

Single-layer feedforward neural networks with dynamic width for domain adaptation

Le YANG¹, Zelin YANG¹, Fan LI^{1*} & C.L. Philip CHEN²

¹*School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

²*School of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China*

Received 24 April 2024/Revised 19 September 2024/Accepted 9 January 2025/Published online 25 March 2025

Abstract Broad learning system (BLS) is a recently proposed single-layer feedforward network (SLFN) with strong generalization ability in different industrial applications. However, the classical BLS assumes that the training and testing data are drawn from the same distribution, which can be often violated in the real world. This paper proposes a novel dynamic domain adaptation (DA) framework based on BLS (DDA-BLS). Compared with most existing DA methods, which follow a static feature learning protocol, the proposed DDA-BLS applies a data-dependent dynamic feature learning procedure for different target inputs. Although such a dynamic feature learning procedure seems to be a more intelligent DA strategy and improves DA performance, it has rarely been explored in the DA fields. Comprehensive experiments on several DA tasks, including image classification and fault diagnosis, demonstrate the effectiveness and efficiency of the proposed DDA-BLS in DA tasks, further indicating the superiority of the dynamic DA strategy.

Keywords domain adaptation, broad learning system (BLS), image classification, fault diagnosis, neural networks

Citation Yang L, Yang Z L, Li F, et al. Single-layer feedforward neural networks with dynamic width for domain adaptation. *Sci China Inf Sci*, 2025, 68(8): 182205, <https://doi.org/10.1007/s11432-024-4273-8>

1 Introduction

Recently, neural networks have been widely studied and applied in various industrial fields [1–3]. Besides the modern deep learning architectures [4], researchers also have developed different single-layer feedforward neural networks (SLFN) to efficiently address classification and regression problems [5], such as broad learning system (BLS) [6]. As an SLFN, the first-layer weights of BLS are randomly initialized without tuning, only the weights connecting the hidden and output layers need to be learned through the pseudo-inverse algorithm, guaranteeing the fast training speed of BLS. Recent studies have also designed its variants to guarantee the model robustness when facing noisy outliers [7] or limited training samples [8]. Although with a single hidden layer, BLS can achieve comparable or even better performance than some deep models in many real-world applications, including network intrusion detection [9], fault diagnosis [10], and so on [11–13].

Compared with other SLFNs, the BLS can be quickly expanded by adding additional nodes into the original model with the incremental learning process [6], which makes it flexible to handle tasks with different complexities without retraining. However, the performance of most BLS methods is based on the assumption that training and testing data are sampled from the same distribution, which can often be violated during applications [1, 2]. Therefore, it is necessary to mitigate the knowledge gap between the abundant labeled training data in the source domains and the unlabeled testing data in the target domain, which refers to domain adaptation (DA) problems [2].

Domain adaptation aims at learning an accurate classifier for the target domain using sufficient source domain labeled samples and unlabeled samples from the target domain, which leads to the major issue in domain adaptation: how to explore feature representations that can effectively mitigate the distribution gap between the two domains [14]. A direct solution is to explore domain invariant features by matching the marginal distributions between source and target domains via minimizing certain measurement metrics (such as empirical maximum mean discrepancy (MMD) [15]) that represent the distances between

* Corresponding author (email: lifan@mail.xjtu.edu.cn)

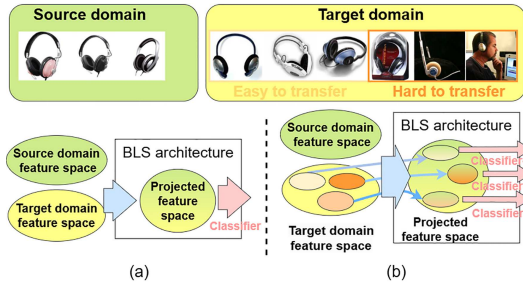


Figure 1 (Color online) Difference between static and dynamic DA processes. (a) The static DA process will transform all target samples in the same way. (b) The dynamic DA process will use different feature transformations based on the characteristics of the target samples.

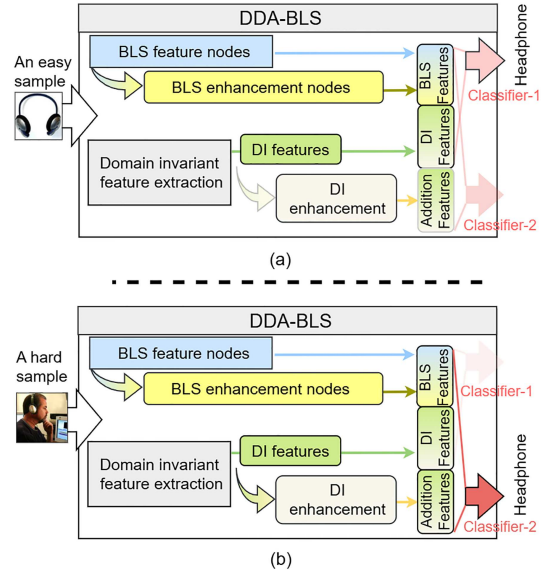


Figure 2 (Color online) Dynamic DA procedure of the proposed DDA-BLS for the samples with different complexities in the target domain. (a) An easy sample; (b) a hard sample.

source and target domains [16–20]. With the development of deep learning, deep DA methods have also been developed to mitigate the domain discrepancy. Domain adaptation network (DAN) [21] uses an MMD-based domain confusion loss and embeds all deep layer features in a reproducing kernel Hilbert space where the source and target means are matched. A series of deep DA methods also propose and show promising performance in visual tasks [22–24] and fault diagnosis tasks [25–31]. Although deep model based DA methods can achieve promising performance in many tasks, these methods usually need high computational costs for both training and inference, which emphasize the necessity of developing SLFNs-based DA methods with high performance and efficiency.

To mitigate the knowledge gap between source and target domains, BLS-based DA methods have been proposed to extract the domain-invariant (DI) features and have been applied in electroencephalography signal classification [32], fault diagnosis [33], textual emotion classification [34], and the drift compensation problems [35]. However, most BLS-based DA methods conduct the DI feature extraction with a static procedure, ignoring the characteristics of each input (shown in Figure 1(a)). For example, all the target samples will be transformed and projected into the same subspace and then classified by a learned classifier. However, considering the differences in target inputs, such a static procedure may not be optimal. A more reasonable way might be to explore the DI features in a fine-grained dynamic way (shown in Figure 1(b)). The “easy-to-transfer” and “hard-to-transfer” target samples might need to be processed by different feature extraction procedures during DA.

In this paper, we propose a novel dynamic domain adaptation framework based on BLS (DDA-BLS) to achieve such a data-dependent feature extraction for DA. From Figure 2, we see that the proposed DDA-BLS framework contains two main components: a basic BLS feature extraction and a DI feature extraction, which are used to build a multi-classifier SLFN architecture to conduct the dynamic domain adaptation. During recognizing the target samples, the trained DDA-BLS can adapt its architecture based on the characteristics of the input. “Easy” target samples showing strong similarity with the source ones will be mainly classified by the first classifier (Classifier-1 in Figure 2) based on basic BLS and DI representations; only for these “hard” target inputs, the additional DI enhancement and Classifier-2 will be activated for target sample recognition. Such a data-dependent dynamic inference procedure enables a fine-grained DA feature learning process for each target input, leading to higher performance in DA tasks.

The proposed DDA-BLS is essentially different from the classic BLS [6] from two aspects. First, the proposed DDA-BLS can effectively mitigate the knowledge gap between source and target domains. More importantly, the architecture of the DDA-BLS can adapt based on different inputs, while the BLS uses

the static network once the training finishes. Compared with most existing DA methods, the dynamic DA procedure enables different knowledge-transferring processes for different target samples, leading to higher final performance. Such a dynamic feature learning strategy has rarely been explored in the existing DA methods. The contributions can be summarized as follows.

- We proposed a novel BLS-based dynamic domain adaptation method, named DDA-BLS, which can effectively and efficiently mitigate the domain gap in domain adaptation tasks.
- We show the effectiveness of the dynamic DA procedure, which enables the DDA-BLS to select suitable hidden representations for each target input, resulting in a more intelligent feature transformation procedure. Such a data-dependent DA process has rarely been explored in DA fields.
- Our method can be implemented in either semi-supervised or unsupervised DA settings. Experiments on visual classification tasks (Office+Caltech-256 [16]) and fault diagnosis tasks (Case Western Reserve University bearing data, CWRU [36], Intelligent Maintenance Systems bearing data, IMS [37]) further demonstrate the effectiveness and efficiency of the proposed DDA-BLS.

