

Unsupervised domain adaptation with fuzzy similarity and fuzzy C-means for classification applications

J. Y. Jhang ¹, C. J. Lin ², X. Q. Lin ³ and H. Cheng ⁴

^{1,2,4} *Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411030, Taiwan, R.O.C.*

³ *Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701401, Taiwan, R.O.C.*

jyjhang@nutc.edu.tw, cjlin@ncut.edu.tw, 4b117051@gmail.com, nocturne7h0m@gmail.com

Abstract

As deep learning applications continue to expand, the performance of network models becomes increasingly critical. To enhance model accuracy, training often relies on a large amount of labeled data. However, manually labeling data is both labor-intensive and expensive. In recent years, Domain Adaptation (DA) techniques have gained significant attention as they enable models to be trained with limited labeled data while maintaining performance on unlabeled data. In particular, Unsupervised Domain Adaptation (UDA) faces a major challenge due to differences in data distributions. Addressing the problem of domain shift and identifying domain-invariant features with limited information remains a crucial task. This paper proposes a fuzzy-theory-based unsupervised domain adaptation approach called the Deep Fuzzy-Similarity-based Local Domain Adaptation Network (DFS-LDAN). The proposed method aims to mitigate domain shift and reduce annotation costs. DFS-LDAN integrates fuzzy similarity and Maximum Mean Discrepancy (MMD), allowing the model to retain more soft information during learning. This enhances its flexibility in handling uncertainties in target domain data. Additionally, this study introduces a Local Domain Adaptation strategy that aligns distributions at the class level. It employs Fuzzy C-Means (FCM) clustering to adaptively adjust feature weights according to membership degrees, thereby enhancing the model's generalization capability. The effectiveness of DFS-LDAN is evaluated on three public datasets: Office-31, ImageCLEF-DA, and Office-Home. Comparative experiments with various domain adaptation methods demonstrate the superiority of DFS-LDAN. The results show that the proposed method, across the best-performing variant, achieves average accuracies of 89.1% (via Product-Sum), 90.4% (via Product-Sum), and 68.4% (via Min-Max), respectively.

Keywords: Unsupervised domain adaptation, maximum mean discrepancy, fuzzy similarity, fuzzy C-means, local domain adaptation.

1 Introduction

Deep learning has been widely applied across various domains, with most tasks relying on large-scale labeled datasets for model training. As data volume grows rapidly, the demand for manually labeled data has become a bottleneck. Data collection requires substantial human effort and time, and the complexity of data annotation varies depending on the application scenario. The performance of deep learning models is often affected by distribution discrepancies between training and testing data. Most models are trained and tested on data drawn from the same distribution. However, in real-world applications, models often encounter variations caused by different environmental conditions, such as weather and location. These distributional shifts lead to performance degradation, a phenomenon known as domain shift. Adapting models to unlabeled data from new domains effectively is a critical challenge. Domain

adaptation (DA) addresses the issue of performance degradation caused by discrepancies between the source domain (where labeled training data is available) and the target domain (where labeled data is scarce or unavailable). In real-world applications, data distributions vary across tasks and environments, and a model lacking sufficient generalization capability may struggle with domain shifts. Depending on the availability of labeled data in the target domain, domain adaptation can be categorized into Unsupervised Domain Adaptation (UDA) [11, 23, 36, 38] where target domain data is entirely unlabeled, and Semi-Supervised Domain Adaptation (SSDA) [5, 25], where a subset of the target domain data is labeled. Furthermore, domain adaptation can be classified based on the number of source and target domains, including single-source single-target [15, 16, 18, 27, 30, 31, 39], multi-source single-target [13, 14, 24], single-source multi-target [10, 32, 37], and multi-source multi-target [21] adaptation.

To address domain shift, existing domain adaptation techniques can be broadly classified into two major categories. The first category is adversarial-based methods, which leverage adversarial learning frameworks, such as Generative Adversarial Networks (GANs) [6], to align feature distributions across domains. These methods typically involve a feature extractor, a classifier, and a domain discriminator. The feature extractor attempts to learn domain-invariant features, which are then used by the classifier for prediction, while the domain discriminator aims to distinguish whether a feature originates from the source or target domain. By jointly optimizing these components, the distributional gap between domains is minimized. A representative approach is Domain-Adversarial Neural Network (DANN) [3], which integrates the concept of GANs and introduces the Gradient Reversal Layer (GRL). GRL inverts the gradient from the domain discriminator before passing it to the feature extractor, facilitating the learning of domain-invariant features while optimizing both the feature extractor and discriminator simultaneously. This approach alleviates the multi-step training challenge in GANs. Ma et al. [23] proposed an Adversarial Entropy Optimization technique, where entropy minimization enhances feature discriminability when samples are independently drawn from the source and target domains. Simultaneously, entropy maximization over joint distributions ensures domain confusion, making it difficult for the discriminator to differentiate between the two domains. The second category is discrepancy-based methods, which measure and minimize the statistical divergence between the source and target domains. The goal is to reduce distributional differences by employing distance metrics such as Kullback-Leibler (KL) divergence [11, 38] and Maximum Mean Discrepancy (MMD) [5, 36]. Kim et al. [11] utilized an Adversarial Autoencoder to map data into a latent subspace, where a Gaussian Mixture Model (GMM) clusters features. KL divergence is then used to quantify the similarity between clusters in the source and target domains, reducing distributional shifts. Zhang et al. [38] proposed the DCCAL model, which incorporates a density-based clustering method (Density-Based Spatial Clustering of Applications with Noise) to cluster target domain samples and generate pseudo-labels, adaptively. The model also integrates a Channel Attention Mechanism and KL divergence to mitigate domain shift and label noise. Ye et al. [36] applied the MMD method during classifier training, employing a probabilistic Linear Support Vector Machine (SVM) for constraint optimization. This re-weighting strategy enhances classification boundaries and reduces negative transfer. Gretton et al. [7] established the theoretical foundation for MMD by introducing the kernel two-sample test, which quantifies the discrepancy between probability distributions by mapping them into a high-dimensional reproducing kernel Hilbert space (RKHS) and utilizing kernel functions to measure their statistical distance.

Maximum Mean Discrepancy (MMD) is a non-parametric feature learning method that aligns source and target domain distributions by comparing their feature representations. Unlike adversarial learning, MMD-based approaches do not rely on adversarial training but instead measure distributional discrepancies in a high-dimensional space. MMD has been widely used for aligning intermediate layers in deep neural networks, ensuring that source and target domain data share common feature representations. Several representative methods leverage MMD for domain adaptation. Deep Domain Confusion (DDC) [27] introduces an adaptation layer within a deep neural network, minimizing MMD loss while training for classification. This encourages the model to learn shared feature representations across domains, enhancing cross-domain recognition. Deep Adaptation Network (DAN) [15, 16] extends this approach by incorporating Multiple Kernel MMD (MK-MMD) across multiple network layers, enabling the model to align different levels of feature distributions and improve generalization. Joint Adaptation Network (JAN) [18] further integrates adversarial training strategies with MK-MMD, proposing Joint MMD (JMMD), which maximizes JMMD to distinguish domain-specific features while minimizing classification loss, achieving an adversarial effect to enhance network performance. Manifold Embedded Distribution Alignment (MEDA) [30] extends MMD by introducing Manifold Feature Learning, which addresses feature degradation issues through manifold alignment. Additionally, it dynamically adjusts the MMD loss weight to accommodate varying inter-domain distributional differences, improving the model's alignment capability. Transferable Attention for Domain Adaptation (TADA) [31] and Attention-guided MMD (AMMD) [33] integrate MMD with attention mechanisms to assign higher weights to discriminative features, enhancing domain adaptation stability. Domain-Specific Alignment Network (DSAN) [39] introduces Local Maximum Mean Discrepancy (LMMD) to align subdomains corresponding to the same class across different domains. This approach not only considers global domain shifts but also stabilizes class boundaries, reducing classification errors caused by domain transfer. However, the

forementioned methods overlook the issue of uncertainty. There exists a significant interdependence between model performance and the amount of available information. When training under limited information conditions, it is crucial to account for the soft information arising from uncertainty. To address this challenge, recent studies have explored the integration of fuzzy techniques into deep learning. Fuzzy Theory-based methods provide an effective means of representing feature uncertainty [20], offering a potential solution for improving robustness in domain adaptation.

Fuzzy theory is a mathematical framework explicitly designed to model uncertainty and imprecision in complex systems. Although the modern data mining community predominantly relies on probabilistic architectures for classification, traditional probabilistic paradigms face inherent bottlenecks in UDA. Under severe cross-domain distribution shifts, standard Softmax layers tend to produce overconfident yet erroneous predictions for ambiguous target samples situated near decision boundaries. This rigid, sharp-boundary probability representation forces the model to discard valuable soft information, propagating pseudo-label noise and causing severe misalignment during local domain adaptation. In contrast, fuzzy set theory offers a fundamentally more resilient mechanism by replacing absolute probabilities with continuous fuzzy membership degrees. This allows a single ambiguous sample to simultaneously belong to multiple semantic categories with varying intensities, gracefully capturing the structural uncertainty inherent in unlabelled target domains. To theoretically validate this integration, recent hybrid studies have successfully coupled fuzzy concepts with modern learning systems. Eftekhari et al. [2] comprehensively reviewed the contributions of fuzzy concepts to machine learning, demonstrating that fuzzy set theory effectively enhances data representation, feature selection, clustering, classification, and uncertainty management. Their study highlighted that fuzzy methodologies provide a flexible mechanism for handling ambiguous and noisy data, thereby improving the reliability and adaptability of machine learning models. Furthermore, the authors emphasized that the integration of fuzzy logic with modern learning paradigms enables more human-like reasoning and supports the development of intelligent systems capable of dealing with real-world uncertainties. Building upon these unique advantages, researchers have increasingly incorporated fuzzy theory into deep architectures to enhance cross-domain adaptability. Hansan et al. [8] applied fuzzy theory to pooling layers and proposed the Fuzzy Pooled Convolutional Neural Network. This approach fuzzifies feature maps extracted by convolutional layers, utilizing multiple membership functions to determine optimal pooling strategies. Unlike traditional pooling methods, which select only a single pooling operation, this technique allows networks to consider multiple pooling options simultaneously, leading to more robust feature representations. To address uncertainty in domain adaptation, researchers have increasingly incorporated fuzzy theory into their methodologies, leveraging fuzzy uncertainty to enhance cross-domain adaptability. Xu et al. [35] and Lou et al. [19] introduced the Takagi-Sugeno-Kang (TSK) fuzzy system to map data into a fuzzy feature space, minimizing distributional discrepancies between domains. Compared to conventional shared feature space approaches, the fuzzy system enhances interpretability and adapts more effectively to nonlinear transformations. Wang et al. [29] designed a Fuzzy Inference Attention module within an attention mechanism. This module evaluates the quality of data based on its category and domain, applying customized fuzzy rules to identify samples with high class-specific information but minimal domain-specific bias. These samples are assigned higher weights, making them ideal for domain transfer and mitigating negative transfer effects—achieved without requiring additional trainable parameters. Experimental results demonstrated that integrating this module into various networks improves their performance. Lu et al. [22] employed a TSK fuzzy model to learn fuzzy rules from source domain data. They then used an Infinite Gaussian Mixture Model to merge fuzzy rules from multiple source domains while preserving domain-specific characteristics. This method generates an appropriate number of rules for the target domain and applies Fuzzy C-means (FCM) clustering to find corresponding fuzzy rules for domain adaptation. Xiao et al. [34] combined adversarial learning and fuzzy theory to propose a Joint Adversarial Fuzzy-weighted Method. This approach utilizes fuzzy clustering membership degrees to weight unknown class classifiers. Additionally, they introduced Multi-channel Multi-kernel Weighted Local Maximum Mean Discrepancy (M2W-LMMD), which refines conditional distribution alignment at a class-specific level, facilitating subdomain-level adaptation. Li et al. [14] addressed both class-wise and domain-specific soft information by incorporating FCM into a TSK fuzzy model. They proposed the Source-free Multidomain Adaptation with Fuzzy Rule-based Deep Neural Networks, where FCM is used to compute clustering centers for each category in both source and target domains. The method aligns domains by minimizing distances between corresponding source and target cluster centers.

