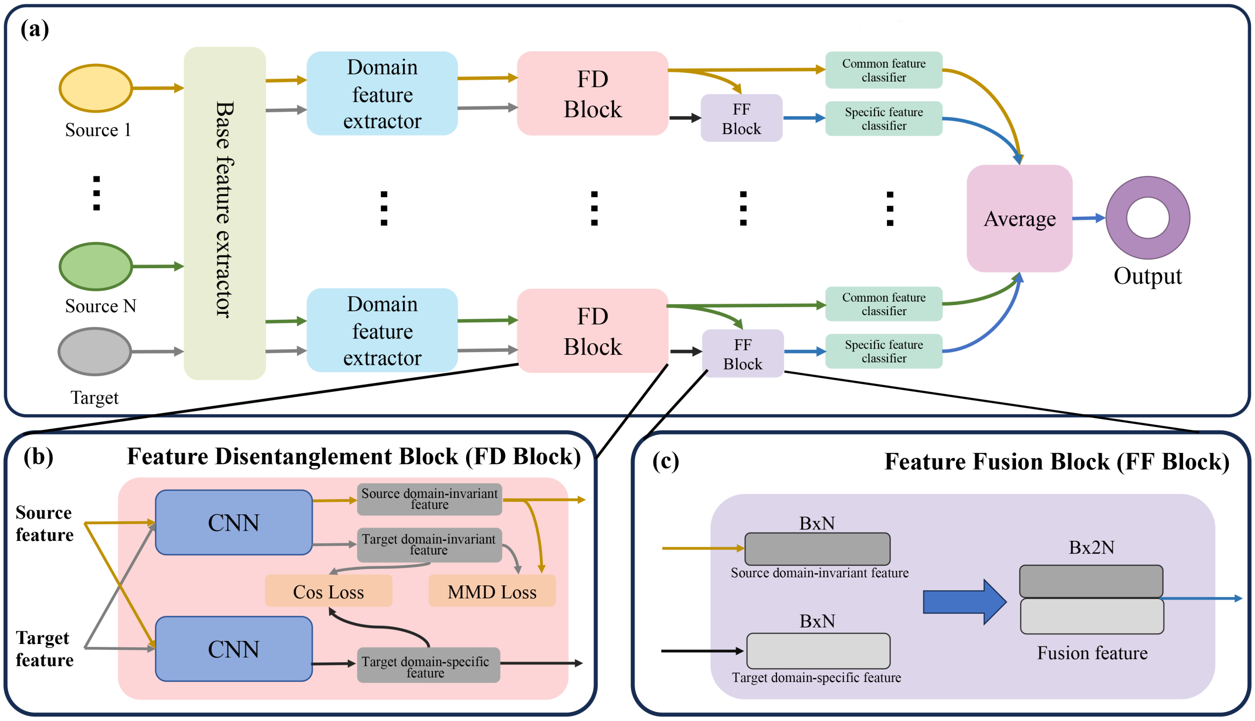
(a)
Source 1
Source N
Target
Base feature extractor
Domain feature extractor
Domain feature extractor
FD Block
FD Block
FF Block
FF Block
Common feature classifier
Specific feature classifier
Common feature classifier
Specific feature classifier
Average
Output
(b)
Feature Disentanglement Block (FD Block)
Source feature
Target feature
CNN
CNN
Source domain-invariant feature
Target domain-invariant feature
Cos Loss
MMD Loss
Target domain-specific feature
(c)
Feature Fusion Block (FF Block)
BxN
Source domain-invariant feature
BxN
Target domain-specific feature
Bx2N
Fusion feature

Figure 1. The overall of our model.

get domain. For all K source domains $\{D_{s_k}\}_{k=1}^{K}$, we have complete access to their labels Y_{s_k} and data X_{s_k}, denoted as $D_{s_k} = \{(Y_{s_k}, X_{s_k})\} = \{(x_i^{s_k}, y_i^{s_k})\}_{i=1}^{n_{s_k}}$, where n_{s_k} is the number of samples in the k-th source domain. Let the feature space be denoted as $\mathcal{X}_{s_k} = \mathcal{X}_t$. The data distribution D_{s_k} and D_t exhibit discrepancies, namely $p \neq q$. The primary objective of the Feature Disentanglement and Fusion Model (FDFM) is to train a model $FDFM(\cdot)$ capable of combining domain-invariant and domain-specific features extracted from multi-source and target domains. This aims to mitigate the impact of data distribution differences on the model in the target domain.

3.2. Domain Base Feature Extractor

Seeking a common feature space for diverse data is not suitable, particularly as the variation between data distributions in different domains increases, a phenomenon exacerbated with a higher number of domains. The substantial disparities in data distribution render domain-invariant features sparse across all domains. In the pursuit of extracting domain-invariant features for all domains, the feature extractor tends to overlook a significant number of discriminative features. Consequently, not only is it challenging to identify a common feature space for all domains, but it also leads to the loss of discriminative features.

To circumvent this challenge, we employ a Domain Base Feature Extractor, which identifies a common feature space for each pair of source and target domains. By increasing the number of common feature spaces, we aim to alleviate the difficulty of finding a single common feature space and maximize the retention of discriminable features. The Domain Base Feature Extractor consists of two components: a Base Feature Extractor $f(\cdot)$ and multiple Domain Feature Extractors $h_j(\cdot)$. The base feature extractor is tasked with extracting low-level base features for all domains. Intuitively, using multiple base feature extractors for each pair of source and target domains seems beneficial. However, given that feature extractors are typically implemented using large deep networks, using a single base feature extractor for all domains effectively reduces computational resources. The process of feature extraction by the base feature extractor can be represented as $x^b = f(x)$, where x is the data from a source or target domain, x^b is the base feature extracted by the base feature extractor, and $f(\cdot)$ is the base feature extractor. Domain feature extractors aim to identify a suitable feature space for each pair of source and target domains to capture domain features containing both domain-invariant and domain-specific features. These domain feature extractors are designed as simple Convolutional Neural Networks (CNN) with fewer convolutional layers. The process of domain feature extraction by domain feature extractors can be represented as $x^d = h_k(x^b)$, where $h_k(\cdot)$ is the k-th domain feature extractor, and x^d is the k-th source or target domain feature.