2 Preliminaries

To make the paper self-contained, this section briefly introduces BLS [6], which is illustrated in Figure 3(a). Let $\{\mathbf{X}, \mathbf{Y}\} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ be a dataset with N samples, where $\mathbf{X} \in \mathbb{R}^{N \times D}$, $\mathbf{Y} \in \mathbb{R}^{N \times C}$, $\mathbf{x}_i \in \mathbb{R}^{1 \times D}$, and $\mathbf{y}_i \in \mathbb{R}^{1 \times C}$ are the inputs and labels (one-hot), respectively. C is the number of classes and D is the original dimension of inputs.

2.1 Basic BLS

The BLS firstly applies random weights to project the feature from the original space to a random high-dimension space, which is realized by mapping the inputs \mathbf{X} to n feature mappings. The i th mapped features \mathbf{Z}_i can be represented by

$$\mathbf{Z}_i = \phi(\mathbf{X}\mathbf{W}_{z_i} + \mathbf{1}_N \cdot \mathbf{b}_{z_i}), \quad i = 1, 2, \dots, n, \quad (1)$$

where $\mathbf{W}_{z_i} \in \mathbb{R}^{D \times p}$ and $\mathbf{b}_{z_i} \in \mathbb{R}^{1 \times p}$ are randomly generated with the proper dimensions, $\mathbf{1}_N$ is the all-one vectors with N elements, and $\phi(\cdot)$ is the nonlinear activation function. Then the n groups of feature nodes can be collected to obtain $\mathbf{Z}^n \equiv [\mathbf{Z}_1, \dots, \mathbf{Z}_n]$, which is utilized to generate m groups of enhancement nodes. The j th enhancement nodes \mathbf{H}_j can be represented by

$$\mathbf{H}_j = \xi(\mathbf{Z}^n \mathbf{W}_{h_j} + \mathbf{1}_{(np)} \cdot \mathbf{b}_{h_j}), \quad j = 1, 2, \dots, m, \quad (2)$$

where $\mathbf{W}_{h_j} \in \mathbb{R}^{(np) \times q}$ and $\mathbf{b}_{h_j} \in \mathbb{R}^{1 \times q}$ are randomly generated and $\xi(\cdot)$ is the nonlinear activation function. The collection of enhancement features can be defined as $\mathbf{H}^m \equiv [\mathbf{H}_1, \dots, \mathbf{H}_m]$.

In practice, the number of n and m can be selected depending on the complexity of the modeling tasks. After feature generation, the BLS features can be represented as

$$\mathbf{A}_n^m = [\mathbf{Z}^n \mid \mathbf{H}^m] \in \mathbb{R}^{N \times (np+mq)}. \quad (3)$$

BLS aims to learn the best $\boldsymbol{\beta} \in \mathbb{R}^{(np+mq) \times C}$ connecting the outputs and hidden layers to represent the label matrix \mathbf{Y} , which minimizes the sum of the squared losses of the prediction errors, which leads to the following problem:

$$\min_{\boldsymbol{\beta}} J_{\text{BLS}} = \frac{\lambda}{2} \|\boldsymbol{\beta}\|^2 + \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\boldsymbol{\beta}\|^2, \quad (4)$$

where the first term in (4) containing λ is the regularization term to prevent the over-fitting. By setting the gradient of J_{BLS} w.r.t $\boldsymbol{\beta}$ to 0, the optimal solution can be calculated efficiently by

$$\boldsymbol{\beta}_{\text{BLS}}^* = (\lambda \mathbf{I} + \mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{Y}, \quad (5)$$

where \mathbf{I} is the identical matrix with proper dimension.

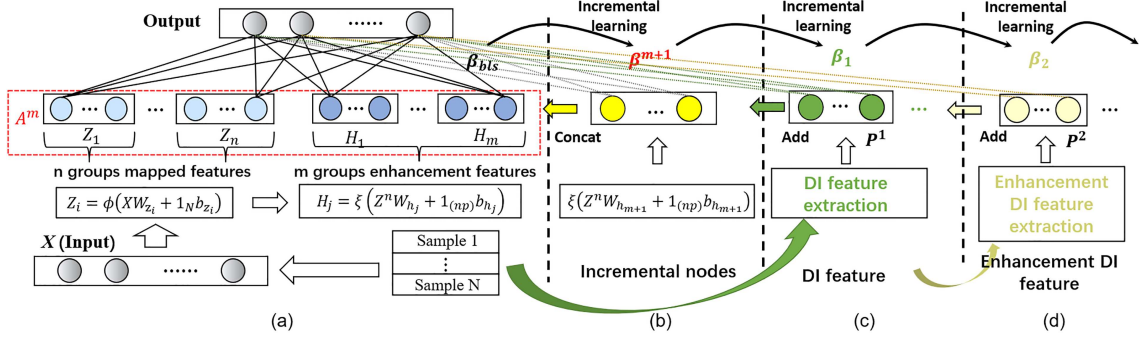


Figure 3 (Color online) Illustrations of BLS and DDA-BLS. The classical BLS only contains (a) the basic BLS and (b) the incremental learning. DDA-BLS further contains (c) the DI feature injection and (d) the DI feature enhancement.

2.2 Incremental learning

If the learning cannot reach the desired accuracy, BLS can be incrementally updated without retraining the whole model in three incremental learning manners [6]: increment of enhancement nodes, feature nodes, and inputs. Here we mainly introduce the increment of enhancement nodes. With a trained BLS model¹⁾ with A^m and β^m as the BLS features and the optimal solution, then we expand the BLS by adding l enhancement nodes. The expanded hidden features can be represented by

$$A^{m+1} \equiv [A^m | \xi(Z^n W_{h_{m+1}} + \mathbf{1}_{(np)} \cdot b_{h_{m+1}})], \quad (6)$$

where $W_{h_{m+1}} \in \mathbb{R}^{(np) \times l}$ and $b_{h_{m+1}} \in \mathbb{R}^{1 \times l}$ are the random weights, and the pseudo-inverse of A^{m+1} can be updated by

$$(A^{m+1})^+ = \begin{bmatrix} (A^m)^+ - DB^T \\ B^T \end{bmatrix}, \quad (7)$$

where $D = (A^m)^+ \xi(Z^n W_{h_{m+1}} + \mathbf{1}_{(np)} \cdot b_{h_{m+1}})$,

$$B^T = \begin{cases} (E)^+, & \text{if } E \neq 0, \\ (\mathbf{1} + D^T D)^{-1} B^T (A^m)^+, & \text{if } E = 0, \end{cases} \quad (8)$$

and $E = \xi(Z^n W_{h_{m+1}} + \mathbf{1}_{(np)} \cdot b_{h_{m+1}}) - A^m D$.

Then, the new weights can be incrementally updated by

$$\beta^{m+1} = \begin{bmatrix} \beta^m - DB^T Y \\ B^T Y \end{bmatrix}. \quad (9)$$

The detailed BLS model and learning procedure can be found in [6]. Because the optimization of the BLS only involves pseudo-inverse calculation, the model can be trained and updated with high efficiency.

3 BLS-based dynamic domain adaptation

3.1 Overall architecture of DDA-BLS

An overall illustration of the proposed DDA-BLS is shown in Figure 3. It contains a basic BLS (Figure 3(a)) and a DI feature processing sub-network (Figures 3(c) and (d)). Generally, we can view the proposed DDA-BLS as an SLFN with multiple classifiers. Based on the calculated hidden states, the additional hidden representations will be successively generated and added to the DDA-BLS. Simultaneously, a series of classifiers which connect different hidden representations will be learned.

As shown in Figures 3(a) and (b), the DDA-BLS applies the basic BLS [6]. Given the input data X , the hidden representation A will be calculated using (3). Then the basic classifier β will be learned.

1) Let A^m denote A_n^m for simplicity, and $(A^m)^+ = \lim_{\lambda \rightarrow 0} (\lambda I + A^T A)^{-1} A^T$.

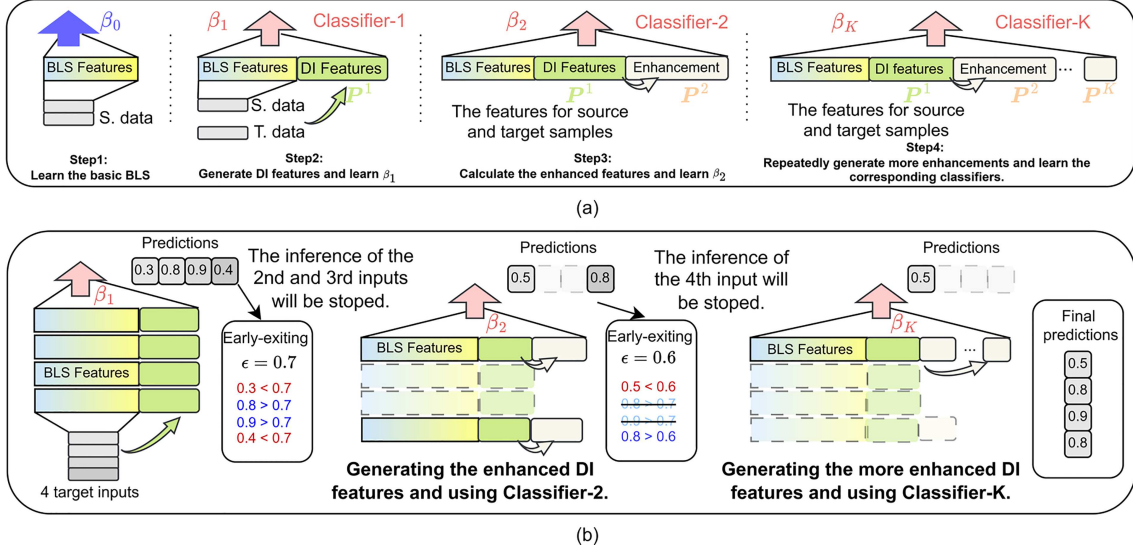


Figure 4 (Color online) Illustrations of (a) training and (b) dynamic inference procedure of DDA-BLS with K classifiers.