To enhance cross-domain generalization and address data uncertainty, this paper proposes the Deep Fuzzy-similarity-based Local Domain Adaptation Network (DFS-LDAN). MMD-based methods primarily focus on aligning global distributions between domains, often overlooking class-wise relationships. In contrast, our approach introduces a Fuzzy Similarity-based Local Maximum Mean Discrepancy, which assigns weights to MMD distances based on fuzzy similarity characteristics. This preserves soft information while improving feature alignment. This paper makes the following key contributions:

- We propose a novel unsupervised domain adaptation network, DFS-LDAN, which integrates fuzzy similarity and local MMD to enhance domain adaptation by preserving fuzzy features and identifying class-wise domain-invariant

representations.

- By incorporating FCM, we evaluate the membership degree between features and categories, preserving soft information and improving generalization.
- The proposed method achieves state-of-the-art performance on three benchmark domain adaptation datasets.

The remainder of this paper is structured as follows. Section 2 presents the overall model architecture and learning algorithm. Section 3 reports experimental results and compares our approach with existing methods. Section 4 concludes the study and discusses future research directions.

2 The proposed DFS-LDAN

This section provides a detailed introduction to the proposed Deep Fuzzy-Similarity-Based Local Domain Adaptation Network (DFS-LDAN), divided into two parts: (1) Overall Model Architecture, including the use of MMD, Fuzzy Similarity-Based Local Domain Adaptation, and FCM. (2) Learning Algorithm for DFS-LDAN.

2.1 DFS-LDAN model architecture

The overall architecture of the proposed DFS-LDAN network is illustrated in Figure 1. The input consists of a source domain dataset $X_s = \{(x_i^s, y_i^s) | i = 1, \dots, n_s\}$ and a target domain dataset $X_t = \{x_j^t | j = 1, \dots, n_t\}$, where x_i^s represents a source domain sample; x_j^t represents a target domain sample; y_i^s denotes the label information of the source domain data; n_s and n_t indicate the number of samples in the source and target domains, respectively. The input data is first processed by a backbone network $F(\cdot)$ to extract feature representations, yielding: $Z_s = F(X_s)$ and $Z_t = F(X_t)$. These extracted features are then passed through a classifier, generating predicted output vectors $\hat{y}_s$ and $\hat{y}_t$. After feature extraction, the features are fed into both the Fuzzy Similarity-Based Local Domain Adaptation Module and the FCM Module. The Fuzzy Similarity-Based Local Domain Adaptation module enables the network to identify domain-invariant features for each category. By leveraging fuzzy similarity, this module aligns feature distributions between source and target domains within each class, minimizing the local class-wise feature distribution distance $d_{src \leftrightarrow tar}^{cls_n}$. The FCM Module employs fuzzy clustering to compute the membership degree U of each feature to different categories. This information is used to weight the predicted output $\hat{y}_t$ for target samples, enhancing the model's generalization capability.

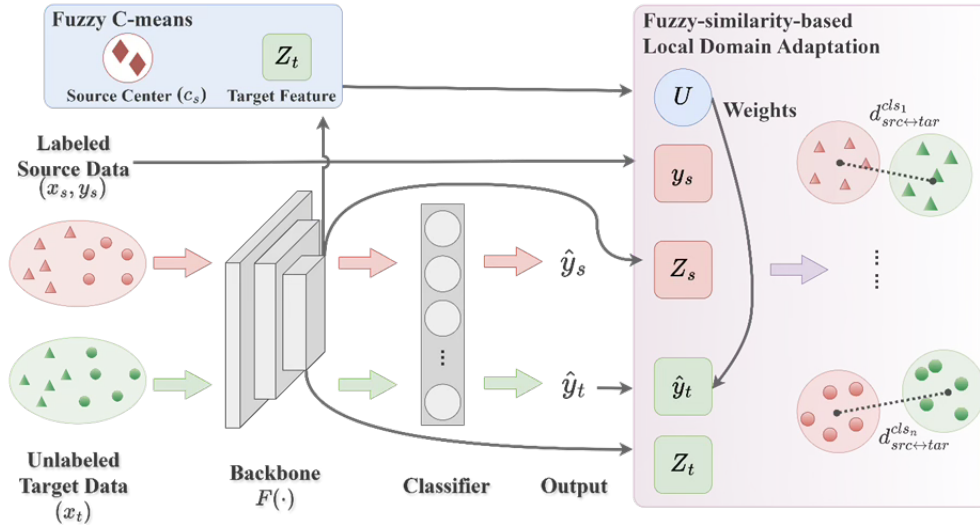


Figure 1: Architecture of the DFS-LDAN Network.

2.1.1 Maximum mean discrepancy

MMD is a statistical distance metric that quantifies the discrepancy between two probability distributions after mapping data into a different feature space. Given the source domain dataset X_s and the target domain dataset X_t , let their corresponding data distributions be denoted as P and Q , respectively. The definition of MMD is as follows:

$$\text{MMD}[F, X_s, X_t] = \sup_{f \in F} (\mathbb{E}_{X_s \sim P}[f(X_s)] - \mathbb{E}_{X_t \sim Q}[f(X_t)]), \quad (1)$$

where F represents a family of functions, f is a mapping function, and $\mathbb{E}_{X_s}$ and $\mathbb{E}_{X_t}$ denote the expectations of the mapped source and target domain samples under distributions P and Q , respectively. The term $\sup_{f \in F}$ signifies finding the function f within the function family F that maximizes the mean discrepancy.

MMD primarily relies on Reproducing Kernel Hilbert Space (RKHS) for feature mapping. It employs a kernel function to map data into a higher-dimensional feature space. Common kernel functions include the linear kernel, polynomial kernel, and radial basis function (RBF) kernel, also known as the Gaussian kernel. The Gaussian kernel maps data into a high-dimensional RKHS and measures similarity between samples through the inner product of their mapped representations. The value of the Gaussian kernel ranges from 0 to 1, decreasing as the distribution distance increases. When two samples are very close, the kernel value approaches 1, while a value approaching 0 indicates low similarity. The Gaussian kernel is widely used in MMD due to its strong flexibility and local sensitivity, making it effective for handling complex nonlinear data distributions. The Gaussian kernel function K is defined as:

$$K(x_i^s, x_j^t) = \exp\left(-\frac{\|x_i^s - x_j^t\|^2}{2\sigma^2}\right), \quad (2)$$

where X_i^s and X_j^t are source and target domain samples, respectively, $\|x_i^s - x_j^t\|^2$ denotes the Euclidean distance, and σ is the standard deviation of the Gaussian kernel.

Since the expectation values in (1) are not directly obtainable in practice, an empirical estimation is applied by computing the sample mean. The reformulated MMD expression is given by:

$$\text{MMD}^2(X_s, X_t) = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(x_i^s) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(x_j^t) \right\|_H^2 = \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} K(x_i^s, x_j^s) + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} K(x_i^t, x_j^t) - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} K(x_i^s, x_j^t), \quad (3)$$

where n_s and n_t are the number of samples in the source and target domains, respectively; H represents the RKHS space, $\phi(\cdot)$ denotes the feature mapping function, and K is the Gaussian kernel function.

The difference between Global DA and Local DA is illustrated in Figure 2, where triangles and circles represent different classes, and the dashed line denotes the decision boundary. Figure 2 (a) is source domain data distribution. Figure 2 (b) is target domain data distribution. As shown in Figure 2 (c), the global DA aligns the overall distributions, improving cross-domain feature extraction. However, it may cause the network to lose class-specific information, leading to misclassification of the ambiguous features near the decision boundary, which degrades classifier performance. Figure 2 (d) shows that Local DA ensures that samples of the same class remain close in feature space while preserving the separation between different classes. This approach enhances model generalization and retains class-specific information, mitigating issues caused by misaligned distributions.

Unlike Global DA, Local DA focuses on aligning distributions at the class level, thereby improving target domain learning performance. In addition to ensuring global distribution similarity, Local DA requires that the distribution of each class be aligned between the source and target domains. This fine-grained approach is particularly effective when category distributions differ between domains. To achieve Local DA, the MMD formulation is modified as follows:

$$\begin{aligned} \text{Local DA}(X_s, X_t) &= \sum_{c=1}^C \left\| \frac{1}{n_s^c} \sum_{i=1}^{n_s^c} \phi(x_i^s) - \frac{1}{n_t^c} \sum_{j=1}^{n_t^c} \phi(x_j^t) \right\|_H^2 \\ &= \sum_{c=1}^C \left[\frac{1}{(n_s^c)^2} \sum_{i=1}^{n_s^c} \sum_{j=1}^{n_s^c} K(x_i^s, x_j^s) + \frac{1}{(n_t^c)^2} \sum_{i=1}^{n_t^c} \sum_{j=1}^{n_t^c} K(x_i^t, x_j^t) - \frac{2}{n_s^c n_t^c} \sum_{i=1}^{n_s^c} \sum_{j=1}^{n_t^c} K(x_i^s, x_j^t) \right], \end{aligned} \quad (4)$$

where C is the total number of classes, n_s^c and n_t^c denote the number of samples in class C from the source and target domains, respectively.