3.3. Feature Disentanglement

The domain feature comprises both domain-invariant and domain-specific features. To effectively utilize the domain-specific features, it is crucial to separate these features, a process known as feature disentanglement. Feature disentanglement enhances the focus of each block in the network on its designated task. For this purpose, we introduce the Feature Disentanglement Block (FD), consisting of two simple CNNs as illustrated in Figure 1. The upper CNN is responsible for extracting domain-invariant features, denoted as the invariant-CNN, while the lower CNN extracts domain-specific features, known as the specific-CNN. The input to the invariant-CNN is the domain features of the source and target domains. The process can be represented as $x^{invariant} = ICNN(x^d)$, where $ICNN(\cdot)$ is the invariant-CNN, and $x^{invariant}$ is the domain-invariant feature. To measure the distance between domain distributions, we introduce Maximum Mean Discrepancy (MMD). MMD maps different domain sample feature distributions into the Reproducing Kernel Hilbert Space (RKHS) and computes their mean distances. By minimizing the mean embedding distance between them, we constrain the feature extractor to extract domain-invariant features. The calculation of MMD is expressed as follows:

$$d_{mmd}(p,q) \triangleq \left\|\mathbf{E}_{x^s\sim p}\left[\phi\left(x^s\right)\right] - \mathbf{E}_{x^t\sim q}\left[\phi\left(x^t\right)\right]\right\|_{\mathcal{H}}^2, \tag{1}$$

Here, $\mathcal{H}$ represents the RKHS equipped with a characteristic kernel, $kernel\left(X_s, X_t\right) =< \phi\left(X_s\right), \phi\left(X_t\right) >$, where $< \cdot, \cdot >$ denotes the inner product of vectors, $\phi(\cdot)$ signifies the mapping from a feature distribution to RKHS, and p and q are the distributions of x^s and x^t, respectively. Typically, the empirical estimate of MMD, $d_{mmd}(p,q)$, can be further factorized as follows:

$$\begin{aligned}\hat{d}_{mmd}(p,q) =& \left\| \frac{1}{n_s}\sum_{x_i\in\mathcal{D}_s}\phi\left(x_i\right) - \frac{1}{n_t}\sum_{x_j\in\mathcal{D}_t}\phi\left(x_j\right)\right\|_{\mathcal{H}}^2 \\ =& \frac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s} k\left(x_i^s, x_j^s\right) + \frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t} k\left(x_i^t, x_j^t\right)\cdot \\ & - \frac{2}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t} k\left(x_i^s, x_j^t\right)\end{aligned} \tag{2}$$

In summary, the MMD loss of our model can be calculated as follows:

$$\mathcal{L}_{mmd} = \sum_{j=1}^{K} \hat{d}_{mmd}(p,q)(f(h_j(x_{s_j})), f(h_j(x_t))) \tag{3}$$

For domain-specific features, we consider domain features to be composed of two parts: domain-invariant features and domain-specific features. In other words, domain-specific features can be obtained by removing domain-invariant features from domain features. To obtain domain-specific features from domain features, we aim to maximize the distance between domain-invariant features and domain-specific features. We utilize cosine similarity to measure this distance. Cosine similarity is a metric that calculates the angle between two vectors in a multidimensional space, serving as a measure of similarity between two feature vectors. When considering two feature vectors A and B, each feature vector can be conceptualized as a vector in a high-dimensional space, where each dimension represents a specific feature, and the value of that feature represents the value of that dimension. Consequently, if two eigenvectors exhibit similar values in the same dimension, these vectors will be less angular in the space, resulting in higher cosine similarity. This metric takes into account not only the values of the features in the feature vector but also the relationships between the features. It finds utility in various scenarios where the similarity of features needs to be compared. The cosine similarity is calculated as follows:

$$\cos(A,B) = \frac{A\cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4}$$

We employ specific-CNN to extract domain-specific features from the domain features by calculating the cosine similarity between the domain-specific features and the domain-invariant features of the target domain. To ensure that the specific-CNN efficiently extracts domain-specific features, we constrain it by maximizing the cosine similarity between these features. This process can be represented as $x_t^{specific} = SCNN(x_t^d)$, where $x_t^{specific}$ is the target domain-specific feature, x_t^d is the domain feature of the target domain, and $SCNN(\cdot)$ is the specific-CNN. The cosine similarity loss can be expressed as follows:

$$\mathcal{L}_{cos} = 1 - \cos(SCNN(x_t^d), ICNN(x_t^d)) \tag{5}$$

3.4. Feature Fusion and Classification

As we lack labels for the target domain data, direct utilization of target domain-specific features is not feasible. To fully leverage the target domain-specific features, we employ the Feature Fusion Block (FF) mechanism. Feature fusion mechanisms have demonstrated advantages in various image classification scenarios, combining features from different layers or branches to provide more robust features for the classifier. There are several main types of feature fusion mechanisms: 1) Concatenation: Multiple feature maps are concatenated in the depth dimension to create a more comprehensive representation of features. For example, in the encoder and decoder, low-level features in the encoder and high-level features in the decoder are concatenated, enhancing the perceptual ability of the decoder. 2) Summation: Summing involves adding multiple feature maps element-wise to obtain the average value of the features. 3) Multiplication: Multiplying multiple feature maps element-wise enhances the semantic information of the features while preserving detail information. 4) Attention Mechanism: The attention mechanism improves the response of important features by learning a set of weights to weight features at different scales. 5) Pyramid Pooling: Features at different scales are pyramid pooled to obtain feature representations at various scales, and more. Here, we opt for the simple yet effective concatenation method for feature fusion. Specifically, we concatenate the source domain-invariant features (size: BN) with the target domain-specific features (size: BN) to obtain fused features (size: $B2N$). At this point, the labels associated with the hybrid features transition from hard labels to soft labels.