Moreover, the enhancement nodes will be generated using (6), and the BLS classifier β_0 can be updated by (9).

The DI feature processing sub-network is designed to extract the DI features for the DDA-BLS. Due to the incremental learning mechanism of BLS, these extracted DI features \mathbf{P} can be directly injected into the DDA-BLS, where the new classifiers can be updated by (9). Moreover, the enhancement nodes can be also generated based on the DI features \mathbf{P} , and the additional classifiers can also be updated incrementally, which is further illustrated in Figures 3(c) and (d).

3.2 Training and inference procedure of DDA-BLS

3.2.1 Training of DDA-BLS

The training procedure of DDA-BLS is shown in Figure 4(a). Given the source samples $\mathcal{D}_S = \{(\mathbf{x}_{s_i}, y_{s_i})\}_{i=1}^{n_s} = \{\mathbf{X}_s, \mathbf{y}_s\}$ and target samples $\mathcal{D}_T = \{\mathbf{x}_{t_i}\}_{i=1}^{n_t} = \{\mathbf{X}_t\}$, the basic BLS in the DDA-BLS will be first trained using \mathcal{D}_S by (5) and (9) to obtain the basic β_0 , which can be used for generating the pseudo labels for target samples if necessary. Then, a feature extraction-based domain adaptation algorithm will be applied to extract the domain invariant features²⁾, which can be defined as \mathbf{P} . We can use the basic BLS to generate the pseudo labels for target samples if necessary during exploring DI features \mathbf{P} . Then, these \mathbf{P} will be injected as the additional features, and the weight of Classifier-1, β_1 , can be learned incrementally as (9). Subsequently, the enhancement DI features will be generated using the same way of (6), and the weight of the corresponding classifier β_2 can be learned again by (9). Such a procedure is repeated until we have K classifiers.

3.2.2 Dynamic inference

After training, we will have k different classifiers in DDA-BLS. The dynamic inference procedure is shown in Figure 4(b). Given the target inputs \mathbf{X}_t , the DDA-BLS will first generate the BLS features \mathbf{A} and DI features \mathbf{P}^1 and then classify the inputs with β_1 . After selecting and removing the samples that have already achieved high confidence (higher than a give exiting threshold, ϵ_1) results at Classifier-1, the DDA-BLS will generate the additional incremental DI features for the rest of the target samples and then classify them with Classifier-2. Such a procedure is repeated until the prediction confidence is higher than the threshold or the last Classifier- k is applied. Formally, the prediction results $\hat{\mathbf{Y}}^k$ of the k th classifier can be represented as

$$\hat{\mathbf{Y}}^k = f_{\beta_k}([\mathbf{A}, \mathbf{P}^1, \dots, \mathbf{P}^k]) = [\hat{y}_1^k; \dots; \hat{y}_{n_t}^k] \in \mathbb{R}^{n_t \times C}, \quad (10)$$

2) Here, we mainly focus on the training procedure of the proposed DDA-BLS and omit the learning steps of the DA methods.

Algorithm 1 Dynamic DA inference of DDA-BLS.

Require: $\mathcal{D}_T = \{\mathbf{x}_{t_i}\}_{i=1}^{n_t} = \{\mathbf{X}_t\}$, the trained DDA-BLS, the thresholds $\epsilon_1, \dots, \epsilon_{K-1}$;

Ensure:

- 1: Set the input dataset \mathcal{S} containing all target samples;
 - 2: Generate the BLS features for \mathbf{X}_t ;
 - 3: **for** $k = 1, \dots, K$ **do**
 - 4: Generate or enhance the DI features as \mathbf{P}^k for the samples in \mathcal{S} ;
 - 5: Use β_k to calculate $\hat{\mathbf{Y}}^k$ for the samples in \mathcal{S} ;
 - 6: **if** $k = K$ **then**
 - 7: Use β_K to calculate $\hat{\mathbf{Y}}^K$ for the rest samples in \mathcal{S} ;
 - 8: **else**
 - 9: **for** $i = 1, \dots, n_t$ **do**
 - 10: **if** $\max(\hat{\mathbf{y}}_i^k) > \epsilon_k$ **then**
 - 11: Return $\hat{\mathbf{y}}_i^k$ as the final prediction for the i th sample;
 - 12: Remove the i th sample from \mathcal{S} ;
 - 13: **end if**
 - 14: **end for**
 - 15: **end if**
 - 16: **end for**
 - 17: Collect all results and get the final prediction $\hat{\mathbf{Y}}_t$.
-

where $\hat{\mathbf{Y}}^k$ is the output matrix at the k th classifier and $\hat{\mathbf{y}}_i^k$ is the output row vector for the i th target sample, where the element, $\hat{y}_{i,c}^k \in [0, 1]$, represents the prediction confidence of the i th target sample as the c th class index. Normally, a target sample, \mathbf{x}_i , will be recognized as the class with the largest prediction value in each $\hat{\mathbf{y}}_i^k \in \mathbb{R}^{1 \times C}$.

During inference, the largest prediction value of $\hat{\mathbf{y}}_i^k$ will be used to estimate the reliability of the prediction for this sample. The final prediction $\hat{\mathbf{y}}_i$ of \mathbf{x}_i will be the prediction of the first classifier whose largest prediction value is greater than the given threshold, ϵ_k , which can be represented by

$$k^* = \min \{k \mid \max(\hat{\mathbf{y}}_i^k) > \epsilon_k\}, \quad \hat{\mathbf{y}}_i = \hat{\mathbf{y}}_i^{k^*}. \quad (11)$$

To illustrate such a dynamic inference procedure, we provide an example of a trained DDA-BLS with K classifiers in Figure 4(b). Given four different target inputs, the proposed DDA-BLS will first generate the BLS features and DI features, and these inputs will be classified using β_1 . Then, we can obtain the prediction confidence of these samples. The 2nd and 3rd samples will exit at Classifier-1 because they have achieved confidences higher than the given exiting threshold ($\epsilon_1 = 0.7$), meaning that the DDA-BLS has already recognized these target samples. Then, the features of these samples will be removed from the whole feature matrix. While, for the 1st and 4th target samples in Figure 4(b), their outputs at Classifier-1 achieve values lower than the threshold ϵ_1 . Therefore, the DI features of these samples will be enhanced and expanded, and the subsequent classifiers will be used for inference. This procedure will be repeated until the prediction result is higher than the threshold ϵ_k at Classifier- k or the last classifier, Classifier- K , is used. After the whole procedure, the classification results for the target samples can be obtained by collecting the existing results of each classifier. We also summarize the dynamic inference procedure of DDA-BLS in Algorithm 1.

3.3 Implementation of DDA-BLS

The proposed DDA-BLS is a general domain adaptation framework and can be combined with different subspace learning-based DA algorithms. Here, we provide an implementation example using the DA method extracting domain invariant and class discriminative (DICD) features [38].

As shown in Figure 5(a), the basic BLS part in DDA-BLS will be first trained by the labeled source samples, and generate the pseudo labels for target samples. By simultaneously matching the MMD [15] between source and target domains (marginal distribution) and the MMDs of each class between two domains (conditional distributions), the DI features will be extracted from the original data (see details in [38]). Then with the extracted DI features, we can train the DDA-BLS by the aforementioned procedure and generate new pseudo labels for target samples with dynamic DA inference. Such a procedure can be repeated until the performance gets stable. Unless stated otherwise, the DDA-BLS is implemented with DICD in the following experiments.

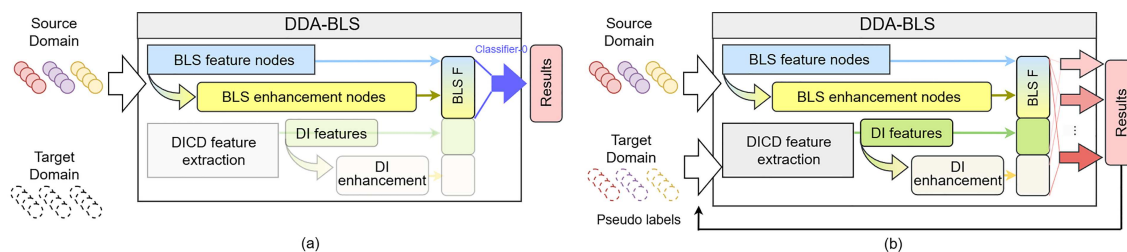


Figure 5 (Color online) Implementation of DDA-BLS with DICD [38]. (a) Step-1: pseudo label generation; (b) step-2: DI feature extraction and DDA-BLS learning.