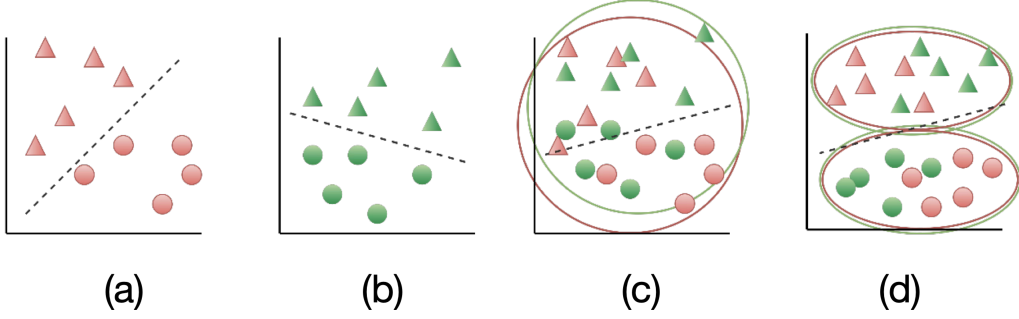


Figure 2: Comparison of Global and Local Domain Adaptation: (a) Source domain distribution, (b) Target domain distribution, (c) Global DA, (d) Local DA.

2.1.2 Fuzzy similarity-based local domain adaptation

The core objective of Local DA is T, align the distributions of the same class in both the source and target domains. However, since target domain data are typically unlabeled and the class boundaries predicted by the model are often unclear, directly using the classifier’s predicted labels may lead to misjudgments. To address this issue, we incorporate fuzzy similarity to handle uncertainty and ambiguous boundaries. The proposed approach considers the uncertainty of target samples by leveraging their predicted probabilities along with the actual labels from the source domain. The similarity between samples is computed and used as a weight for each sample. We extend the Local DA formula into a fuzzy similarity-weighted form, as defined below:

$$\begin{aligned}
 \text{FSLDA}(X_s, X_t, Y_s, \hat{Y}_t) &= \sum_{c=1}^C \left\| \frac{1}{n_s^c} \sum_{i=1}^{n_s^c} W_{si}^c \phi(x_i^s) - \frac{1}{n_t^c} \sum_{j=1}^{n_t^c} W_{tj}^c \phi(x_j^t) \right\|_H^2 \\
 &= \sum_{c=1}^C \left[\frac{1}{(n_s^c)^2} \sum_{i=1}^{n_s^c} \sum_{j=1}^{n_s^c} W_{ss}^c(i, j) K(x_i^s, x_j^s) + \frac{1}{(n_t^c)^2} \sum_{i=1}^{n_t^c} \sum_{j=1}^{n_t^c} W_{tt}^c(i, j) K(x_i^t, x_j^t) \right. \\
 &\quad \left. - \frac{2}{n_s^c n_t^c} \sum_{i=1}^{n_s^c} \sum_{j=1}^{n_t^c} W_{st}^c(i, j) K(x_i^s, x_j^t) \right], \tag{5}
 \end{aligned}$$

where w_{ss}^c represents the fuzzy similarity weight among class C samples in the source domain; w_{tt}^c represents the fuzzy similarity weight among class C samples in the target domain, and w_{st}^c is the fuzzy similarity weight between class C samples in the source and target domains. Additionally, Y_s denotes the source domain labels, and $\hat{Y}_t$ represents the predicted labels of the target domain.

A fuzzy set is an extension of a traditional set that describes imprecise concepts, where each element has a membership degree between 0 and 1, indicating the extent to which it belongs to a particular set. Common fuzzy set operations include union, intersection, and complement. Given a universal set X and a fuzzy subset A , the relationship between an element and the set is defined as follows:

$$\mu_A(x) : X \longrightarrow [0, 1], \tag{6}$$

where μ_A is the membership function of set A , $\mu_A(x)$, denotes the membership degree of element x in A . A value of $\mu_A(x)$ close to 1 indicates a high degree of membership, while a value close to 0 indicates a low degree of membership. Fuzzy similarity extends fuzzy set theory to measure the similarity between two matrices. Unlike conventional similarity measures, fuzzy similarity captures soft information among features, emphasizing fuzziness and uncertainty. In the practical training pipeline, let B denote the mini-batch size and C represent the total number of target classes (e.g., $B = 32$ and $C = 31$ for the Office-31 dataset). The source-domain samples are represented by a batch-level class-label matrix $S \in \mathbb{R}^{B \times C}$, constructed from one-hot encoded ground-truth labels, whereas the target-domain samples are represented by a batch-level class-probability matrix $T \in \mathbb{R}^{B \times C}$, obtained from the classifier’s predictions. The fuzzy composition is explicitly performed between the class-affiliation vectors of individual source-target sample pairs. Accordingly, the global pairwise fuzzy similarity matrix $W_{st} \in \mathbb{R}^{B \times B}$, is obtained through the fuzzy composition of

S and the transpose of T , denoted as $W_{st} = S \circ T^T$. For a specific source sample i and target sample j , the scalar similarity weight $W_{st}(i, j)$, which is utilized as the coupling coefficient in the adaptation loss, is computed as follows:

$$W_{st}(i, j) = \bigcup_{k=1}^C (s_{ik} \cap t_{jk}). \quad (7)$$

The fuzzy intersection operator ($\cap$, T-norm) measures the degree of semantic commonality between two membership values. In this framework, either the standard minimum operator or the algebraic product is employed:

$$s_{ik} \cap t_{jk} = \begin{cases} \min(s_{ik}, t_{jk}), & \text{for Standard Intersection} \\ s_{ik} \cdot t_{jk}, & \text{for Algebraic Product} \end{cases} \quad (8)$$

Subsequently, the fuzzy union operator ($\cup$, T-conorm) aggregates these element-wise intersection results across all C classes to quantify the overall semantic overlap between the two distinct samples. Depending on the selected fuzzy composition scheme, this class-wise aggregation is defined as:

$$\bigcup_{k=1}^C x_k = \begin{cases} \max_{k=1}^C (x_k), & \text{for Standard Max-Min Composition} \\ \bigoplus_{k=1}^C (x_k), & \text{for Algebraic Product-Sum Composition} \end{cases} \quad (9)$$

where $\bigoplus$ denotes the iterative algebraic sum operator (i.e., $x_1 \oplus x_2 = x_1 + x_2 - x_1 \cdot x_2$). Consequently, the fuzzy composition successfully transforms the class-wise membership distributions into a precise scalar similarity coefficient $W_{st}(i, j) \in [0, 1]$, which directly quantifies the cross-domain alignment between the i -th source sample and the j -th target sample. Therefore, although the composition aggregation operates along the class dimension C , the resulting matrix W_{st} inherently characterizes the pairwise source-target structural relationships within a mini-batch, yielding a beautifully aligned dimension of $B \times B$.

2.1.3 Fuzzy C-means

Fuzzy C-Means (FCM) is a fuzzy clustering algorithm that operates similarly to traditional clustering methods. However, conventional clustering techniques strictly assign each sample to a single category, which may lead to the loss of certain relevant information. In contrast, FCM allows samples to belong to multiple clusters simultaneously, representing their association with each cluster using membership degrees. Compared to traditional clustering methods, FCM is more suitable for handling data with ambiguous boundaries or uncertainty. In our model, we incorporate FCM to enhance pseudo-label confidence while accounting for the uncertainty of target samples. The FCM algorithm optimizes an objective function by iteratively updating membership degrees and cluster centers. The objective function is defined as follows:

$$J_m(U, V) = \sum_{j=1}^N \sum_{i=1}^C |\mu_{ij}|^m \|x_j - c_i\|^2, \quad (10)$$

where N represents the number of samples, C is the number of cluster centers, x_j denotes a sample, and c_i represents a cluster center. The term μ_{ij} is the membership degree of sample j with respect to cluster center i , and m is the fuzziness coefficient, typically set to $m > 1$.

The algorithm begins with the random initialization of parameters, including the number of clusters (C), the fuzziness coefficient (m), and the membership matrix (U). The membership matrix has a size of $N \times C$, where each row must satisfy the constraint $\sum_{i=1}^C \mu_{ij} = 1$. For instance, suppose we have 10 data points, denoted as $x_1 \sim x_{10}$, and two cluster centers c_1 and c_2 . Figure 3 illustrates the relationship between the data points and the cluster centers, with membership degrees (μ_{ij}) representing their association. Figure 3(a) shows the membership degrees of each data point with respect to the first cluster center, while Figure 3(b) shows their membership degrees for the second cluster center. Each cluster center assigns a membership degree to every data point, and conversely, each data point has two membership degrees. For any given data point, the sum of its membership degrees across all cluster centers equals 1. The membership degree for each sample is computed using the following formula:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^C \left\| \frac{x_j - c_i}{x_j - c_k} \right\|^{\frac{2}{m-1}}}, \quad (11)$$

where $\|x_j - c_i\|$ represents the distance between data point x_j and cluster center c_i , and m is the fuzziness coefficient.

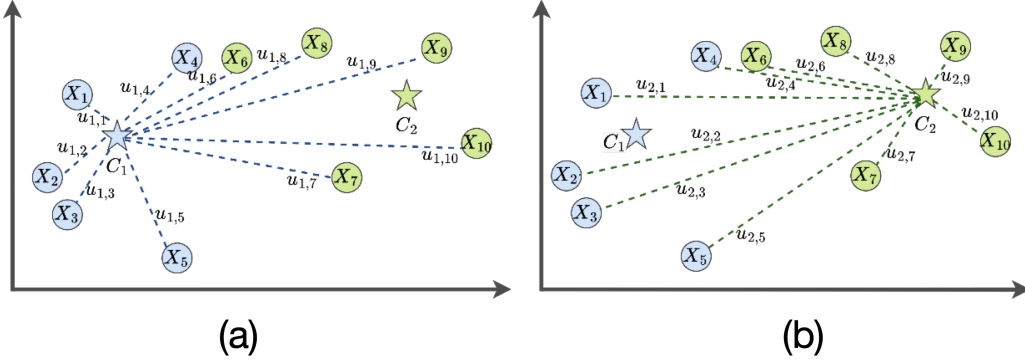


Figure 3: Relationship between data points and cluster centers. (a) Membership degrees of data points for the first cluster center, (b) Membership degrees of data points for the second cluster center.