In the soft labels of fused features, accurate labels for the source domain can be obtained, while the labels for the target domain are unavailable. To address this, we introduce another classifier, the Invariant Feature Classifier, to assist in obtaining the target domain label. The Invariant Feature Classifier is trained with data and labels from the source domain, and we use cross-entropy to optimize it. This can be expressed as:

$$\mathcal{L}_{invariant} = \mathcal{L}_{ce}(IFC\left(f(h_j\left(x^{s_k}\right))\right), y^{s_k}) \tag{6}$$

Here, $\mathcal{L}_{ce}$ denotes the cross-entropy loss, K is the number of source domains, and $IFC(\cdot)$ represents the Invariant Feature Classifier. Subsequently, we constrain the classifier (Specific Feature Classifier) to yield a classification result similar to the Invariant Feature Classifier. We use the L_1 loss loss to compute the similarity between them, expressed as:

$$\mathcal{L}_1 = L_1(SFC(x_t^{specific}), IFC(x_t^{invariant})) \tag{7}$$

Here, $L_1(\cdot,\cdot)$ denotes the L_1 loss, and $SFC(\cdot)$ represents the Specific Feature Classifier. For the part corresponding to the source domain in soft labels, we train the Specific Feature Classifier with the labels and data of the source domain. This can be expressed as:

$$\mathcal{L}_{cls} = \mathcal{L}_{ce}(SFC\left(f(h_j\left(x^{s_k}\right))\right), y^{s_k})) \tag{8}$$

In summary, the final classification loss of the Specific Feature Classifier can be obtained as follows:

$$\mathcal{L}_{specific} = \mathcal{L}_{cls} + \mathcal{L}_1 \tag{9}$$

To enhance the performance of the Invariant Feature Classifier, we aim for consistent classification results across all target domain data on different classifiers. We employ an L_1 loss constraint to ensure similarity among the results obtained by all Invariant Feature Classifiers. This process can be expressed as:

$$\begin{aligned}\mathcal{L}_{disc} = & \frac{2}{K \times (K-1)} \sum_{j=1}^{K-1} \sum_{i=n+1}^{K} \mathbf{E}_{x \sim X_t} \left[\left| \, IFC_i\left(f(h_i\left(x^{s_j}\right))\right) \right.\right. \\ & \left.\left. - IFC_j\left(f(h_j\left(x^{s_j}\right)))\right) \, \right| \right]\end{aligned} \tag{10}$$

Here, IFC_i and IFC_j represent the Invariant Feature Classifier of the i-th and j-th source domains, respectively. Finally, we average the weighted outputs of all classifiers to obtain the final classification result.

3.5. Overall of Model

Algorithm 1 FDFM

```
Input: Data: $K$ source domains $\mathcal{D}_{s_k} = \{(x_i^{s_k}, y_i^{s_k})\}_{i=1}^{n_{s_k}}$
    and target domain $\mathcal{D}_t = \{x_j^t\}_{j=1}^{n_t}$. Networks: FDFM
    model $M$. Parameters: hyperparameter $\alpha$.
Output: The optimal FDFM $M$.
for $n$ to $N$ do
    $f_{s_k}^d \leftarrow h_k(f(x_i^{s_k}))$; $f_{t_k}^d \leftarrow h_k(f(x_j^t))$.
    $f_{s_k}^{invariant} \leftarrow ICNN_k(f_{s_k}^d)$; $f_{t_k}^{invariant} \leftarrow$
        $ICNN_k(f_{t_k}^d)$; $f_{t_k}^{specific} \leftarrow SCNN_k(f_{t_k}^d)$.
    Calculate $\mathcal{L}_{mmd}$ in Eq. (3) with each pair of
        $f_{s_k}^{invariant}$ and $f_{t_k}^{invariant}$.
    Calculate $\mathcal{L}_{cos}$ in Eq. (5) with each pair of
        $f_{t_k}^{invariant}$ and $f_{t_k}^{specific}$.
    Concatenate $f_{t_k}^{specific}$ and $f_{s_k}^{invariant}$ to obtain
        $f_{s_k}^{fusion}$.
    $y_{s_k} \leftarrow IFC(f_{s_k}^{invariant})$; $y_{s_k}^{fusion} \leftarrow$
        $SFC(f_{s_k}^{fusion})$.
    Calculate $\mathcal{L}_{invariant}$, $\mathcal{L}_{specific}$ and $\mathcal{L}_{disc}$, in Eq.
        (6), Eq. (9) and Eq. (10).
    Compute $\mathcal{L}_{total}$ in Eq.(11)
    Update parameters of $M$ by minimizing $\mathcal{L}_{total}$.
end for
return $M$.
```

In the context of multi-source domains, extracting features with strong discriminability is challenging. Additionally, domain adaptation in both source and target domains leads to a loss of discriminability in the extracted features. To address this, we propose the Feature Disentanglement and Fusion Model. In FDFM, we leverage disentangled domain features to extract domain-specific features, and collaboratively enhance the model's classification performance through a feature fusion mechanism. Our model comprises a base feature extractor, N domain feature extractors, N FD blocks, and 2N classifiers, as depicted in Figure 1. Generally, our features encompass four components: source domain classification loss $\mathcal{L}_{invariant}$, domain adaptation loss $\mathcal{L}_{mmd}$, feature disentanglement loss $\mathcal{L}_{fd} = \mathcal{L}_{cos} + \mathcal{L}_{specific}$ and classifier alignment loss $\mathcal{L}_{disc}$. The overall optimized loss of our model is expressed as:

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{invariant} + (1-\alpha)(\mathcal{L}_{mmd} + \mathcal{L}_{fd} + \mathcal{L}_{disc}) \quad (11)$$

In the equation, α is a hyperparameter that governs the weighting between the source-domain classification loss and the other losses. The pseudo-code for our model is shown in Algorithm 1.

4. Experiment

In this section, we assess the effectiveness of our model against other state-of-the-art adaptation models on four datasets, namely Office-Home, Office-CALTECH10, Office-31, and PACS.

4.1. Data sets and Setup

Office-Home[43] is a widely used dataset for transfer learning, consisting of four domains: Art (Ar), Clipart (Cl), Product (Pr), and Real World (Rw). Each domain in Office-Home contains 65 categories, totaling 15,500 images. We formulate four tasks for multi-source domain adaptation, where three domains act as source domains, and one domain serves as the target domain. The tasks are structured as follows: $Ar, Cl, Pr \rightarrow Rw$; $Ar, Cl, Rw \rightarrow Pr$; $Ar, Rw, Pr \rightarrow Cl$ and $Rw, Cl, Pr \rightarrow Ar$.

Office-CALTECH-10[44] comprises 10 categories from the Amazon (A), Webcam (W), DSLR (D), and Caltech256 (C) image domains. Each category consists of samples ranging from 8 to 151, resulting in a total of 2,533 images. Similarly, we have devised four tasks for multi-source domain adaptation to assess our model: $A, W, D \rightarrow C$; $A, W, C \rightarrow D$; $A, C, D \rightarrow W$ and $C, W, D \rightarrow A$.