3.4 Discussion

Although the proposed DDA-BLS is developed based on BLS, and both use the incremental learning strategy, our method is essentially different from the classical BLS. First, the performance of the classic BLS can only be guaranteed when the training and test samples are drawn from the same distribution. While the proposed DDA-BLS can effectively mitigate the domain gap and achieve high performance, the model is evaluated even on the domain-shifted test dataset. Second, most of the existing BLS models follow the static inference procedure, which ignores the characteristics of the inputs. For example, the recently proposed adaptive BLS [33] can effectively update the BLS architecture incrementally, while the whole model will be fixed for different inputs during inference. Recent studies [39–44] have found that the multi-classifier-based dynamic inference procedure, which adapts the network architecture during inference, can improve the overall efficiency of deep learning models in image classification tasks. This paper proves that this dynamic inference procedure can be even more effectively implemented in BLS due to its incremental learning strategy.

We further discuss the essential difference between the proposed DDA-BLS and the existing DA methods. As most existing DA methods apply the same feature learning or feature transformation procedure for all target inputs ignoring the characteristics of the samples, they can all be viewed as the static method. Intuitively, different target samples may have different optimal ways to mitigate the domain variance between the source domain samples, which might lead to sub-optimal DA performance. The proposed DDA-BLS can deal with different target inputs considering their characteristics and therefore is a dynamic DA method. We also show the strong potential of the dynamic DA feature learning processing in improving DA performance through extensive experiments on different DA tasks. The visualization results further show that such a dynamic DA process meets our intuition. We found these target samples exiting from the first classifier in DDA-BLS will be strongly similar to the samples from the source domain. Therefore, the trained DDA-BLS will consider these target samples easy ones. The samples exiting from the final classifier normally contain more complex semantic information or the target objects show a non-typical appearance, and therefore, the enhancement features are necessary for accurate recognition. The trained DDA-BLS will consider these target samples hard ones. It is worth mentioning that the complexity of target samples is defined by the DDA-BLS itself rather than pre-defined. Interestingly, the definition of the complexity by the trained DA model meets our intuition, indicating that the dynamic DA procedure is indeed more intelligent compared with the static ones.

4 Experiments

Both image classification and fault diagnosis tasks are used in the experiments. All experiments are conducted on Win 10, MATLAB R2021a, Intel-i5 2.4 GHz CPU, and 8G DDR3 RAM with 3 times to obtain the average performance.

4.1 Case 1: DA on image classification tasks

4.1.1 Experimental settings

The experiments are conducted on Office+Caltech-256 [45], which contains the images with 10 shared categories from four different domains: AMAZON (A), CALTECH (C), DSLR (D), and WEBCAM (W), leading to 12 cross-domain tasks, e.g., “A → C”, “A → D”, . . . , “W → C” and “W → D”.

Table 1 Average classification accuracy \pm standard deviation on Office+Caltech-256 from source to target domain. The best results are in bold.

Task	BLS	RTML	DTSL	ARTL	TJM	TKL	DICD	DDA-BLS
A \rightarrow C	80.32 \pm 0.54	86.43 \pm 0.00	85.75 \pm 0.00	84.42 \pm 0.00	80.79 \pm 0.05	80.32 \pm 0.00	86.02 \pm 0.00	87.80\pm0.09
A \rightarrow D	80.25 \pm 0.64	84.36 \pm 0.00	82.17 \pm 0.00	87.26 \pm 0.00	86.62 \pm 0.00	82.80 \pm 0.00	83.44 \pm 0.00	91.72\pm0.36
A \rightarrow W	72.31 \pm 0.39	80.26 \pm 0.00	73.56 \pm 0.00	80.00 \pm 0.00	79.66 \pm 0.00	81.35 \pm 0.00	81.36 \pm 0.00	89.15\pm0.19
C \rightarrow A	89.63 \pm 0.48	90.62 \pm 0.00	91.54 \pm 0.00	91.86 \pm 0.00	89.66 \pm 0.00	89.14 \pm 0.00	91.02 \pm 0.00	93.53\pm0.11
C \rightarrow D	80.47 \pm 0.97	89.32 \pm 0.00	87.90 \pm 0.00	85.99 \pm 0.00	87.89 \pm 0.00	79.62 \pm 0.00	93.63 \pm 0.00	95.54\pm0.68
C \rightarrow W	75.03 \pm 0.39	85.38 \pm 0.00	76.61 \pm 0.00	77.97 \pm 0.00	80.67 \pm 0.00	76.61 \pm 0.00	92.20 \pm 0.00	93.56\pm0.58
D \rightarrow A	81.04 \pm 0.16	91.86 \pm 0.00	84.97 \pm 0.00	83.09 \pm 0.00	89.66 \pm 0.00	81.32 \pm 0.00	92.17 \pm 0.00	93.00\pm0.06
D \rightarrow C	75.33 \pm 0.71	85.72 \pm 0.00	75.24 \pm 0.00	78.27 \pm 0.00	79.16 \pm 0.00	68.92 \pm 0.00	86.11 \pm 0.00	88.33\pm0.13
D \rightarrow W	97.06 \pm 0.71	98.98 \pm 0.00	99.32 \pm 0.00	96.95 \pm 0.00	98.30 \pm 0.00	87.46 \pm 0.00	98.98 \pm 0.00	99.66\pm0.20
W \rightarrow A	47.59 \pm 1.01	91.37 \pm 0.00	75.47 \pm 0.00	80.79 \pm 0.00	86.74 \pm 0.00	73.38 \pm 0.00	89.67 \pm 0.00	93.22\pm0.24
W \rightarrow C	46.96 \pm 0.72	83.13 \pm 0.00	72.75 \pm 0.00	81.74 \pm 0.00	78.62 \pm 0.00	72.48 \pm 0.00	83.97 \pm 0.00	87.53\pm0.10
W \rightarrow D	78.98 \pm 0.64	100.00\pm0.00	100.00\pm0.00	98.73 \pm 0.00	98.72 \pm 0.00	93.63 \pm 0.00	100.00\pm0.00	100.00\pm0.00
AVERAGE	75.41	88.95	83.77	85.59	86.37	80.59	89.88	92.75

In unsupervised DA setting, seven machine learning methods are used during comparison, including the basic BLS [6], robust transfer metric learning (RTML) [17], discriminative transfer subspace learning (DTSL) [18], adaptation regularization for transfer learning (ARTL) [20], transfer joint matching (TJM) [46], transfer kernel learning (TKL) [19], and DICD feature learning [38]. Our experiments follow most of the settings provided in [38]. The optimal final dimension of the extracted subspace is searched from the range $d = \{10, 20, \dots, 100\}$. Moreover, following [38], the hyper-parameters of the DDA-BLS are selected using the hyper-parameters with the best performance on the target domain. The number of classifiers of DDA-BLS is $K = 5$, with corresponding $\epsilon = (0.8, 0.5, 0.4, 0.6)$. Note that the rest samples will be forced to exit from the last classifier. It does not need a threshold during inference.

In the semi-supervised experiment setting, we randomly select 10% of the target samples, label them, and then use the rest of the target data to test the performance. We use the basic BLS, ARTL, TKL, and the BLS-based source domain adaptation (BLS-SDA) [12], which is designed for semi-supervised DA.

4.1.2 Experimental results

The classification accuracy of DA results on the Office+Caltech-256 features with unsupervised and semi-supervised settings is listed in Tables 1 and 2, respectively. The results show that the proposed methods can outperform other baseline methods on all cross-domain tasks. The performance of the basic BLS is significantly lower than that of other DA-based algorithms, meaning that domain adaptation is essential in tasks when there are knowledge gaps between source and target domains. Moreover, the high performance of DDA-BLS compared with DICD indicates that the dynamic DA process indeed benefits the domain adaptation. Compared with the results of DDA-BLS with an unsupervised learning setting, the DDA-BLS with a small portion of labeled target samples can improve its DA performance, which meets our intuition. The superiority of the DDA-BLS in both settings demonstrates the effectiveness of our method.