The cluster centers in FCM are computed based on all data points and their respective membership degrees. The cluster centers are updated iteratively using the new membership matrix, as shown in Figure 4. The update formula for the cluster centers is given by:

$$c_i = \frac{\sum_{j=1}^N \mu_{ij}^m \cdot x_j}{\sum_{j=1}^N \mu_{ij}^m}, \quad (12)$$

where x_j represents a data point, μ_{ij}^m is the membership degree, and N is the total number of samples.

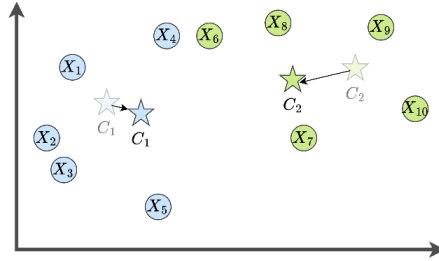


Figure 4: Updated cluster center positions.

2.2 DFS-LDAN learning algorithm

This section provides a detailed explanation of the training process for DFS-LDAN, as illustrated in Figure 5. The training framework consists of two main components: generating source domain cluster centers (yellow block) and DFS-LDAN training (purple block).

First, the detailed training steps for the yellow block are shown in Algorithm 1. The model takes the labeled source domain data (x_s, y_s) as input and extracts features Z_s using ResNet-50 [9]. These features are then fed into the classifier to generate predicted labels $\hat{y}_s$ for the source domain. The predicted labels are compared with the ground truth labels y_s to compute the cross-entropy loss L_{ce} , which is used to optimize the classification performance of the model. The cross-entropy loss is defined as follows:

$$L_{ce} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \sum_{c=1}^C y_{s,i}^{(c)} \log(\hat{y}_{s,i}^{(c)}), \quad (13)$$

where $y_{s,i}^{(c)}$ represents the ground-truth label of the i -th source domain sample with respect to the k -th class, $\hat{y}_{s,i}^{(c)}$ denotes the corresponding predicted class probability outputted by the model, n_s is the total number of source samples, and C is the total number of classes.

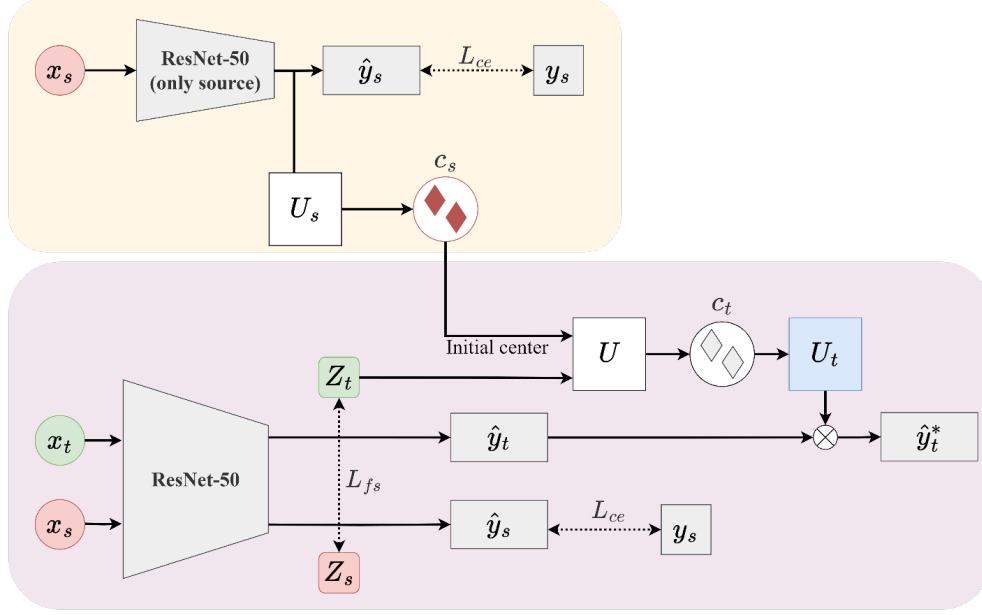


Figure 5: Training flowchart of DFS-LDAN.

Algorithm 1 Generation of source-domain cluster centers

-
- 1: **Input:** Source data $\{X_s, Y_s\}$.
 - 2: **Initialize parameters:** Backbone $F_s(\cdot)$, Cluster centers c_i .
 - 3: **for** $e \leq \text{epoch}$ **do**
 - 4: $\{\hat{y}_s, Z_s\} \leftarrow F_s(X_s)$
 - 5: Calculate cross-entropy loss L_{ce} as in Eq. (13).
 - 6: Update network weights $\hat{\theta}_{F_s}$ minimizing L_{ce} .
 - 7: **end for**
 - 8: **for** $e \leq \text{epoch}$ **do**
 - 9: Calculate fuzzy membership matrix $U_s \leftarrow \text{FCM}(c_i, Z_s)$ as in Eq. (11).
 - 10: Calculate updated cluster centers $c_i \leftarrow \text{Update}(Z_s, U_s)$ as in Eq. (12).
 - 11: **end for**
 - 12: **Output:** Source domain cluster centers c_s .
-

To optimize the network parameters, the objective function is formulated as follows:

$$\hat{\theta} = \arg \min_{\theta_F} L_{ce}(\theta_F), \quad (14)$$

where $\hat{\theta}$ represents the optimized parameters obtained through training, $\arg \min_{\theta_F}$ indicates minimizing the parameters of the feature extractor θ_F and L_{ce} is the cross-entropy loss.

Furthermore, based on the extracted source domain features Z_s and the randomly initialized cluster centers c_i , we compute the membership matrix U_s , which represents the membership degree of each source domain sample for every class. This matrix is then used to update the source domain cluster centers. Finally, the generated source domain cluster centers c_s serve as the initialization points for the target domain cluster centers, accelerating the convergence speed of the domain adaptation network in learning the target domain clusters. For the DFS-LDAN training process (purple block), the detailed training steps are outlined in Algorithm 2. The labeled source domain data $\{X_s, Y_s\}$ and the unlabeled target domain data X_t are fed into ResNet-50 to extract features $\{Z_s, Z_t\}$. Simultaneously, the final-layer features $\{Z_s, Z_t\}$ extracted from the backbone network and the source domain cluster centers c_s obtained from the yellow block are input into the FCM algorithm to compute the membership matrix U . This process updates the target domain cluster centers c_t . Next, the membership matrix U_t is computed based on the target domain features and used as weights for the predicted outputs of the target domain, formulated as follows:

Algorithm 2 Training of the DFS-LDAN model

-
- 1: **Input:** Source data $\{X_s, Y_s\}$, Target data $\{X_t\}$, Pre-trained source domain cluster centers c_s .
 - 2: **Initialize parameters:** Feature extractor and classifier backbone $F(\cdot)$, Target domain cluster centers $c_t \leftarrow c_s$.
 - 3: **for** $e \leq \text{epoch}$ **do**
 - 4: Extract predictions and backbone features simultaneously:
 $\{\hat{y}_s, Z_s\}, \{\hat{y}_t, Z_t\} \leftarrow F(X_s, X_t)$
 - 5: Calculate source domain cross-entropy loss L_{ce} as in Eq. (13).
 - 6: Compute comprehensive membership matrix via FCM: $U \leftarrow \text{FCM}(c_t, Z_t)$ as in Eq. (11).
 - 7: Update target domain cluster centers iteratively: $c_t \leftarrow \text{Update}(Z_t, U)$ as in Eq. (12).
 - 8: Extract target-specific fuzzy membership matrix $U_t \leftarrow \text{FCM}(c_t, Z_t)$ as in Eq. (11).
 - 9: Calibrate target predicted outputs: $\hat{y}_t^* \leftarrow \text{Calibrate}(\hat{y}_t, U_t)$ as in Eq. (15).
 - 10: Calculate fuzzy similarity-guided loss L_{fs} using Eq. (16).
 - 11: Compute total optimization objective loss: $L = L_{ce} + \alpha L_{fs}$ as in Eq. (17).
 - 12: Update entire network weights $\hat{\theta}_F$ via backpropagation minimizing L .
 - 13: **end for**
 - 14: **Output:** Optimally trained network parameters $\hat{\theta}_F$.
-

$$\hat{y}_t^* = \hat{y}_t \odot U_t, \quad (15)$$

where $\hat{y}_t \in \mathbb{R}^{n_t \times C}$ denotes the predicted class-probability matrix, $U_t \in \mathbb{R}^{n_t \times C}$ denotes the fuzzy membership matrix, and $\odot$ represents the element-wise Hadamard product. The resulting calibrated prediction matrix $\hat{y}_t^* \in \mathbb{R}^{n_t \times C}$ preserves the same dimensionality as the original prediction matrix while successfully incorporating uncertainty-aware fuzzy weighting information.

This paper applies Fuzzy Similarity Local Domain Adaptation (FSLDA) to measure the distribution discrepancy between the source and target domains using Equation (5). This encourages the model to learn domain-invariant features. The domain adaptation loss function is defined as follows:

$$L_{fs} = \text{FSLDA}(X_s, X_t, Y_s, \hat{Y}_t^*), \quad (16)$$

where X_s and X_t are the source and target domain data, respectively; Y_s is the source domain label, and $\hat{Y}_t^*$ is the weighted prediction result of the target domain. FSLDA computes the domain discrepancy using the local domain adaptation formula.

The overall loss function of the model consists of the classification loss L_{ce} and the fuzzy similarity local domain adaptation loss L_{fs} , formulated as follows:

$$L = L_{ce} + \lambda L_{fs}, \quad (17)$$

where λ is the penalty coefficient for fuzzy similarity local domain adaptation.

To train the model and reduce the domain discrepancy, we minimize both the classification loss and the fuzzy similarity local domain adaptation loss. The optimization objectives are expressed as follows:

$$\hat{\theta}_F = \arg \min_{\theta_F} (L_{ce}(\theta_F) + \lambda L_{fs}(\theta_F)), \quad (18)$$

where $\hat{\theta}_F$ represents the optimized parameters obtained through training, $\arg \min$ denotes minimizing the parameters of the feature extractor, L_{ce} is the cross-entropy loss function, and L_{fs} is the fuzzy similarity local domain adaptation loss function.