Office-31[45], a widely recognized dataset in the realm of transfer learning, comprises 4,652 images of 31 different objects commonly found in office settings, such as laptops, filing cabinets, keyboards, and so on. The images primarily come from Amazon (A), Webcam (W), and DSLR (D). These images are predominantly sourced from online e-commerce images, low-resolution images taken by webcams, and high-resolution images captured by DSLR cameras. We evaluated our model on three tasks: $A, W \rightarrow D$; $A, D \rightarrow W$ and $D, W \rightarrow A$.

PACS[46] is a significant transfer learning dataset with a total of 9,991 images distributed across 7 categories: Dog, Elephant, Giraffe, Guitar, Horse, House, Person, organized into 4 domains: Art painting (A), Cartoon (C), Photo (P), and Sketch (S). We evaluated our model on the following tasks independently: $A, C, P \rightarrow S$; $A, C, S \rightarrow P$; $A, S, P \rightarrow C$ and $S, C, P \rightarrow A$.

Setup: In the setup section, we followed the approach of MFSAN[11] to report the average classification accuracy across five random trials. We conducted three standard comparisons: 1) Single Best: This reports the optimal results among multiple single-source domain adaptations in the multi-source domain, contrasting the upper bounds of single-source and multi-source domain adaptation. 2) Source Combine: This merges different source domains into one, regardless of the distribution variance. 3) Multi-Source: This compares the classification outcomes of multi-source domain adaptation methods. We compared our model with cutting-edge algorithms designed for image classification problems. Our model was implemented using the PyTorch framework, with an initial learning rate set at 0.002, which decreased over iterations. We utilized the SGD optimizer with a momentum of 0.9. The backbone network for feature extraction was ResNet50[47]. For other algorithms, ResNet50 also served as the backbone network. We set a learning rate of 0.002 and used SGD as the optimizer in these models. For all MMD-based methods, we employed a Gaussian kernel with the bandwidth set to the median pairwise squared distances on the training data.

Table 1. Performance Comparison of Classification Accuracy (%) on The Office-Home Dataset.

Standards	Method	Ar,Cl,Pr-Rw	Ar,Cl,Rw-Pr	Ar,Pr,Rw-Cl	Cl,Pr,Rw-Ar	AVG
Single Best	ResNet[47]	75.4	79.7	49.6	65.3	67.5
	DDC[28]	75.0	78.2	50.8	64.1	67.0
	DAN[29]	75.9	80.3	56.5	68.2	70.2
	D-CORAL[48]	76.3	80.3	53.6	67.0	69.3
	RevGrad[49]	75.8	80.4	55.9	67.9	70
	MRAN[50]	77.5	82.2	60.0	70.4	72.5
	MDDA[51]	77.8	81.8	57.6	67.9	71.3
	DDAN[52]	72.7	78.9	56.6	65.1	68.3
	ALDA[53]	77.1	82.1	56.3	70.2	71.4
Source Combine	DAN[29]	82.5	79.0	59.4	68.5	72.4
	D-CORAL[48]	82.7	79.5	58.6	68.1	72.2
	RevGrad[49]	82.7	79.5	59.1	68.4	72.4
Multi-Source	MFSAN[11]	81.8	80.3	62.0	72.1	74.1
	MetaMDA[54]	83.4	81.2	60.5	70.2	73.8
	MSCLDA[12]	80.6	79.9	61.4	71.6	73.4
	DCA[55]	81.4	80.5	63.6	72.1	74.4
	Ours	**84.1**	**83.7**	**64.6**	**74.1**	**76.6**

4.2. Results

The experimental results for Office-Home, Office-31, Office-CALTECH10, and PACS are collectively presented in Tables 1-4. It is evident that our model consistently outperforms existing models across most transfer tasks. Notably, our model demonstrates a significant improvement in the average result, enhancing it from 74.4% to 76.6% on the Office-Home dataset. These results affirm the effectiveness of our model in extracting domain-specific features to enhance feature discriminability.

The experimental results unveiled several noteworthy observations.

- In standard domain adaptation experiments, multi-source domain adaptation models consistently outper-

Table 2. Performance Comparison of Classification Accuracy (%) on The Office-31 Dataset.

Standards	Method	A,W-D	A,D-W	D,W-A	AVG
Single Best	ResNet[47]	99.3	96.7	62.5	86.2
	DDC[28]	98.2	95	67.4	86.9
	DAN[29]	99.5	96.8	66.7	87.7
	D-CORAL[48]	99.7	98	65.3	87.7
	RevGrad[49]	99.1	96.9	68.2	88.1
	RTN[21]	99.4	96.8	66.2	87.5
	MRAN[50]	99.8	96.9	70.9	89.2
	MDDA[51]	99.2	97.1	73.2	89.8
	DDAN[52]	100.0	96.7	65.3	87.3
	ALDA[53]	100.0	97.7	72.5	90.1
	MADA[56]	99.6	97.4	70.3	89.1
	TCA[57]	95.2	93.2	51.6	80.0
	GFK[58]	95.0	95.6	52.4	81.0
	DRCN[59]	99.0	96.4	56.0	83.8
Source Combine—	DAN[29]	99.6	97.8	67.6	88.3
	D-CORAL[48]	99.3	98.0	67.1	88.1
	RevGrad[49]	99.7	98.1	67.6	88.5
	DCTN[60]	99.3	98.2	64.2	87.2
Multi-Source	MFSAN[11]	99.5	98.5	72.7	90.2
	MSCLDA[12]	**99.8**	98.8	73.7	90.8
	DCA[55]	99.6	**98.9**	**75.1**	**91.2**
	Ours	**99.8**	**98.9**	74.1	90.9

Table 3. Performance Comparison of Classification Accuracy (%) on OFFICE-CALTECH10 Dataset.