4.2 Case 2: DA for bearing fault diagnosis under different working conditions

4.2.1 Experimental settings

This experiment uses CWRU [36], which contains the signals measuring the rolling bearing under two sampling frequencies (48 kHz and 12 kHz) with four loads and speeds: HP0 (horse power 0) (1797 r \cdot min $^{-1}$), HP1 (1772 r \cdot min $^{-1}$), HP2 (1750 r \cdot min $^{-1}$), and HP3 (1730 r \cdot min $^{-1}$), leading to 12 cross-domain transfer learning tasks, e.g., “HP0 \rightarrow HP1”, \dots , “HP3 \rightarrow HP2”. The faults are realized by machining pits of different depths on the inner race (IR), outer race (OR), and ball elements (BALL), with depths of 0.18, 0.36, and 0.53, respectively. Two different experiment settings are applied in this paper. The first one follows the setting in [1], where the data with 12 kHz is used and one location of the pit is considered one category, resulting in four different classes. In the second setting, the 48 kHz data is divided into one normal state and nine fault states (a total of 10 classes), all containing roughly equal amounts of data (approximately 400 samples for each class). Following [11], the extracted wavelet energy features are used as data pre-processing.

Table 2 Semi-supervised performance evaluation on Office+Caltech-256 (left panel) and CWRU with 48 kHz data (right panel). The best results are in bold.

Task	ARTL	TKL	BLS-SDA	DDA-BLS	Tasks	ARTL	TKL	BLS-SDA	DDA-BLS
A → C	86.35±0.00	82.39±0.00	84.54±0.59	89.42±0.36	HP0 → HP1	68.11±0.00	72.00±0.00	64.67±0.18	76.22±0.18
A → D	88.65±0.00	82.97±0.00	92.20±0.00	94.80±0.41	HP0 → HP2	67.89±0.00	73.42±0.00	65.84±0.39	76.79±0.39
A → W	82.64±0.00	83.77±0.00	80.75±0.00	92.33±0.21	HP0 → HP3	65.69±0.00	75.69±0.00	67.55±0.17	78.92±0.17
C → A	92.34±0.00	90.37±0.00	91.22±0.37	93.97±0.20	HP1 → HP0	65.83±0.00	75.03±0.00	70.94±0.03	88.65±0.03
C → D	87.94±0.00	80.85±0.00	86.52±0.71	95.51±0.81	HP1 → HP2	86.83±0.00	87.06±0.00	84.28±0.20	89.27±0.20
C → W	84.91±0.00	77.35±0.00	85.16±0.58	96.10±0.43	HP1 → HP3	80.75±0.00	82.08±0.00	79.60±0.17	85.16±0.17
D → A	91.76±0.00	88.97±0.00	90.52±0.13	93.70±0.13	HP2 → HP0	66.33±0.00	78.61±0.00	73.52±0.44	89.23±0.44
D → C	88.03±0.00	81.40±0.00	87.57±0.15	90.57±0.05	HP2 → HP1	84.64±0.00	85.92±0.00	84.12±0.34	89.48±0.34
D → W	98.11±0.00	87.92±0.00	98.49±0.38	99.37±0.43	HP2 → HP3	82.25±0.00	83.64±0.00	81.42±0.49	85.62±0.49
W → A	91.53±0.00	88.40±0.00	87.70±0.12	93.93±0.07	HP3 → HP0	63.89±0.00	82.00±0.00	78.83±0.81	89.69±0.81
W → C	83.09±0.00	80.51±0.00	83.06±0.06	88.92±0.10	HP3 → HP1	76.19±0.00	79.69±0.00	77.59±0.39	85.10±0.39
W → D	98.58±0.00	92.90±0.00	99.29±0.00	100.00±0.00	HP3 → HP2	80.16±0.00	82.25±0.00	81.60±0.36	87.23±0.36
AVERAGE	89.49	84.82	88.92	94.05	AVERAGE	74.05	79.78	75.83	85.11

Table 3 Average fault diagnosis ± standard deviation on the CWRU dataset (12 kHz) from source to target. The best results are in bold.

Task	BLS	ARTL	TJM	TKL	DICD	DTLCNN	DCSAN	DJDAN	DDA-BLS
HP0 → HP1	91.17±0.35	95.33±0.00	96.50±0.00	94.75±0.00	94.42±0.00	98.42±0.12	99.07±0.23	98.04±0.35	99.92±0.00
HP0 → HP2	87.75±0.35	91.75±0.00	96.91±0.00	86.33±0.00	93.67±0.00	97.67±0.71	98.14±0.34	98.79±0.76	99.84±0.12
HP0 → HP3	84.09±0.47	91.66±0.00	95.16±0.00	94.25±0.00	93.25±0.00	96.88±0.90	98.86±0.92	98.13±0.06	99.33±0.11
HP1 → HP0	92.71±0.53	95.25±0.00	97.58±0.00	96.33±0.00	96.67±0.00	99.09±0.12	98.21±0.65	97.66±0.83	99.83±0.12
HP1 → HP2	93.13±0.18	93.08±0.00	93.16±0.00	97.83±0.00	95.08±0.00	99.25±0.11	99.17±0.45	99.13±0.29	99.92±0.17
HP1 → HP3	90.88±0.42	92.33±0.00	95.75±0.00	97.58±0.00	90.58±0.00	98.38±0.53	99.33±0.89	98.67±0.35	99.67±0.00
HP2 → HP0	87.13±0.77	93.58±0.00	95.50±0.00	88.66±0.00	96.67±0.00	97.46±0.23	97.50±0.34	97.59±0.12	100.00±0.00
HP2 → HP1	91.50±0.59	94.50±0.00	97.50±0.00	93.83±0.00	96.75±0.00	99.00±0.47	98.93±0.56	98.46±0.88	99.86±0.26
HP2 → HP3	91.29±0.41	94.58±0.00	96.83±0.00	99.16±0.00	96.08±0.00	97.42±0.58	98.48±0.12	98.63±0.17	99.50±0.11
HP3 → HP0	86.13±0.06	91.50±0.00	95.00±0.00	90.00±0.00	91.50±0.00	96.42±0.23	97.79±0.73	95.98±0.39	99.58±0.34
HP3 → HP1	88.87±0.29	91.58±0.00	94.41±0.00	90.75±0.00	93.25±0.00	97.71±0.65	97.26±0.21	98.17±0.83	98.78±0.06
HP3 → HP2	91.84±0.47	92.33±0.00	96.08±0.00	95.67±0.00	94.50±0.00	97.63±0.42	98.43±0.96	97.79±0.65	100.00±0.00
AVERAGE	89.71	93.12	95.87	93.76	94.37	97.94	98.43	98.09	99.64

We compare the proposed methods with 4 classical DA methods, including ARTL [20], TJM [46], TKL [19], DICD [38], and 3 deep learning-based DA models, including dynamic joint distribution alignment network (DJDAN) [47], deep transfer learning convolutional neural network (DTL CNN) [1], and deep convolution and self-attention fused networks (DCSAN) [48]. The basic BLS [6] is also used as the baseline model. The optimal final dimension of the extracted subspace is selected by searching $d = \{10, 20, \dots, 100\}$. The number of classifiers of DDA-BLS is $K = 3$, with corresponding $\epsilon = (0.3, 0.4)$. Other hyper-parameters are the same as those in the previous experiments. In the semi-supervised experiment setting, we randomly select 10% of the target samples with labels and use the rest of the target data to test the performance.

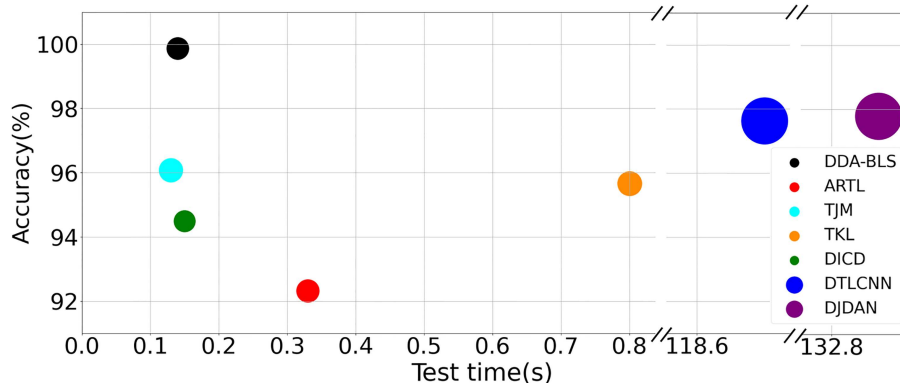
4.2.2 Experimental results

The fault diagnosis results of the CWRU data with 12 and 48 kHz are listed in Tables 3 and 4, respectively. Moreover, we show the results with 48 kHz of semi-supervised setting in Table 2. All the results show the superiority of the proposed DDA-BLS in DA tasks.