2.3 Theoretical convergence analysis

In this section, we give a formal theoretical analysis of proposed method in terms of the $\mathcal{H}\Delta\mathcal{H}$ -distance, which is widely used in the statistical learning theory of the unsupervised domain adaptation. Formally, for a given hypothesis set $\mathcal{H}$, the discrepancy distance between the source domain distribution $\mathcal{D}_s$ and the target domain distribution $\mathcal{D}_t$ is defined as:

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) = 2 \sup_{h_1, h_2 \in \mathcal{H}} |\mathbb{P}_{x \sim \mathcal{D}_s}[h_1(x) \neq h_2(x)] - \mathbb{P}_{x \sim \mathcal{D}_t}[h_1(x) \neq h_2(x)]|. \quad (19)$$

Using this notion, one can obtain a probabilistic generalization bound on the expected risk $\epsilon_t(h)$ of a classifier h evaluated on the target domain, given its empirical risk $\epsilon_s(h)$ on the source domain:

$$\epsilon_t(h) \leq \epsilon_s(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) + \lambda^*, \quad (20)$$

where λ^* represents the error of the ideal joint hypothesis that minimizes the combined risk across both domains. To establish the connection between our method and Equation (20), we analyze the fine-grained sub-domain structures. In a multi-class setting, global alignment often collapses due to conditional distribution shifts. Therefore, we decompose the total discrepancy into class-conditional $\mathcal{H}\Delta\mathcal{H}$ -distances, weighted by their cross-domain semantic alignments. Let $\mathcal{H}_K$ be the Reproducing Kernel Hilbert Space (RKHS) endowed with a characteristic kernel $K(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}_K}$. If we define the hypothesis family $\mathcal{H}$ as the unit ball in the RKHS (i.e., $\|h\|_{\mathcal{H}_K} \leq 1$), by leveraging the basic property of MMD theory, the supremum over the function space can be tightly upper-bounded by the distance of the mean embedding:

$$\sup_{h \in \mathcal{H}} |\mathbb{E}_{x \sim \mathcal{D}_s^c}[h(x)] - \mathbb{E}_{x \sim \mathcal{D}_t^c}[h(x)]| = \|\mathbb{E}_{x \sim \mathcal{D}_s^c}[\phi(x)] - \mathbb{E}_{x \sim \mathcal{D}_t^c}[\phi(x)]\|_{\mathcal{H}_K}. \quad (21)$$

In our framework, the proposed FSLDA acts as the empirical, fuzzy-weighted estimator of these localized RKHS distances across all C classes. By incorporating the bounded pairwise fuzzy similarity weights $W(i, j) \in [0, 1]$ to weight the individual sample pairs within each class c , the global $\mathcal{H}\Delta\mathcal{H}$ -distance is strictly upper-bounded by our metric target:

$$\frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) \leq \gamma \sum_{c=1}^C \left\| \frac{1}{n_s^c} \sum_{i=1}^{n_s^c} W_{si}^c \phi(x_i^s) - \frac{1}{n_t^c} \sum_{j=1}^{n_t^c} W_{tj}^c \phi(x_j^t) \right\|_{\mathcal{H}_K} + \epsilon_{\text{residual}}, \quad (22)$$

where γ is a positive scaling constant associated with the kernel profile, and $\epsilon_{\text{residual}}$ represents the statistical estimation error that converges to zero at a rate of $\mathcal{O}\left(\frac{1}{\sqrt{n_s}} + \frac{1}{\sqrt{n_t}}\right)$ as the sample size increases. By squaring the RKHS norm via the kernel trick, we show that minimizing the FSLDA loss directly minimizes the empirical upper bound of the $\mathcal{H}\Delta\mathcal{H}$ -distance. Consequently, the optimization process is theoretically guaranteed to converge, thereby reducing the target-domain risk $\epsilon_t(h)$ toward its minimum and promoting robust cross-domain generalization.

3 Experimental results

This section evaluates the proposed DFS-LDAN through experiments on multiple widely used domain adaptation datasets to assess its effectiveness and generalization capability. Section 3.1 details the experimental setup, including the system environment and datasets. Section 3.2 presents the experimental results alongside comparative analyses with existing methods. Finally, an ablation study is conducted to examine the impact of key components within DFS-LDAN.

3.1 Datasets

The proposed method is evaluated on three commonly used domain adaptation datasets. The detailed dataset information is shown in Table 1.

Office-31 [26] is a fundamental dataset for domain adaptation. It consists of three different subdomains: Amazon (A), Webcam (W), and DSLR (D), all of which contain real-world images. Each subdomain includes 31 categories covering common office supplies, such as computers, scissors, and pens. The dataset contains a total of 4,110 images, but the number of samples per class is imbalanced. The domain adaptation tasks are defined as six transfer scenarios: $A \rightarrow W$, $A \rightarrow D$, $W \rightarrow A$, $W \rightarrow D$, $D \rightarrow A$, and $D \rightarrow W$. Figure 6 illustrates examples of the same category across different subdomains.

ImageCLEF-DA [1] is a benchmark dataset derived from the ImageCLEF 2014 domain adaptation challenge. It includes three subdomains: Caltech-256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC 2012 (P). Each subdomain contains 12 categories, with 50 images per category, making it a well-balanced domain adaptation dataset. The defined transfer tasks include six scenarios: $C \rightarrow I$, $C \rightarrow P$, $I \rightarrow C$, $I \rightarrow P$, $P \rightarrow C$, and $P \rightarrow I$. Figure 7 shows examples of the same category across different subdomains.

Office-Home [28] is another domain adaptation dataset, featuring greater domain discrepancies compared to Office-31. The dataset includes both real-world images and cartoon-style illustrations, offering more diverse and challenging scenarios. It consists of 65 categories covering daily objects such as chairs, bookshelves, and spoons. The dataset spans four subdomains: Art (A), Clipart (C), Product (P), and Real World (R), with a total of 15,588 images (approximately

Table 1: Detailed statistics of the evaluation datasets.

Datasets	Classes	Domains	Samples	Total
Office-31 [26]	31	Amazon (A)	2,817	4,110
		Webcam (W)	795	
		DSLr (D)	498	
ImageCLEF-DA [1]	12	Caltech (C)	600	1,800
		ImageNet (I)	600	
		Pascal (P)	600	
Office-Home [28]	65	Art (A)	2,427	15,588
		Clipart (C)	4,365	
		Product (P)	4,439	
		Real world (R)	4,357	

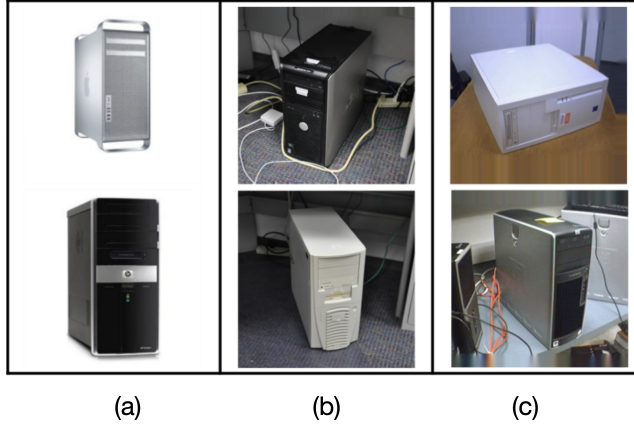


Figure 6: Office-31 dataset. (a) Amazon (b) DSLR (c) Webcam.

70 images per category). The defined transfer tasks include 12 scenarios: $A \rightarrow C$, $A \rightarrow P$, $A \rightarrow R$, $C \rightarrow A$, $C \rightarrow P$, $C \rightarrow R$, $P \rightarrow A$, $P \rightarrow C$, $P \rightarrow R$, $R \rightarrow A$, $R \rightarrow C$, and $R \rightarrow P$. Figure 8 illustrates examples of the same category across different subdomains.

3.2 Experimental setup

The experiments were conducted on a workstation with the hardware specifications listed in Table 2. The software environment consists of Python for development, with PyTorch 1.8.0 as the primary deep learning framework.

Table 2: Hardware configuration of the experimental platform.

Component	Specification
CPU	Intel(R) Core(TM) i7-14700 @ 2.10GHz
Memory	32 GB
GPU	NVIDIA GeForce RTX 3050 (8 GB)

During the training process, to ensure a fair comparison with existing methods, we adopted ResNet-50 [9], pretrained on the ImageNet dataset, as the backbone network and initialized it with the corresponding pretrained weights. Following the final average pooling layer, and in accordance with common practices in domain adaptation studies [11, 27], a hidden layer consisting of 256 neurons was introduced to project the extracted features into a compact latent representation. This hidden-layer design reduces the feature dimensionality, facilitates the computation of distribution alignment losses, and achieves an effective balance between feature discriminability and computational efficiency. Finally, a fully

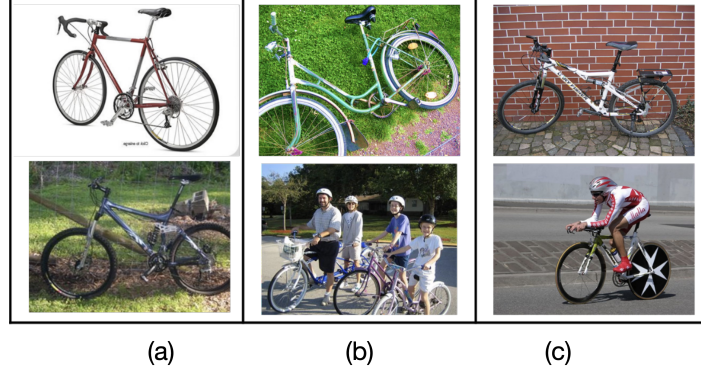


Figure 7: ImageCLEF-DA dataset. (a) Caltech-256 (b) ImageNet ILSVRC 2012 (c) Pascal VOC 2012.

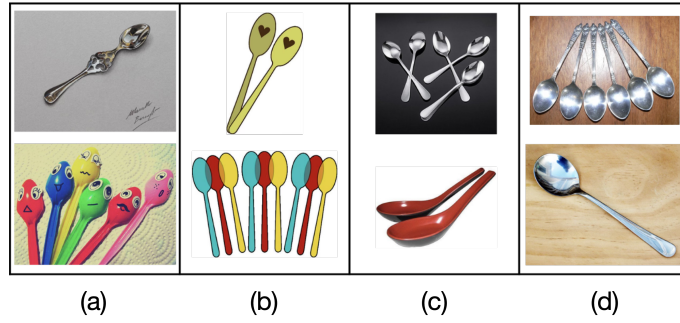


Figure 8: Office-Home dataset. (a) Art (b) Clipart (c) Product (d) Real world.

connected layer is connected to a classifier, where the number of output units corresponds to the number of dataset categories, performing image classification and generating the final predictions.