Standards	Method	A,D,W-C	C,D,W-A	A,C,D-W	A,C,W-D	AVG
Single Best	ResNet[47]	82.5	91.2	98.9	99.2	93.0
	ADDA[25]	88.8	94.5	99.1	98	95.1
	CyCADA[26]	89.7	96.2	98.9	97.3	95.5
Source Combine	DAN[29]	89.7	94.8	99.3	98.2	95.5
	ADDA[25]	90.2	95.0	99.4	98.2	95.7
	CyCADA[26]	91.0	95.9	99.0	97.8	95.9
	DCTN[60]	90.2	92.7	99.4	99.0	95.3
Multi-Source	M3SDA[61]	92.2	94.5	99.5	98.2	96.1
	MFSAN[11]	93.8	95.1	99.1	98.7	96.7
	MSCLDA[12]	94.1	95.3	99.1	98.5	96.8
	DCA[55]	94.7	**96.0**	**99.7**	99.1	97.4
	Ours	**95.6**	**96.0**	99.4	**100.0**	**97.8**

Table 4. Performance Comparison of Classification Accuracy (%) on The PACS Dataset.

Methods	A	C	P	S	Avg
ResNet[47]	81.22	78.54	95.45	72.54	81.94
MDAN[62]	83.54	82.34	92.91	72.42	82.80
DCTN[60]	84.67	86.72	95.60	71.84	84.71
M3SDA[61]	84.20	85.68	94.47	74.62	84.74
MDDA[62]	86.73	86.24	93.89	77.56	86.11
LtC-MSDA[63]	90.19	90.47	97.23	81.53	89.85
DAC-Net[39]	91.39	91.39	97.93	**84.97**	91.42
Ours	**92.24**	**92.40**	**98.08**	84.65	**91.84**

formed their single-source counterparts in both the single best and source combined scenarios. This outcome underscores the ability of multi-source domain adaptive models to effectively leverage features from diverse domains, as opposed to a mere combination in single-source adaptation models.

- Notably, in comparison to models lacking domain-specific features, such as DCA, M3SDA, MetaMDA, and MFSAN (which shares a similar structure with our model), our experimental results demonstrate a substantial improvement. This suggests that the inclusion of domain-specific features in our model effectively enhances the feature representation. While prior methods overlook the considerable discriminative information within target domain-specific features, our model adeptly utilizes this information, enhancing feature discriminability through a feature fusion approach.

- Our model is compared against a range of recent state-of-the-art (SOTA) models, and the results consistently highlight the effectiveness of our approach.

- In contrast to the MSCLDA model, which relies on weighting based on the different contributions of the source domains, our model employs a feature fusion mechanism to combine domain-invariant and domain-specific features at the feature level. Experimental results demonstrate that our feature fusion mechanism is more effective when compared to other methods.

- For the experiments on the Office-31 dataset, although our model achieved the best results on both tasks, it performed slightly worse on one task and the average result. This discrepancy may be attributed to the fact that the Office-31 dataset has only three domains, resulting in a smaller number of source domains compared to the other datasets. This smaller dataset size may lead to increased errors in our model when predicting domain-specific feature labels.

Table 5. Performance Comparison of Classification Accuracy (%) on OFFICE-Home Dataset With Different Loss.

$\mathcal{L}_{mmd}$	$\mathcal{L}_{cos}$	$\mathcal{L}_{specific}$	Rw	Pr	Cl	Ar	Avg
✓	✓	✓	**84.1**	**83.7**	**64.6**	**74.1**	**76.7**
	✓	✓	83.6	82.7	64.2	73.6	76.0
✓		✓	83.2	82.2	63.4	73.3	75.5
✓	✓		80.9	81.6	62.4	72.6	74.4
			80.8	79.5	59.6	68.2	72.0

The aforementioned outcomes collectively illustrate the efficacy of our model in addressing multi-source domain adaptation tasks.

4.3. Analysis

To comprehensively investigate our model, we conducted a series of supplementary experiments and analyzed the obtained results.

- Ablation Study: To assess the contribution of each module in our model, we conducted ablation experiments on four tasks using the Office-Home dataset, and the results are presented in Table 5. We established the average-weighted multi-source domain classification model as the baseline, involving solely the classification loss, without incorporating $\mathcal{L}mmd$, $\mathcal{L}cos$,

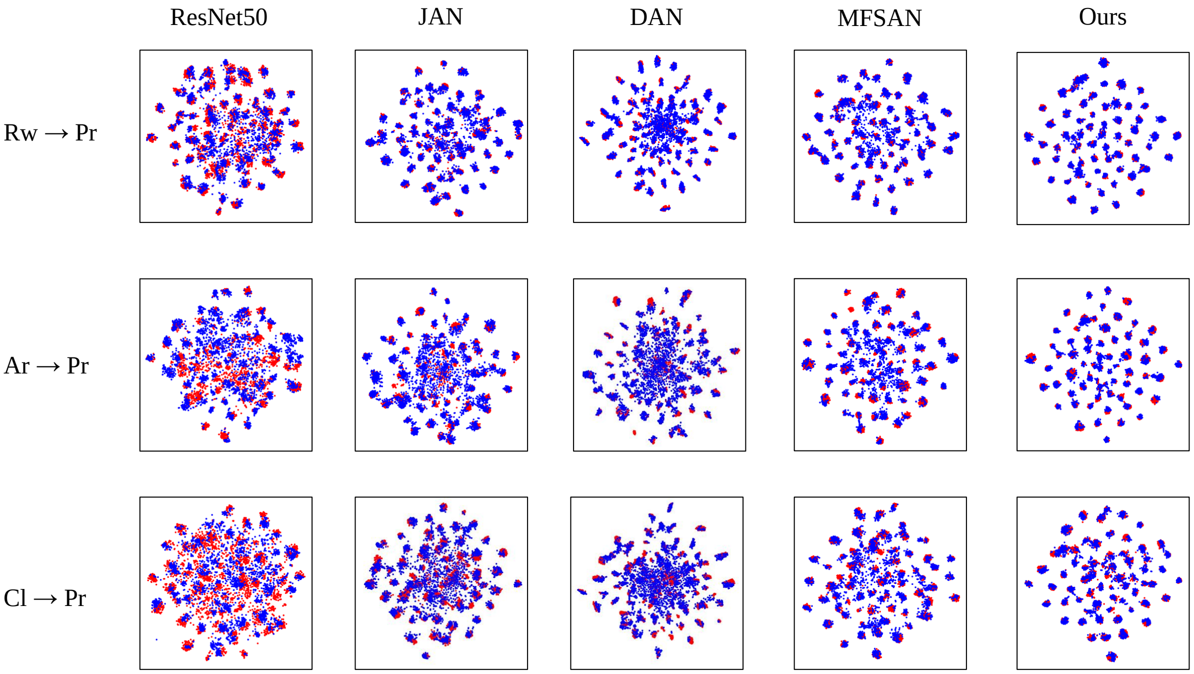

Figure 2. The visualization of feature representations learned on Office-Home. The color of the source samples are blue and the color of the target samples is red.

and $\mathcal{L}specific$. A comparison between the first and fourth rows indicates a 2.3% reduction in accuracy after the exclusion of $\mathcal{L}specific$, underscoring the significant enhancement in performance provided by domain-specific features. Additionally, the comparison between the second and third rows reveals that feature disentanglement encourages blocks to focus more on their respective tasks, demonstrating that $\mathcal{L}cos$ can, to some extent, replace $\mathcal{L}mmd$. Experiments show that the combination from three different losses exemplifies the superiority of our model.