From Table 3, it can be seen that all DA methods achieve performance over 90% except the classical BLS, which shows that domain adaptation is necessary for fault diagnosis of rolling bearings with different working conditions. The performance of DDA-BLS is superior to all other compared methods. The sub-optimal of the deep learning-based method can be due to the limited number of training data in our experiments. Moreover, the superiority of DDA-BLS compared with DICD indicates that the implementation of a dynamic DA process can improve the DA performance. The results on the CWRU data with 48 kHz in Table 4 show similar trends as those in Table 3. The proposed DDA-BLS achieves an average accuracy of 77.02%, which is higher than other compared methods in the experiments. Moreover, it is

Table 4 Average fault diagnosis \pm standard deviation on the CWRU dataset (48 kHz) from source to target. The best results are in bold.

Task	BLS	ARTL	TJM	TKL	DICD	DTLCNN	DCSAN	DJDAN	DDA-BLS
HP0 \rightarrow HP1	50.35 \pm 0.50	65.63 \pm 0.00	57.13 \pm 0.00	68.98 \pm 0.00	63.95 \pm 0.00	69.90 \pm 0.80	69.29 \pm 0.57	70.31 \pm 0.64	72.34\pm0.12
HP0 \rightarrow HP2	50.32 \pm 0.67	64.55 \pm 0.00	56.57 \pm 0.00	64.63 \pm 0.00	64.48 \pm 0.00	69.76 \pm 0.75	68.57 \pm 0.85	69.72 \pm 0.16	72.12\pm0.16
HP0 \rightarrow HP3	52.00 \pm 0.42	60.15 \pm 0.00	55.50 \pm 0.00	65.28 \pm 0.00	62.98 \pm 0.00	68.49 \pm 0.11	67.50 \pm 0.46	69.03 \pm 0.32	69.26\pm0.01
HP1 \rightarrow HP0	55.41 \pm 0.18	67.70 \pm 0.00	60.35 \pm 0.00	67.95 \pm 0.00	65.08 \pm 0.00	75.44 \pm 0.61	72.49 \pm 0.18	75.85\pm0.17	70.86 \pm 0.02
HP1 \rightarrow HP2	70.13 \pm 0.81	75.20 \pm 0.00	62.63 \pm 0.00	81.07 \pm 0.00	82.17 \pm 0.00	86.34 \pm 0.72	84.43 \pm 0.37	86.13 \pm 0.53	86.98\pm0.14
HP1 \rightarrow HP3	67.65 \pm 0.29	71.13 \pm 0.00	60.03 \pm 0.00	77.23 \pm 0.00	75.00 \pm 0.00	78.32 \pm 0.40	76.86 \pm 0.11	78.14 \pm 0.33	79.72\pm0.11
HP2 \rightarrow HP0	50.35 \pm 0.51	64.90 \pm 0.00	57.75 \pm 0.00	64.33 \pm 0.00	64.22 \pm 0.00	74.64 \pm 0.56	73.21 \pm 0.72	75.20\pm0.67	73.60 \pm 0.02
HP2 \rightarrow HP1	68.96 \pm 0.55	75.08 \pm 0.00	60.87 \pm 0.00	79.98 \pm 0.00	81.73 \pm 0.00	85.38 \pm 0.16	83.64 \pm 0.69	86.53 \pm 0.53	87.30\pm0.07
HP2 \rightarrow HP3	69.94 \pm 0.55	77.40 \pm 0.00	64.65 \pm 0.00	81.37 \pm 0.00	76.20 \pm 0.00	81.36 \pm 0.24	79.14 \pm 0.45	82.83\pm0.37	82.12 \pm 0.23
HP3 \rightarrow HP0	48.24 \pm 0.29	65.18 \pm 0.00	57.70 \pm 0.00	64.50 \pm 0.00	61.78 \pm 0.00	67.53 \pm 0.45	65.78 \pm 0.34	66.80 \pm 0.51	68.25\pm0.18
HP3 \rightarrow HP1	61.92 \pm 0.24	71.55 \pm 0.00	57.95 \pm 0.00	76.40 \pm 0.00	73.05 \pm 0.00	77.02 \pm 0.32	76.22 \pm 0.87	78.12 \pm 0.89	79.37\pm0.27
HP3 \rightarrow HP2	66.78 \pm 0.59	75.35 \pm 0.00	61.68 \pm 0.00	79.88 \pm 0.00	76.72 \pm 0.00	80.06 \pm 0.12	79.73 \pm 0.31	81.05 \pm 0.16	82.33\pm0.04
AVERAGE	59.34	69.49	59.40	72.63	70.61	76.23	74.74	76.64	77.02

**Figure 6** (Color online) Performance vs. test time on CWRU. The size of the dot represents the training time, where a natural logarithm function (\ln) is employed for better visualization.

interesting that the performance of DA tasks containing the HP0 domain is, on average, lower than these tasks without the HP0 domain. As the HP0 means that there is no load on bearings, we indicate that the cross-domain knowledge can be more easily transferred among the working conditions with loads. Furthermore, the lower average accuracy of the results in Table 4 indicates that the DA tasks considering both the fault locations and fault types are much more challenging compared with the DA tasks only considering the fault locations.

The training and test time consumption of different DA methods on HP1 \rightarrow HP2 (12 kHz) is provided in Figure 6. Since the DDA-BLS contains the multi-exit architecture, its training and test time is slightly longer than DICD, ARTL, and TJM. However, the training procedure of DDA-BLS is still much faster than the two deep learning-based fault diagnosis DA methods. Moreover, due to the simplicity of SLFN architecture, DDA-BLS can run with faster inference speed than deep learning-based models. Although the test time of DDA-BLS can be a little bit slower than TJM, our method significantly outperforms the TJM in terms of accuracy. Overall, considering the performance, training, and inference time, the DDA-BLS can be viewed as the most effective and efficient DA method among the compared algorithms.

4.3 Case 3: DA for fault diagnosis with different bearing rigs

4.3.1 Experimental settings

The CWRU and IMS datasets [37] are utilized to assess DA capability for bearing fault diagnosis across different scenarios. The IMS dataset comprises data from four states sampled from three bearings: normal operation, IR fault, OR fault, and BALL fault, with a sampling frequency of 20 kHz. 1000 random samples were selected for each of the four operating states, with each sample consisting of 500 data points. Additionally, the CWRU data with 48 kHz (considering four different types: health, IR, OR, BALL) with 250 samples in each condition, totaling 1000 random samples across four load conditions are

Table 5 Cross dataset accuracy (%). The best results are in bold.

Method	CWRU \rightarrow IMS	IMS \rightarrow CWRU	AVERAGE
BLS	34.87 \pm 0.72	26.15 \pm 0.12	30.51
ARTL	50.37 \pm 0.00	46.30 \pm 0.00	48.34
TJM	51.78 \pm 0.00	38.70 \pm 0.00	45.24
TKL	41.92 \pm 0.00	36.93 \pm 0.00	39.43
DICD	41.58 \pm 0.00	47.43 \pm 0.00	44.51
DDA-BLS	78.64\pm0.10	77.32\pm0.09	77.98

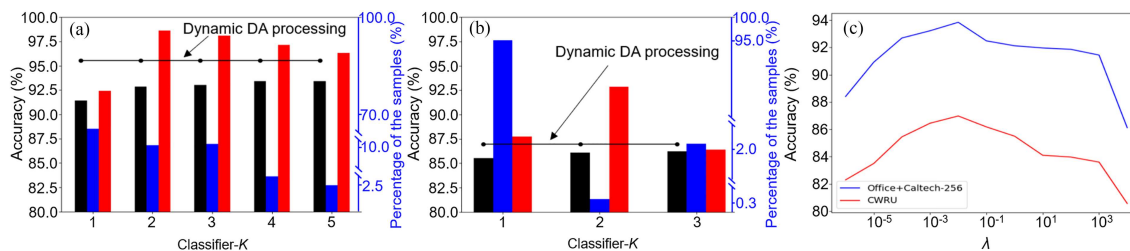


Figure 7 (Color online) Analytic experiments on Office+Caltech-256 ($C \rightarrow D$) and CWRU (HP1 \rightarrow HP2). In (a) and (b), the black bars represent the performance of DDA-BLS when all samples are forced to be classified by a certain classifier. The blue bars mean the ratios of samples exiting from the corresponding classifiers. The red bars are the accuracy of the exiting samples at the k th classifier. (a) Office+Caltech-256; (b) CWRU; (c) the effects of λ to DDA-BLS.

selected in experiments. By treating the IMS dataset and CWRU dataset as source and target domains, respectively, two cross-domain DA tasks can be established: “CWRU \rightarrow IMS”, and “IMS \rightarrow CWRU”. We compare the proposed methods with 4 classical DA methods including ARTL [20], TJM [46], TKL [19], and DICD [38]. The number of classifiers of DDA-BLS is $K = 3$, with corresponding $\epsilon = (0.5, 0.5)$. Other hyper-parameters are the same as those in the previous experiments.

4.3.2 Experimental results

From the results in Table 5, we can indicate that the DA task in Case 3 is much more difficult than the DA tasks in Case 2, since the test rig, running loads, and the sampling frequency are all different between the two domains. The proposed DDA-BLS significantly outperforms other evaluated DA methods by a large margin, which demonstrates the effectiveness of our method. The low detection accuracy of BLS means that the DA is necessary for many real-world applications, especially when the training and test samples have large distribution gaps.