The detailed training configurations and hyperparameter settings are summarized in Table 3. All input images are resized to 224×224 pixels, and the mini-batch size is set to 32. Model optimization is performed using Stochastic Gradient Descent (SGD). To preserve pre-trained feature representations while accelerating classifier convergence, a differential learning rate strategy is employed: the backbone network is assigned a baseline learning rate of 0.001, whereas the classifier layer is configured with a learning rate ten times larger (i.e., 0.01). The entire network is trained for 200 epochs. For the local domain adaptation components, the loss weight (λ) balancing the fuzzy similarity objective is set to 0.5. Additionally, the Fuzzy C-Means (FCM) clustering module is employed in an offline preprocessing stage to generate source-domain cluster centers. The fuzziness coefficient is set to $m = 2$, and the maximum number of FCM iterations is fixed at 100. The resulting cluster centers are subsequently kept fixed and used throughout the target-domain training process without further FCM updates.

Table 3: Hyperparameters and training configurations of the proposed model.

Parameters	Value
Input image size	224×224
Batch size	32
Optimizer	SGD
Learning rate (Backbone / Classifier)	[0.001, 0.01]
Total training epochs	200
Fuzzy loss weight (λ)	0.5
Fuzziness coefficient (m)	2
Maximum FCM iterations	100

Table 4: Classification accuracy (%) on Office-31 domain adaptation tasks.

Methods	A→W	D→W	W→D	A→D	D→A	W→A	Avg (%)
ResNet [9]	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DDC [27]	75.8	95.0	98.2	77.5	67.4	64.0	79.7
DAN [16]	83.8	96.8	99.5	78.4	66.7	62.7	81.3
DANN [3]	82.0	96.9	99.1	79.7	68.2	67.4	82.2
Conditional DAN [17]	94.1	98.6	100.0	92.9	71.0	69.3	87.7
TADA [31]	94.3	98.7	99.8	91.6	72.9	73.0	88.4
DSAN [39]	93.6	98.3	100.0	90.2	73.5	74.8	88.4
RDAOT [4]	94.1	99.0	100.0	89.6	76.5	74.4	88.9
MUDA [12]	88.2	98.7	99.8	90.0	71.2	69.0	86.1
DANN+FIA [29]	92.6	98.6	100	85.4	74.2	73.7	87.4
DFS-LDAN (Min-Max)	93.2	98.4	100.0	90.8	72.2	72.2	87.8 ± 0.09
DFS-LDAN (Product-Sum)	94.2	98.9	100.0	91.8	75.8	73.9	89.1 ± 0.07

Methods (Ours Only)	Avg Recall	Avg F1	Training Time	Memory Usage
DFS-LDAN (Min-Max)	85.7 ± 0.1	87.1 ± 0.08	1 h 26 min	7.52 G
DFS-LDAN (Product-Sum)	87.6 ± 0.07	88.0 ± 0.04	1 h 28 min	7.53 G

3.3 Experimental results of DFS-LDAN

In this experiment, we evaluate the performance of the proposed DFS-LDAN model using three different datasets and compare its accuracy with other methods. Accuracy is defined as follows:

$$\text{Accuracy} = \frac{\text{Number of correctly classified samples}}{\text{Total number of samples}} \quad (23)$$

The experimental setting is denoted as A→W. The proposed DFS-LDAN method employs two common fuzzy set operations: Minimum-Maximum (Min-Max) and Product-Sum, which are denoted as DFS-LDAN (Min-Max) and DFS-LDAN (Product-Sum), respectively. The subsequent analysis presents the experimental results using these notations. Each experiment was independently repeated five times. The reported results are presented as average accuracy ± standard deviation. Table 4 summarizes the experimental results of DFS-LDAN on the Office-31 dataset, compared with other domain adaptation methods. The ResNet baseline [9], which does not incorporate domain adaptation, achieves relatively low performance across all domain shifts, with an average accuracy of 76.1%. This highlights the challenge of handling domain discrepancies solely with a pretrained network. Traditional domain adaptation methods, such as DDC [27], DAN [16], and DANN [3], align global data distributions using distribution distance minimization or adversarial learning. For instance, DAN [8] achieves an average accuracy of 81.3%. More recent approaches, such as TADA [31], RDAOT [4], and MUDA [12], further improve performance compared to traditional methods. Specifically, RDAOT [4] leverages Optimal Transport (OT) theory with the Wasserstein distance to perform local domain adaptation, achieving an average accuracy of 88.9%. Our proposed DFS-LDAN (Min-Max) and DFS-LDAN (Product-Sum) demonstrate high stability across different domain adaptation tasks, with DFS-LDAN achieving an average accuracy of 89.1%, surpassing existing methods.

Table 5 presents the experimental results of DFS-LDAN on the ImageCLEF-DA dataset. Traditional methods such as DDC [27] and DAN [16] improve upon the ResNet baseline [9], reaching an average accuracy of 83.4%, indicating that deep feature-based distribution alignment effectively mitigates domain shift-induced performance degradation. Our proposed DFS-LDAN (Product-Sum) exhibits notable advantages in the P→I and C→I tasks, where the product-sum fuzzy set operation achieves superior domain adaptation performance. Overall, DFS-LDAN (Product-Sum) achieves the highest average accuracy of 90.4%.

Table 6 presents the results on the Office-Home dataset. The TADA [31] method, which integrates an attention mechanism to enhance domain-invariant feature learning, achieves an average accuracy of 67.6%, with particularly strong performance in A→P, A→R, C→P, and C→R tasks. Our proposed DFS-LDAN (Min-Max) achieves an average accuracy of 68.4%, outperforming RDAOT [4] and other existing methods across multiple domain adaptation scenarios.

To intuitively demonstrate the effectiveness of the proposed fuzzy similarity-guided local domain adaptation network, we employ t-SNE to visualize the feature distributions extracted before and after adaptation. Figure 9 illustrates

Table 5: Classification accuracy (%) on ImageCLEF domain adaptation tasks.

Methods	I→P	P→I	I→C	C→I	C→P	P→C	Avg (%)
ResNet [9]	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DDC [27]	74.6	85.7	91.1	82.3	68.3	88.8	81.8
DAN [16]	75.0	86.2	93.3	84.1	69.8	91.3	83.3
DANN [3]	75.0	86.0	96.2	87.0	74.3	91.5	85.0
Conditional DAN [17]	77.7	90.7	97.7	91.3	74.2	94.3	87.7
DSAN [39]	80.2	93.3	97.2	93.8	80.8	95.9	90.2
RDAOT [4]	80.4	93.3	97.5	93.7	82.1	94.5	90.2
DFS-LDAN (Min-Max)	79.2	92.3	96.5	93.8	79.5	95.3	88.3 ± 0.62
DFS-LDAN (Product-Sum)	80.0	93.8	97.3	94.5	80.8	95.8	90.4 ± 0.43

Methods (Ours Only)	Avg Recall	Avg F1	Training Time	Memory Usage
DFS-LDAN (Min-Max)	88.5 ± 0.54	89.4 ± 0.82	28 min	7.31 G
DFS-LDAN (Product-Sum)	89.6 ± 0.51	90.0 ± 0.74	30 min	7.33 G

Table 6: Classification accuracy (%) on Office-Home domain adaptation tasks.

Methods	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg (%)
ResNet [9]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [16]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN [3]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	58.3
Conditional DAN [17]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
TADA [31]	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
DSAN [39]	54.4	70.8	75.4	60.4	67.8	68.0	62.6	55.9	78.5	73.8	60.6	83.1	67.6
RDAOT [4]	55.1	70.2	75.5	61.2	69.1	69.2	64.1	55.6	78.8	73.6	60.9	83.7	68.1
DANN+FIA [29]	53.7	65.9	75.0	59.3	67.3	69.5	61.4	54.0	79.1	72.4	61.0	82.0	66.7
DFS-LDAN (Min-Max)	55.2	70.9	75.6	61.9	70.1	69.2	62.8	56.8	78.9	74.0	61.2	84.1	68.4 ± 0.23
DFS-LDAN (Product-Sum)	54.5	70.0	75.6	61.3	69.0	68.3	63.2	56.1	78.0	73.5	61.0	83.2	67.8 ± 0.16

Methods (Ours Only)	Avg Recall	Avg F1	Training Time	Memory Usage
DFS-LDAN (Min-Max)	63.2 ± 0.4	64.1 ± 0.3	4 h	7.22 G
DFS-LDAN (Product-Sum)	65.5 ± 0.3	66.6 ± 0.2	4 h 4 min	7.25 G

the comparative visualizations across three representative transfer tasks: Amazon to Webcam ($A \rightarrow W$) on Office-31, ImageNet to Pascal ($I \rightarrow P$) on ImageCLEF-DA, and Art to Clipart ($A \rightarrow C$) on Office-Home. As observed in the left column of Figure 9, the initial features extracted by the source-only baseline model are highly scattered and heavily entangled, indicating a severe domain shift where boundaries between different classes are profoundly ambiguous. Conversely, the right column vividly illustrates the transformative impact of our DFS-LDAN framework. For the Office-31 and ImageCLEF-DA datasets, the post-adaptation features are tightly condensed into distinct, highly separable clusters. This compelling structural improvement confirms that our fuzzy similarity mechanism successfully aligns local domain features while simultaneously preserving the intrinsic semantic discrimination of the classification task. Furthermore, the visualization of the Office-Home $A \rightarrow C$ task provides important insights into the failure cases of the proposed method. Although the post-adaptation features in this highly imbalanced 65-class scenario exhibit a noticeable improvement in overall structural organization compared with their initially chaotic distribution, the resulting clusters remain relatively dispersed compared with those observed on the less challenging datasets. In such a densely populated 65-class latent space, even minor ambiguities in the fuzzy weights can cause semantically similar categories to overlap and cross-contaminate in the feature space, thereby preventing the network from learning well-defined decision boundaries. These observations suggest that achieving highly discriminative feature representations under extreme style shifts remains a key limitation of the current framework.