- Feature Visualization: To delve deeper into the transfer capabilities of our model, we visualized the features using t-SNE embedding learning, as illustrated in Fig. 2. Our experiments were conducted on the Office-Home dataset, and we compared our model with the standard baseline model ResNet50, the traditional single-source domain adaptation model JAN[64], the deep single-source domain adaptation model DAN, and the deep multi-source domain adaptation model MFSAN. In Fig. 2, target domain samples are denoted by blue points, while red points represent source domain samples. The experimental results reveal that our model exhibits an enhanced feature alignment capability in comparison to other models. Specifically, ResNet50 demonstrates inferior alignment compared to other models, while the multi-source domain adaptation model MFSAN exhibits superior alignment compared to JAN and DAN. In feature visualization, our model demonstrates smaller intra-class distances and larger inter-class distances compared to other methods. This is attributed to our model's ability to prompt the domain-invariant feature extractor to focus more on domain-invariant features through feature disentanglement.

- Fusion Feature Validity: To assess the efficacy of the feature fusion block and fusion features in our model, we tracked the number of correct samples with different feature-based classifiers in domains Ar and Cl for the task $Ar, Cl, Rw \rightarrow Pr$ within the Office-Home dataset. The experimental results, depicted in Fig. 4, showcase the number of correct samples for the classifier based on fusion features in orange, while the blue denotes the number of correct samples based on domain-invariant features. Figures 4 exhibit a similar trend. In the early stages of training, the correct samples of the fusion feature-based classifier remain at a low level, owing to the initial imprecision of the soft labels. However, as the number of iterations increases, the fusion feature-based classifier's correct samples surpass those of the domain-invariant feature-based classifier. In the final stages, the number of

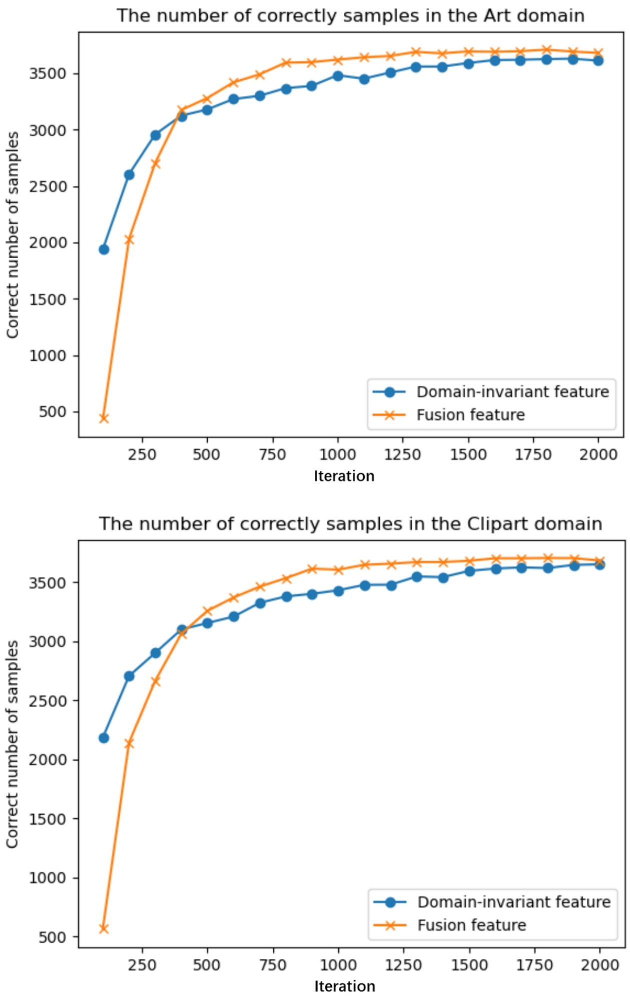

Figure 3. The number of samples correctly classified by different classifiers in the Art and Clipart domains on the Product is the target domain task.

correct samples for the domain-invariant feature-based classifier gradually converges, highlighting the advantage of feature disentanglement. Classifiers based on fusion features maintain a slight lead. These experimental results comprehensively demonstrate the effectiveness of our model's feature fusion mechanism.

- Convergence: To illustrate the convergence of our model, we plotted loss curves for the task $Ar, Cl, Rw \to Pr$ on the Office-Home dataset, as depicted in Fig. 5. Our model's loss exhibits a significant decline in the initial stages of training and begins to stabilize around 1000 iterations, maintaining this trend until the completion of training. These experimental results provide additional evidence of the enhanced convergence and stability of our model.

- Hyperparameter Setting: In our exploration of hyperparameters, we varied the values for α within the set 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9. The model's accuracy in the $Ar, Cl, Rw \to Pr$ task on the Office-Home dataset was recorded. The results, presented in Fig. 6, exhibit a bell-shaped curve, with accuracy initially increasing and then decreasing. Consequently, we selected α = 0.6 as the optimal value, producing the most favorable results for our model.

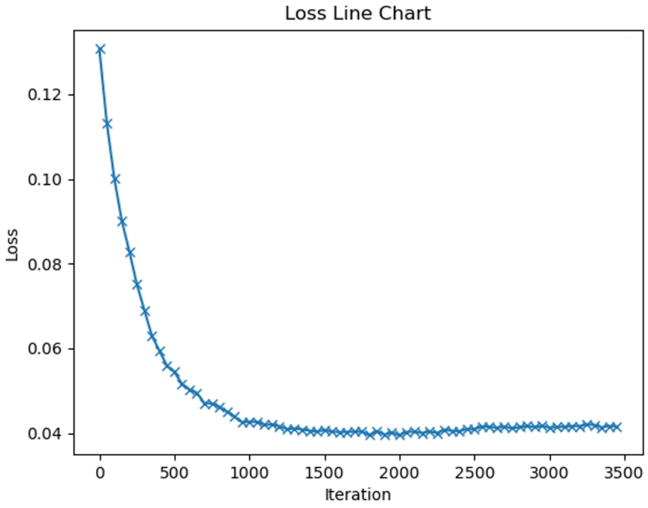

Figure 4. The loss of our model for task $Ar, Cl, Rw \to Pr$ on the Office-Home dataset.