4.4 Analytical experiments

4.4.1 Dynamic DA process in DDA-BLS

As we have stated in Subsection 3.1, the essential difference between the proposed DDA-BLS and the existing DA methods is the dynamic DA procedure, which enables the DDA-BLS to apply the data-dependent fine-grained feature learning procedure for each target input during domain adaptation, resulting in a more intelligent feature transformation procedure. Such a data-dependent DA process has rarely been explored in DA fields. Therefore, we first evaluate whether such a fine-grained dynamic DA procedure benefits the DA performance.

To verify the effectiveness of dynamic processing, we test the performance of DDA-BLS when all target samples are forced to exit from a certain classifier. From the results in Figures 7(a) and (b), we see that the later classifiers indeed achieve better performance than the first one since the enhancement of hidden nodes improves the generalization ability of DDA-BLS. The performance of dynamic DA inference is still superior to the final classifier, indicating the correctness of using a data-dependent DA procedure. Different target samples indeed need different feature learning procedures to achieve better performance in DA tasks.

We further provide the visualization results in Figure 8(a) to investigate what kind of target samples need to be enhanced in the dynamic DA process. The backpack and headphone classes in the $C \rightarrow A$ DA task are selected in experiments. The samples in the second row are the samples that achieve high prediction values and exit from Classifier-1 during inference. We found these samples to be strongly



Figure 8 (Color online) Visualization results of the samples with different complexities and t-SNE visualization (different colors represent different classes). (a) Images from the Office+Caltech-256 with different complexities; (b) visualization results.

similar to the samples from the source domain. Therefore, the trained DDA-BLS will consider these target samples “easy” ones. The third row shows the samples that exit from the final classifier. We found that these target samples normally contain more complex semantic information or the target objects show a non-typical appearance, and therefore, the enhancement features are necessary for accurate recognition. The trained DDA-BLS will consider these target samples “hard” ones. It is worth mentioning that the complexity of target samples is defined by the DDA-BLS itself rather than pre-defined. Interestingly, the definition of the complexity by the trained DA model meets our intuition, indicating that the dynamic DA procedure is indeed more intelligent compared with the static ones.

Moreover, the results in Figure 7(a) show that about 70% of target samples exit from the first classifier in the image classification DA task. Although the samples exiting from Classifier-1 achieve an accuracy of over 91.44%, the early existing “easy” target samples can still be wrongly classified by the early classifiers by high prediction values. Moreover, as most target samples (about 70%) will exit from the DDA-BLS from the first classifier, and only about 2.5% of target samples will recursively use all the classifiers during inference, the implementation of the multi-classifier architecture is not time-consuming.

4.4.2 Effects of λ to DDA-BLS

The λ is the coefficient of the regularization term in (5) of the classic BLS [6], which controls the ability of DDA-BLS to avoid the over-fitting problem. A lower value of λ can lead to the over-fitting issue of the learned classifier in BLS. A large λ can avoid the over-fitting problem while leading to a sub-optimal final performance. Therefore, selecting an appropriate value of λ is important to build a high-performance DA method. We use the same λ for all DDA-BLS classifiers to simplify the hyper-parameter selection procedure. The test results are shown in Figure 7(c), from which we see that the λ in (5) should be within an appropriate range to prevent the over-fitting problem and guarantee the high performance of DDA-BLS. From the results, we see that a small λ leads to an unsatisfied DA performance due to the over-fitting problem, which can be effectively addressed by increasing the value of λ . However, a too-large λ will affect the DDA-BLS to learn an accurate classifier, which also harms the final DA performance. Following the hyper-parameters strategy in [38, 49], we select the λ with the best performance on the target domain during performance evaluation.

4.4.3 2-D visualization of DI features

The t-SNE [50] visualization of the extracted features for task $C \rightarrow A$ of the proposed DDA-BLS and DICD are shown in Figure 8(b). From the results, we see that both methods can extract the DI features with class discriminative information, which can generally guarantee the separability among different target classes. Although the extracted features of DICD are more compact, there are a series of miss-classified samples (depicted by the red circle in the figure) that lead to the sub-optimal performance of DICD. Meanwhile, for DDA-BLS, due to the dynamic DA inference procedure, most of the target samples can be correctly recognized and gathered together. The high-quality class discriminative features can be extracted using the proposed method with fewer miss-classified samples, demonstrating the effectiveness of the proposed DDA-BLS.

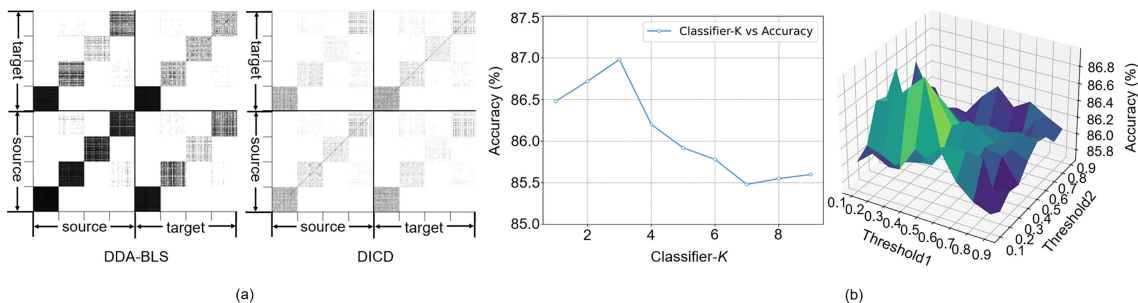


Figure 9 (Color online) Similarity of learned DI features from two domains (darker color means stronger similarity) and the effects of classifiers K and thresholds ϵ to DDA-BLS. (a) Domain similarity for HP1 \rightarrow HP2 (12 kHz) of DDA-BLS and DICD features; (b) DDA-BLS with different classifiers and thresholds.

4.4.4 Similarity of learned DI features

Following [38], we calculate the similarity matrix of learned domain invariant features to evaluate if a DA method successfully explores high-quality features. The results are shown in Figure 9(a), where each entry in the matrix measures the similarity of one data pair, where the similarity is calculated by the inner product of the learned features of two samples. To visualize the similarity matrix more clearly, all matrix values are binarized with a threshold of 0.5, meaning that we will keep the values larger than 0.5 and set the values to 0 when they are smaller than 0.5.

The CWRU with 12 kHz (HP1 to HP2) is used for similarity analysis. The top-left and bottom-right sub-matrix present the domain similarity between the source and target. From the top-right sub-matrix, we see that the DDA-BLS features show a strong correlation among samples from the same class and a weak correlation among samples from different classes. This indicates that the features extracted by DDA-BLS can effectively aggregate samples of the same class. Moreover, from the top-left and bottom-right sub-matrices, we see that the samples from the same classes have a high correlation between the source and target domain. This demonstrates that the extracted features are domain invariant, which leads to the high performance of the proposed methods in DA tasks.

4.4.5 Number of classifiers K and thresholds ϵ

We further evaluate how the number of classifiers and thresholds affect the performance of DDA-BLS. The results on CWRU HP1 to HP2 domain are shown in Figure 9(b). From the results, we see that a large number of classifiers leads to a decrease in performance, which can be due to the over-fitting problem caused by too many weight parameters. For CWRU, the optimal number of classifiers is 3. We further test the existing thresholds for the DDA-BLS with three different classifiers. The results show that the thresholds should be within an appropriate range to ensure that a portion of target samples can exit from the previous classifiers. However, in all ranges, the overall performance of DDA-BLS is higher than 85.5%, better than most evaluated methods in Table 4.

5 Conclusion

This paper proposes a novel broad learning system-based dynamic domain adaptation framework. Unlike most static DA models, the proposed DDA-BLS dynamically explores the DI features based on the property of the input samples. Such a data-dependent DA process considers the characteristics of inputs from the target domain, leading to a higher DA performance. The experiments demonstrate the effectiveness of our methods in DA tasks. Treating the target samples in a fine-grained data-dependent manner can benefit the knowledge-transferring process, which might bring a new aspect of view to improve the domain adaptation performance.

Acknowledgements This work was partially supported by National Science and Technology Major Project (Grant No. 2022ZD-0115803), National Natural Science Foundation of China (Grant No. 62206215), and China Post-Doctoral Science Foundation (Grant No. 2022M712537).