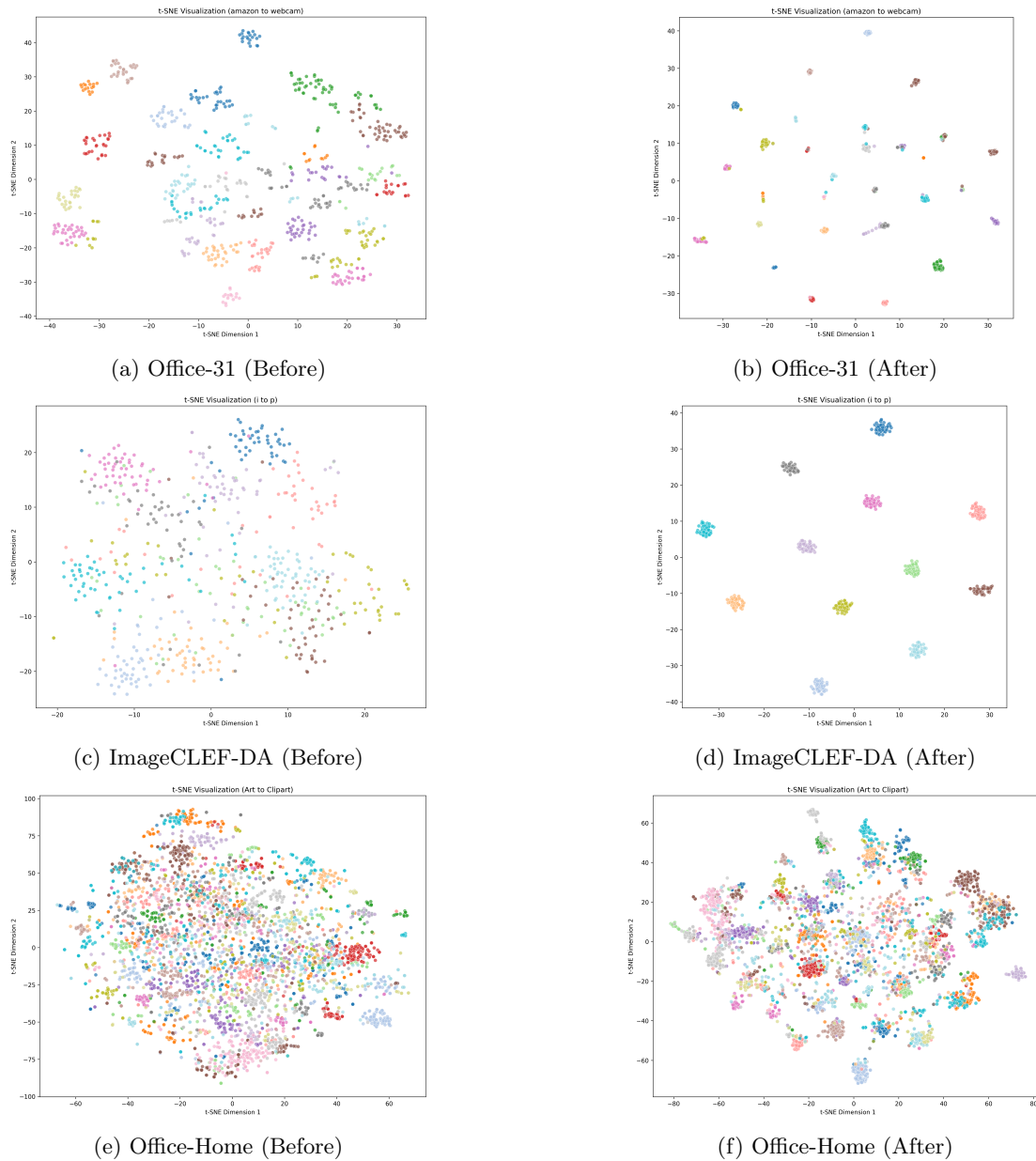


Figure 9: The t-SNE visualizations of feature representations before and after domain adaptation using the proposed DFS-LDAN framework across three benchmark datasets.

Table 7: Ablation study on the Office-Home dataset for domain adaptation.

Methods	FCM	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg (%)
DFS-LDAN (Min-Max)	×	54.6	70.4	75.4	61.1	70.4	69.5	62.8	55.9	78.6	73.9	60.7	83.6	68.1 ± 0.41
	✓	55.1	70.9	75.6	61.9	70.1	69.2	62.8	56.8	78.9	74.0	61.0	84.1	68.4 ± 0.23
DFS-LDAN (Product-Sum)	×	54.9	69.5	75.0	60.3	68.4	68.1	61.0	55.4	76.5	73.6	60.3	83.5	67.2 ± 0.19
	✓	54.5	70.0	75.6	61.3	69.0	68.3	63.2	56.1	78.0	73.5	61.2	83.2	67.8 ± 0.16

3.4 Ablation study

To verify the effectiveness of the FCM weighting in the proposed DFS-LDAN method, we conducted an ablation study on the Office-Home dataset. The results are summarized in Table 7. This experiment includes two different fuzzy set operations: (1) Minimum-Maximum (Min-Max) operation, denoted as DFS-LDAN (Min-Max). (2) Product-Sum operation, denoted as DFS-LDAN (Product-Sum). Additionally, the notation FCM in the table indicates whether the Fuzzy C-Means (FCM) weighting is applied. From the results, we observe that incorporating FCM weighting leads to a marginal improvement on average, accompanied by some task-specific degradation in certain variants (e.g., $A \rightarrow C$ for Product-Sum, and $C \rightarrow P$ for Min-Max). Specifically, the overall average accuracy increases by 0.3% to 0.6%. This indicates that while FCM weighting may introduce minor performance fluctuations in individual tasks, it generally exerts a net positive impact on stabilizing and enhancing the overall cross-domain adaptation performance.

3.5 Sensitivity analysis of the penalty coefficient λ and fuzzy operator evaluation

To evaluate the robustness of the proposed framework and justify our parameter selection, we conducted a comprehensive sensitivity analysis for the penalty coefficient λ . The hyperparameter λ plays a critical role in balancing the primary task classification loss and the fuzzy similarity-guided local domain adaptation constraint. A lower λ value implies insufficient constraint on domain discrepancy, which may lead to under-alignment of local feature distributions. Conversely, an excessively large λ overemphasizes domain alignment, potentially disrupting the intrinsic discriminative boundaries learned by the classification objective. We evaluated both variants of our model, namely DFS-LDAN (Min-Max) and DFS-LDAN (Product-Sum), across all three benchmark datasets by varying $\lambda \in \{0.3, 0.5, 0.7\}$. The average accuracy results are compiled in Table 8.

Table 8: Sensitivity analysis of average accuracy (%) across different λ values.

Dataset	Methods	$\lambda = 0.3$	$\lambda = 0.5$	$\lambda = 0.7$
Office-31	DFS-LDAN (Min-Max)	87.14	87.80	86.87
	DFS-LDAN (Product-Sum)	87.24	89.10	87.31
ImageCLEF-DA	DFS-LDAN (Min-Max)	88.21	88.30	87.40
	DFS-LDAN (Product-Sum)	89.15	90.40	88.90
Office-Home	DFS-LDAN (Min-Max)	66.91	68.40	66.80
	DFS-LDAN (Product-Sum)	66.49	67.80	67.20

As illustrated in Table 8, the accuracy exhibits a clear convex trend, consistently reaching its maximum at $\lambda = 0.5$ across all three datasets. Notably, this value achieves an effective balance between classification learning and domain alignment, despite the substantial differences among the datasets, ranging from the relatively small-scale ImageCLEF-DA dataset with 12 classes to the large-scale Office-Home dataset with 65 classes. Deviating from this optimal value leads to noticeable performance degradation. For example, reducing λ to 0.3 weakens the domain alignment effect, resulting in an accuracy decrease of up to 1.86% for the DFS-LDAN (Product-Sum) model on the Office-31 dataset. Conversely, increasing λ to 0.7 imposes excessive alignment constraints on the feature space, causing a performance drop of up to 1.60% for the DFS-LDAN (Min-Max) model on the Office-Home dataset. Therefore, $\lambda=0.5$ was adopted as the default setting in all experiments to ensure a favorable trade-off between adaptation effectiveness and classification performance.

Beyond the impact of the penalty coefficient, Table 8 also reveals a compelling performance divergence between the two fuzzy operations across different datasets, which can be fundamentally explained by their mathematical properties in relation to dataset complexity. The empirical results demonstrate that the Product-Sum variant achieves superior performance on the Office-31 and ImageCLEF-DA datasets, whereas the Min-Max variant excels on the Office-Home

dataset. The Product-Sum operators calculate fuzzy intersections and unions algebraically, providing a smooth and proportional aggregation of fuzzy memberships that ensures all feature contributions are considered simultaneously during gradient calculation. Such continuous and comprehensive feature fusion is highly beneficial for datasets with fewer classes and relatively balanced distributions, such as Office-31 and ImageCLEF-DA, allowing the model to learn smoother and more precise global domain alignments. Conversely, the Min-Max operators evaluate similarities by strictly selecting extreme values, a property that inherently acts as a robust hard filter against noisy features and prevents dominant major classes from overwhelmingly washing out minor classes during the alignment process. The Min-Max variant demonstrates a distinct advantage on the Office-Home dataset, which is characterized by a higher structural complexity, a larger number of categories, and severe class imbalance. By focusing strictly on the strongest similarity links and enforcing the tightest logical constraints, the Min-Max operations provide enhanced robustness against the noise and imbalanced distributions prevalent in highly complex domain adaptation scenarios.

4 Conclusions

This paper proposes a deep learning model, DFS-LDAN, which integrates fuzzy similarity and Fuzzy C-Means (FCM) for unsupervised domain adaptation in image classification. By combining fuzzy theory with domain adaptation techniques, the proposed model enhances recognition performance in the target domain while effectively reducing distribution discrepancies between the source and target domains. The main contributions of this work are summarized as follows:

1. Effective local domain adaptation using fuzzy similarity: The proposed DFS-LDAN model leverages fuzzy similarity to perform local domain adaptation by minimizing the Maximum Mean Discrepancy (MMD). This allows the model to compute class-wise distances between the source and target domains, guiding it to learn domain-invariant features, thereby reducing distribution differences and improving cross-domain classification accuracy.
2. Incorporation of Fuzzy C-Means for adaptive feature learning: The introduction of FCM enables the model to evaluate the membership degree of features to each class. This flexibility allows the model to better handle uncertainty in data and retain essential target-domain-specific features during cross-domain adaptation, thereby improving generalization capability.
3. Extensive validation on benchmark datasets: The effectiveness of DFS-LDAN is validated on three public domain adaptation datasets—Office-31, ImageCLEF-DA, and Office-Home—through comparisons with multiple existing domain adaptation methods.

The experimental results demonstrate the superiority of DFS-LDAN. On the Office-31 dataset, DFS-LDAN achieves an accuracy of 89.1%. On the ImageCLEF-DA dataset, DFS-LDAN using the Minimum-Maximum (Min-Max) operation achieves 88.3%, while using the Product-Sum operation reaches 90.4%. On the Office-Home dataset, DFS-LDAN outperforms other methods across multiple adaptation tasks (e.g., $A \rightarrow C$, $C \rightarrow A$, $P \rightarrow C$, $P \rightarrow R$, $R \rightarrow A$, $R \rightarrow P$), with DFS-LDAN (Min-Max) achieving the highest accuracy of 68.4%. In future research, we plan to evaluate DFS-LDAN under different scenario constraints, such as semi-supervised domain adaptation, partial domain adaptation, and multi-source/multi-target domain adaptation tasks. Additionally, we will explore the integration of attention mechanisms (e.g., the TADA [31] approach) to enhance feature selection and improve adaptation performance.