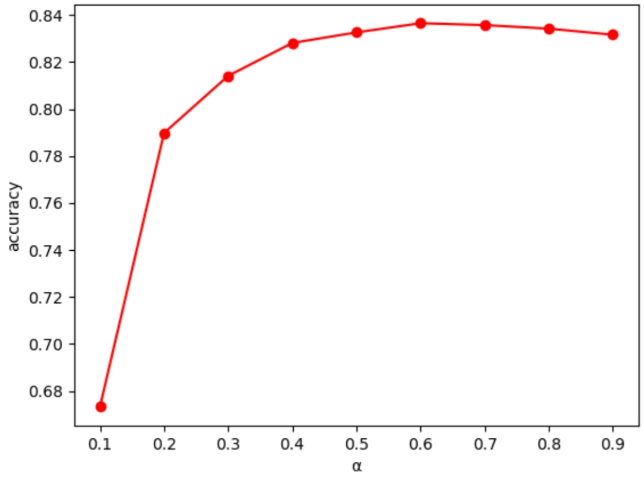

Figure 5. The accuracy of our model with different α.

5. Conclusion and Discussion

Existing adaptation methods have predominantly emphasized the extraction of domain-invariant features, neglecting the significance of target domain-specific features. This paper introduces a novel multi-source domain adaptation approach grounded in feature disentanglement and fusion. By employing a self-supervised technique, we extract target domain-specific features and leverage a feature fusion mechanism to augment feature representation. Extensive experiments on four benchmark datasets affirm the efficacy of our proposed model.

6. Acknowledgement

This work was supported in part by the Yunnan Fundamental Research Projects under Grant 202401AU070151, and in part by Practical Innovation Project of Postgraduate Students in the Professional Degree of Yunnan University ZC-23234251.

References

[1] K. Weiss, T. M. Khoshgoftaar, D. Wang, A survey of transfer learning, Journal of Big data 3 (1) (2016) 1–40. 1

[2] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (10) (2009) 1345–1359. 1

[3] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, in: Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27, Springer, 2018, pp. 270–279. 1

[4] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proceedings of the IEEE 109 (1) (2020) 43–76. 1

[5] B. Tan, E. Zhong, E. W. Xiang, Q. Yang, Multi-transfer: Transfer learning with multiple views and multiple sources, in: Proceedings of the 2013 SIAM International Conference on Data Mining, SIAM, 2013, pp. 243–251. 1

[6] J. Li, W. Wu, D. Xue, P. Gao, Multi-source deep transfer neural network algorithm, Sensors 19 (18) (2019) 3992. 1

[7] S. Niu, Y. Liu, J. Wang, H. Song, A decade survey of transfer learning (2010–2020), IEEE Transactions on Artificial Intelligence 1 (2) (2020) 151–166. 1

[8] H. Zhao, S. Zhang, G. Wu, J. P. Costeira, J. M. Moura, G. J. Gordon, Multiple source domain adaptation with adversarial learning (2018). 2

[9] O. Amosy, G. Chechik, Coupled training for multi-source domain adaptation, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 420–429. 2

[10] J. Hoffman, M. Mohri, N. Zhang, Algorithms and theory for multiple-source adaptation, Advances in neural information processing systems 31 (2018). 2

[11] Y. Zhu, F. Zhuang, D. Wang, Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 33, 2019, pp. 5989–5996. 2, 7, 8

[12] K. Li, J. Lu, H. Zuo, G. Zhang, Multi-source contribution learning for domain adaptation, IEEE Transactions on Neural Networks and Learning Systems 33 (10) (2021) 5293–5307. 2, 7, 8

[13] A. Farahani, S. Voghoei, K. Rasheed, H. R. Arabnia, A brief review of domain adaptation, Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020 (2021) 877–894. 2

[14] M. Wang, W. Deng, Deep visual domain adaptation: A survey, Neurocomputing 312 (2018) 135–153. 2

[15] K. You, M. Long, Z. Cao, J. Wang, M. I. Jordan, Universal domain adaptation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 2720–2729. 2

[16] X. Liu, C. Yoo, F. Xing, H. Oh, G. El Fakhri, J.-W. Kang, J. Woo, et al., Deep unsupervised domain adaptation: A review of recent advances and perspectives, APSIPA Transactions on Signal and Information Processing 11 (1) (2022). 2

[17] X. Chen, H. Fan, R. Girshick, K. He, Improved baselines with momentum contrastive learning, arXiv preprint arXiv:2003.04297 (2020). 2

[18] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International conference on machine learning, PMLR, 2020, pp. 1597–1607. 2

[19] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 9729–9738. 2

[20] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, W. Zuo, Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2272–2281. 3

[21] M. Long, H. Zhu, J. Wang, M. I. Jordan, Unsupervised domain adaptation with residual transfer networks, Advances in neural information processing systems 29 (2016). 3, 8

[22] J. Gao, R. Huang, H. Li, Sub-domain adaptation learning methodology, Information Sciences 298 (2015) 237–256. 3

[23] H. Yan, Z. Li, Q. Wang, P. Li, Y. Xu, W. Zuo, Weighted and class-specific maximum mean discrepancy for unsupervised domain adaptation, IEEE Transactions on Multimedia 22 (9) (2019) 2420–2433. 3

[24] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, The journal of machine learning research 17 (1) (2016) 2096–2030. 3

[25] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7167–7176. 3, 8

[26] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, T. Darrell, Cycada: Cycle-consistent adversarial domain adaptation, in: International conference on machine learning, Pmlr, 2018, pp. 1989–1998. 3, 8

[27] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, W. Zhang, Adversarial domain adaptation with domain mixup, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 34, 2020, pp. 6502–6509. 3

[28] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, T. Darrell, Deep domain confusion: Maximizing for domain invariance, arXiv preprint arXiv:1412.3474 (2014). 3, 7, 8