References

- Zhu J, Chen N, Shen C. A new deep transfer learning method for bearing fault diagnosis under different working conditions. *IEEE Sens J*, 2019, 20: 8394–8402

- 2 Zhu J, Chen N, Shen C. A new multiple source domain adaptation fault diagnosis method between different rotating machines. *IEEE Trans Ind Inf*, 2020, 17: 4788–4797
- 3 Yu W, Zhao C. Robust monitoring and fault isolation of nonlinear industrial processes using denoising autoencoder and elastic net. *IEEE Trans Contr Syst Technol*, 2019, 28: 1083–1091
- 4 He M, He D. Deep learning based approach for bearing fault diagnosis. *IEEE Trans Ind Applicat*, 2017, 53: 3057–3065
- 5 Peng J-X, Li K, Irwin G W. A new Jacobian matrix for optimal learning of single-layer neural networks. *IEEE Trans Neural Netw*, 2008, 19: 119–129
- 6 Chen C L P, Liu Z. Broad learning system: an effective and efficient incremental learning system without the need for deep architecture. *IEEE Trans Neural Netw Learn Syst*, 2017, 29: 10–24
- 7 Liu L, Cai L, Xie T, et al. Self-paced broad learning system. *IEEE Trans Cybern*, 2023, 53: 4029–4042
- 8 Tang H, Dong P, Shi Y. A construction of robust representations for small data sets using broad learning system. *IEEE Trans Syst Man Cybern Syst*, 2019, 51: 6074–6084
- 9 Yang K, Shi Y, Yu Z, et al. Stacked one-class broad learning system for intrusion detection in industry 4.0. *IEEE Trans Ind Inf*, 2022, 19: 251–260
- 10 Yu W, Zhao C. Broad convolutional neural network based industrial process fault diagnosis with incremental learning capability. *IEEE Trans Ind Electron*, 2019, 67: 5081–5091
- 11 Yang L, Yang Z L, Song S J, et al. Twin broad learning system for fault diagnosis of rotating machinery. *IEEE Trans Instrum Meas*, 2023, 72: 1–12
- 12 Yang L, Song S J, Chen C L P. Transductive transfer learning based on broad learning system. In: *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Miyazaki, 2018. 912–917
- 13 Gong X, Chen C L P, Zhang T. Cross-cultural emotion recognition with EEG and eye movement signals based on multiple stacked broad learning system. *IEEE Trans Comput Soc Syst*, 2024, 11: 2014–2025
- 14 Ben D S, Blitzer J, Crammer K, et al. Analysis of representations for domain adaptation. In: *Proceedings of Advances in Neural Information Processing Systems*, Vancouver, 2006. 19: 137–144
- 15 Schölkopf B, Platt J, Hofmann T. A kernel method for the two-sample-problem. In: *Proceedings of Advances in Neural Information Processing Systems*, Vancouver, 2006. 513–520
- 16 Gong B Q, Shi Y, Sha F, et al. Geodesic flow kernel for unsupervised domain adaptation. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, 2012. 2066–2073
- 17 Ding Z, Fu Y. Robust transfer metric learning for image classification. *IEEE Trans Image Process*, 2017, 26: 660–670
- 18 Xu Y, Fang X, Wu J, et al. Discriminative transfer subspace learning via low-rank and sparse representation. *IEEE Trans Image Process*, 2016, 25: 850–863
- 19 Long M, Wang J, Sun J, et al. Domain invariant transfer kernel learning. *IEEE Trans Knowl Data Eng*, 2014, 27: 1519–1532
- 20 Long M, Wang J, Ding G, et al. Adaptation regularization: a general framework for transfer learning. *IEEE Trans Knowl Data Eng*, 2013, 26: 1076–1089
- 21 Long M S, Cao Y, Wang J M, et al. Learning transferable features with deep adaptation networks. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML)*, Lille, 2015. 97–105
- 22 Pinheiro P O. Unsupervised domain adaptation with similarity learning. In: *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, 2018. 8004–8013
- 23 Long M S, Zhu H, Wang J M, et al. Deep transfer learning with joint adaptation networks. In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*, Sydney, 2017. 2208–2217
- 24 Xie B H, Li S, Li M J, et al. SePiCo: semantic-guided pixel contrast for domain adaptive semantic segmentation. *IEEE Trans Pattern Anal Mach Intell*, 2023, 45: 9004–9021
- 25 Li S G, Bu R H, Li S, et al. Principal properties attention matching for partial domain adaptation in fault diagnosis. *IEEE Trans Instrum Meas*, 2024, 73: 1–12
- 26 Wang B, Wen L, Li X Y, et al. Adaptive class center generalization network: a sparse domain-regressive framework for bearing fault diagnosis under unknown working conditions. *IEEE Trans Instrum Meas*, 2023, 72: 1–11
- 27 Wang L, Liu J, Zhang H, et al. KMSA-Net: a knowledge-mining-based semantic-aware network for cross-domain industrial process fault diagnosis. *IEEE Trans Ind Inf*, 2024, 20: 2738–2750
- 28 Chen X H, Yang R, Xue Y H, et al. Deep transfer learning for bearing fault diagnosis: a systematic review since 2016. *IEEE Trans Instrum Meas*, 2023, 72: 1–21
- 29 Qian Q, Qin Y, Luo J, et al. Cross-machine transfer fault diagnosis by ensemble weighting subdomain adaptation network. *IEEE Trans Ind Electron*, 2023, 70: 12773–12783
- 30 An Y, Zhang K, Chai Y, et al. Gaussian mixture variational-based transformer domain adaptation fault diagnosis method and its application in bearing fault diagnosis. *IEEE Trans Ind Inf*, 2023, 20: 615–625
- 31 He Y, Zhao C, Shen W. Cross-domain compound fault diagnosis of machine-level motors via time-frequency self-contrastive learning. *IEEE Trans Ind Inf*, 2024, 20: 9692–9701
- 32 Zhou Y, She Q, Ma Y, et al. Transfer of semi-supervised broad learning system in electroencephalography signal classification. *Neural Comput Applic*, 2021, 33: 10597–10613

- 33 Fu Y, Cao H R, Chen X F. Adaptive broad learning system for high-efficiency fault diagnosis of rotating machinery. *IEEE Trans Instrum Meas*, 2021, 70: 1–11
- 34 Peng S, Zeng R, Cao L, et al. Multi-source domain adaptation method for textual emotion classification using deep and broad learning. *Knowledge-Based Syst*, 2023, 260: 110173
- 35 Liu B, Zeng X, Tian F, et al. Domain transfer broad learning system for long-term drift compensation in electronic nose systems. *IEEE Access*, 2019, 7: 143947
- 36 Smith W A, Randall R B. Rolling element bearing diagnostics using the Case Western Reserve University data: a benchmark study. *Mech Syst Signal Processing*, 2015, 64–65: 100–131
- 37 Qiu H, Lee J, Lin J, et al. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *J Sound Vib*, 2006, 289: 1066–1090
- 38 Li S, Song S, Huang G, et al. Domain invariant and class discriminative feature learning for visual domain adaptation. *IEEE Trans Image Process*, 2018, 27: 4260–4273
- 39 Huang G, Chen D L, Li T H, et al. Multi-scale dense networks for resource efficient image classification. In: *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, Vancouver, 2018
- 40 Yang L, Han Y Z, Chen X, et al. Resolution adaptive networks for efficient inference. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, Seattle, 2020. 2369–2378
- 41 Huang G, Wang Y, Lv K C, et al. Glance and focus networks for dynamic visual recognition. *IEEE Trans Pattern Anal Mach Intell*, 2022, 45: 4605–4621
- 42 Wang Y L, Chen Z X, Jiang H J, et al. Adaptive focus for efficient video recognition. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, Montreal, 2021. 16249–16258
- 43 Pu Y F, Wang Y R, Xia Z F, et al. Adaptive rotated convolution for rotated object detection. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, 2023. 6589–6600
- 44 Han Y Z, Pu Y F, Lai Z H, et al. Learning to weight samples for dynamic early-exiting networks. In: *Proceedings of the 17th European Conference on the Computer Vision*, 2022. 362–378
- 45 Donahue J, Jia Y Q, Vinyals O, et al. DeCAF: a deep convolutional activation feature for generic visual recognition. In: *Proceedings of the 31st International Conference on Machine Learning (ICML)*, Beijing, 2014. 647–655
- 46 Long M S, Wang J M, Ding G G, et al. Transfer joint matching for unsupervised domain adaptation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, 2014. 1410–1417
- 47 Shen C Q, Wang X, Wang D, et al. Dynamic joint distribution alignment network for bearing fault diagnosis under variable working conditions. *IEEE Trans Instrum Meas*, 2021, 70: 1–13
- 48 Yu X, Wang Y J, Liang Z T, et al. An adaptive domain adaptation method for rolling bearings' fault diagnosis fusing deep convolution and self-attention networks. *IEEE Trans Instrum Meas*, 2023, 72: 1–14
- 49 Li S, Liu C H, Su L, et al. Discriminative transfer feature and label consistency for cross-domain image classification. *IEEE Trans Neural Netw Learn Syst*, 2020, 31: 4842–4856
- 50 Laurens V D, Geoffrey E H. Visualizing data using t-SNE. *J Mach Learn Res*, 2008, 9: 2579–2605