References

- [1] B. Caputo, H. Müller, J. Martinez-Gomez, et al., *ImageCLEF 2014: Overview and analysis of the results*, Information Access Evaluation. Multilinguality, Multimodality, and Interaction. CLEF 2014. Lecture Notes in Computer Science, **8685**, Springer, Cham., (2014), 192-211. https://doi.org/10.1007/978-3-319-11382-1_18
- [2] M. Eftekhari, A. Mehrpooya, F. Saberi-Movahed, V. Torra, *How fuzzy concepts contribute to machine learning*, Studies in Fuzziness and Soft Computing, (2022). <https://doi.org/10.1007/978-3-030-94066-9>
- [3] Y. Ganin, E. Ustinova, H. Ajakan, et al., *Domain-adversarial training of neural networks*, Domain Adaptation in Computer Vision Applications. Advances in Computer Vision and Pattern Recognition. Springer, Cham., (2017). https://doi.org/10.1007/978-3-319-58347-1_10
- [4] O. Gilo, J. Mathew, S. Mondal, R. K. Sanodiya, *RDAOT: Robust unsupervised deep sub-domain adaptation through optimal transport for image classification*, IEEE Access, **11** (2023), 102243-102260. <https://doi.org/10.1109/ACCESS.2023.3313623>

- [5] M. Golhar, T. L. Bobrow, M. P. Khoshknab, et al., *Improving colonoscopy lesion classification using semi-supervised deep learning*, IEEE Access, **9** (2021), 631-640. <https://doi.org/10.1109/access.2020.3047544>
- [6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., *Generative adversarial nets*, Communications of the ACM, **63**(11) (2020), 139-144. <https://doi.org/10.1145/3422622>
- [7] A. Gretton, K. M. Borgwardt, M. J. Rasch, et al., *A kernel two-sample test*, Journal of Machine Learning Research, **13**(25) (2012), 723-773. <https://doi.org/10.5555/2188385.2188410>
- [8] M. M. Hasan, M. M. Hossain, M. M. Rahman, et al., *FP-CNN: Fuzzy pooling-based convolutional neural network for lung ultrasound image classification with explainable AI*, Computers in Biology, Medicine, **165** (2023), 107407. <https://doi.org/10.1016/j.combiomed.2023.107407>
- [9] K. He, X. Zhang, S. Ren, J. Sun, *Deep residual learning for image recognition*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016), 770-778. <https://doi.org/10.1109/CVPR.2016.90>
- [10] L. Huang, W. Zhao, Y. Liu, et al., *An evidential multi-target domain adaptation method based on weighted fusion for cross-domain pattern classification*, IEEE Transactions on Neural Networks, Learning Systems, **35**(10) (2024), 14218-14232. <https://doi.org/10.1109/TNNLS.2023.3275759>
- [11] T. Kim, S. Lee, *A novel unsupervised clustering and domain adaptation framework for rotating machinery fault diagnosis*, IEEE Transactions on Industrial Informatics, **19**(9) (2023), 9404-9412. <https://doi.org/10.1109/TII.2022.3228395>
- [12] J. Lee, G. Lee, *Unsupervised domain adaptation based on the predictive uncertainty of models*, Neurocomputing, **520** (2023), 183-193. <https://doi.org/10.1016/j.neucom.2022.11.070>
- [13] 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) (2022), 5293-5307. <https://doi.org/10.1109/TNNLS.2021.3069982>
- [14] K. Li, J. Lu, H. Zuo, G. Zhang, *Source-free multidomain adaptation with fuzzy rule-based deep neural networks*, IEEE Transactions on Fuzzy Systems, **31**(12) (2023), 4180-4194. <https://doi.org/10.1109/TFUZZ.2023.3276978>
- [15] M. Long, Y. Cao, Z. Cao, et al., *Transferable representation learning with deep adaptation networks*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **41**(12) (2019), 3071-3085. <https://doi.org/10.1109/TPAMI.2018.2868685>
- [16] M. Long, Y. Cao, J. Wang, et al., *Learning transferable features with deep adaptation networks*, Proceedings of the 32nd International Conference on Machine Learning, **37** (2015), 97-105. <https://doi.org/10.5555/3045118.3045130>
- [17] M. Long, Z. Cao, J. Wang, M. I. Jordan, *Conditional adversarial domain adaptation*, NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems, (2018), 1647-1657.
- [18] M. Long, H. Zhu, J. Wang, M. I. Jordan, *Deep transfer learning with joint adaptation networks*, Proceedings of the 34th International Conference on Machine Learning, **70** (2017), 2208-2217. <https://doi.org/10.5555/3305890.3305909>
- [19] Q. Lou, W. Sun, W. Zhang, et al., *Rules-based heterogeneous feature transfer learning using fuzzy inference*, IEEE Transactions on Fuzzy Systems, **32**(1) (2024), 306-321. <https://doi.org/10.1109/TFUZZ.2023.3298147>
- [20] J. Lu, V. Behbood, P. Hao, et al., *Transfer learning using computational intelligence: A survey*, Knowledge-Based Systems, **80** (2015), 14-23. <https://doi.org/10.1016/j.knosys.2015.01.010>
- [21] Y. Lu, H. Huang, B. Zeng, et al., *Multi-source and multi-target domain adaptation based on dynamic generator with attention*, IEEE Transactions on Multimedia, **26** (2024), 6891-6905. <https://doi.org/10.1109/TMM.2024.3358062>
- [22] J. Lu, H. Zuo, G. Zhang, *Fuzzy multiple-source transfer learning*, IEEE Transactions on Fuzzy Systems, **28**(12) (2020), 3418-3431. <https://doi.org/10.1109/TFUZZ.2019.2952792>

- [23] A. Ma, J. Li, K. Lu, et al., *Adversarial entropy optimization for unsupervised domain adaptation*, IEEE Transactions on Neural Networks and Learning Systems, **33**(11) (2022), 6263-6274. <https://doi.org/10.1109/TNNLS.2021.3073119>
- [24] G. Ma, J. Lu, G. Zhang, *Multisource domain adaptation with interval-valued target data via fuzzy neural networks*, IEEE Transactions on Fuzzy Systems, **32**(5) (2024), 3094-3106. <https://doi.org/10.1109/TFUZZ.2024.3367456>
- [25] C. Qin, L. Wang, Q. Ma, et al., *Semi-supervised domain adaptive structure learning*, IEEE Transactions on Image Processing, **31** (2022), 7179-7190. <https://doi.org/10.1109/TIP.2022.3215889>
- [26] K. Saenko, B. Kulis, M. Fritz, T. Darrell, *Adapting visual category models to new domains*, European Conference on Computer Vision (ECCV), (2010), 213-226. https://doi.org/10.1007/978-3-642-15561-1_16
- [27] E. Tzeng, J. Hoffman, N. Zhang, et al., *Deep domain confusion: Maximizing for domain invariance*, arXiv:1412.3474, (2014). <https://doi.org/10.48550/arXiv.1412.3474>
- [28] H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, *Deep hashing network for unsupervised domain adaptation*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017), 5385-5394. <https://doi.org/10.1109/CVPR.2017.572>
- [29] Z. Wang, L. Chen, F. Y. Wang, *Fuzzy inference attention module for unsupervised domain adaptation*, IEEE Transactions on Fuzzy Systems, **32**(4) (2024), 1706-1718. <https://doi.org/10.1109/TFUZZ.2023.3332751>
- [30] J. Wang, W. Feng, Y. Chen, et al., *Visual domain adaptation with manifold embedded distribution alignment*, Proceedings of the 26th ACM International Conference on Multimedia, (2018), 402-410. <https://doi.org/10.1145/3240508.3240512>
- [31] X. Wang, L. Li, W. Ye, et al., *Transferable attention for domain adaptation*, in Proceedings of the AAAI Conference on Artificial Intelligence, **33**(01) (2019), 5345-5352. <https://doi.org/10.1609/aaai.v33i01.33015345>
- [32] X. Wei, Z. Ji, B. Wen, et al., *Unsupervised multi-target domain adaptation incremental method based on contrastive learning*, IEEE International Conference on Multimedia, (2024), 1-6. <https://doi.org/10.1109/ICME57554.2024.10687405>
- [33] J. Wu, S. Wang, J. Sun, *AMMD: Attentive maximum mean discrepancy for few-shot image classification*, Pattern Recognition, **155** (2024), 110680. <https://doi.org/10.1016/j.patcog.2024.110680>
- [34] H. Xiao, W. Wang, H. Ogai, et al., *Adversarial fuzzy-weighted deep transfer learning for intelligent damage diagnosis of bridge with multiple new damages*, IEEE Sensors Journal, **22**(17) (2022), 17005-17021. <https://doi.org/10.1109/JSEN.2022.3192307>
- [35] P. Xu, Z. Deng, J. Wang, et al., *Transfer representation learning with TSK fuzzy system*, IEEE Transactions on Fuzzy Systems, **29**(3) (2021), 649-663. <https://doi.org/10.1109/TFUZZ.2019.2958299>
- [36] Y. Ye, T. Pan, Q. Meng, et al., *Online unsupervised domain adaptation via reducing inter- and intra-domain discrepancies*, IEEE Transactions on Neural Networks and Learning Systems, **35**(1) (2024), 884-898. <https://doi.org/10.1109/TNNLS.2022.3177769>
- [37] H. Yu, M. Hu, S. Chen, *Multi-target unsupervised domain adaptation without exactly shared categories*, arXiv:1809.00852, (2018). <https://doi.org/10.48550/arXiv.1809.00852>
- [38] F. Zhang, F. Chen, Z. Su, J. Wei, *Unsupervised domain adaptation via dynamic clustering and co-segment attentive learning for video-based person re-identification*, IEEE Access, **12** (2024), 29583-29595. <https://doi.org/10.1109/ACCESS.2024.3365583>
- [39] Y. Zhu, F. Zhuang, J. Wang, et al., *Deep subdomain adaptation network for image classification*, IEEE Transactions on Neural Networks and Learning Systems, **32**(4) (2021), 1713-1722. <https://doi.org/10.1109/TNNLS.2020.2988928>