[29] M. Long, Y. Cao, J. Wang, M. Jordan, Learning transferable features with deep adaptation networks, in: International conference on machine learning, PMLR, 2015, pp. 97–105. 3, 7, 8

[30] Y. Zhu, F. Zhuang, J. Wang, G. Ke, J. Chen, J. Bian, H. Xiong, Q. He, Deep subdomain adaptation network for image classification, IEEE Transactions on Neural Networks and Learning Systems 32 (4) (2021) 1713–1722. doi:10.1109/TNNLS.2020.2988928. 3

[31] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, P. S. Yu, Visual domain adaptation with manifold embedded distribution alignment, in: Proceedings of the 26th ACM international conference on Multimedia, 2018, pp. 402–410. 3

[32] X. Chen, S. Wang, M. Long, J. Wang, Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation, in: International conference on machine learning, PMLR, 2019, pp. 1081–1090. 3

[33] Q. Jia, J. Guo, P. Yang, Y. Yang, A holistic multi-source transfer learning approach using wearable sensors for personalized daily activity recognition, Complex & Intelligent Systems (2023) 1–13. 3

[34] S. Christodoulidis, M. Anthimopoulos, L. Ebner, A. Christe, S. Mougiakakou, Multisource transfer learning with convolutional neural networks for lung pattern analysis, IEEE journal of biomedical and health informatics 21 (1) (2016) 76–84. 3

[35] J. Li, S. Qiu, Y.-Y. Shen, C.-L. Liu, H. He, Multisource transfer learning for cross-subject eeg emotion recognition, IEEE transactions on cybernetics 50 (7) (2019) 3281–3293. 3

[36] Y. Zhang, H. Tang, K. Jia, M. Tan, Domain-symmetric networks for adversarial domain adaptation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 5031–5040. 3

[37] Y. Yao, G. Doretto, Boosting for transfer learning with multiple sources, in: 2010 IEEE computer society conference on computer vision and pattern recognition, IEEE, 2010, pp. 1855–1862. 3

[38] Q. Jia, J. Guo, F. Du, P. Yang, Y. Yang, A fast texture-to-stain adversarial stain normalization network for histopathological images, in: 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2022, pp. 2294–2301. 3

[39] Z. Deng, K. Zhou, Y. Yang, T. Xiang, Domain attention consistency for multi-source domain adaptation, arXiv preprint arXiv:2111.03911 (2021). 3, 8

[40] M. Xu, H. Wang, B. Ni, Graphical modeling for multi-source domain adaptation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2022). 3

[41] H. Feng, Z. You, M. Chen, T. Zhang, M. Zhu, F. Wu, C. Wu, W. Chen, Kd3a: Unsupervised multi-source decentralized domain adaptation via knowledge distillation., in: ICML, 2021, pp. 3274–3283. 3

[42] M. Mancini, S. R. Bulo, B. Caputo, E. Ricci, Adagraph: Unifying predictive and continuous domain adaptation through graphs, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6568–6577. 3

[43] H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5018–5027. 7

[44] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset (2007). 7

[45] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11, Springer, 2010, pp. 213–226. 7

[46] D. Li, Y. Yang, Y.-Z. Song, T. M. Hospedales, Deeper, broader and artier domain generalization, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 5542–5550. 7

[47] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778. 7, 8

[48] B. Sun, K. Saenko, Deep coral: Correlation alignment for deep domain adaptation, in: Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14, Springer, 2016, pp. 443–450. 7, 8

[49] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: International conference on machine learning, PMLR, 2015, pp. 1180–1189. 7, 8

[50] Y. Zhu, F. Zhuang, J. Wang, J. Chen, Z. Shi, W. Wu, Q. He, Multi-representation adaptation network for cross-domain image classification, Neural Networks 119 (2019) 214–221. 7, 8

[51] S. Zhao, G. Wang, S. Zhang, Y. Gu, Y. Li, Z. Song, P. Xu, R. Hu, H. Chai, K. Keutzer, Multi-source distilling domain adaptation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 12975–12983. 7, 8

[52] J. Wang, Y. Chen, W. Feng, H. Yu, M. Huang, Q. Yang, Transfer learning with dynamic distribution adaptation, ACM Transactions on Intelligent Systems and Technology (TIST) 11 (1) (2020) 1–25. 7, 8

[53] M. Chen, S. Zhao, H. Liu, D. Cai, Adversarial-learned loss for domain adaptation, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 34, 2020, pp. 3521–3528. 7, 8

[54] D. Li, T. Hospedales, Online meta-learning for multi-source and semi-supervised domain adaptation, in: European Conference on Computer Vision, Springer, 2020, pp. 382–403. 7

[55] K. Li, J. Lu, H. Zuo, G. Zhang, Dynamic classifier alignment for unsupervised multi-source domain adaptation, IEEE Transactions on Knowledge and Data Engineering 35 (5) (2022) 4727–4740. 7, 8

[56] Z. Pei, Z. Cao, M. Long, J. Wang, Multi-adversarial domain adaptation, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 32, 2018. 8

[57] S. J. Pan, I. W. Tsang, J. T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, IEEE transactions on neural networks 22 (2) (2010) 199–210. 8

[58] B. Gong, Y. Shi, F. Sha, K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in: 2012 IEEE conference on computer vision and pattern recognition, IEEE, 2012, pp. 2066–2073. 8

[59] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, W. Li, Deep reconstruction-classification networks for unsupervised domain adaptation, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14, Springer, 2016, pp. 597–613. 8

[60] R. Xu, Z. Chen, W. Zuo, J. Yan, L. Lin, Deep cocktail network: Multi-source unsupervised domain adaptation with category shift, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3964–3973. 8

[61] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, B. Wang, Moment matching for multi-source domain adaptation, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1406–1415. 8

[62] H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, G. J. Gordon, Adversarial multiple source domain adaptation, Advances in neural information processing systems 31 (2018). 8

[63] H. Wang, M. Xu, B. Ni, W. Zhang, Learning to combine: Knowledge aggregation for multi-source domain adaptation, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16, Springer, 2020, pp. 727–744. 8

[64] M. Long, H. Zhu, J. Wang, M. I. Jordan, Deep transfer learning with joint adaptation networks, in: International conference on machine learning, PMLR, 2017, pp. 2208–2217. 9