



Wheel condition assessment of high-speed trains under various operational conditions using semi-supervised adversarial domain adaptation



Si-Xin Chen ^{a,b}, Lu Zhou ^{a,b,*}, Yi-Qing Ni ^{a,b}

^a Hong Kong Branch of National Transit Electrification and Automation Engineering Technology Research Center, Hung Hom, Kowloon, Hong Kong Special Administrative Region

^b Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong Special Administrative Region

ARTICLE INFO

Communicated by Eleni Chatzi

Keywords:

Wheel condition assessment
Structural health monitoring
Deep learning
Transfer learning
Domain adaptation

ABSTRACT

Train wheels, among other components, are critical for the safety and ride comfort of high-speed rail systems. Various machine learning methods have been used together with onboard monitoring data to assess the wheel health conditions. However, only in some well-controlled experiments or authorized circumstances (source domain) can the well-labelled monitoring data for supervised learning be obtained. Even so, due to the difference in operational conditions, directly applying the model learned from this case to the case of interest (target domain) is not reliable. Facing this challenge, we propose an adversarial domain adaptation (DA) approach to transfer knowledge from a well-controlled monitoring test in one rail section to the rail section of interest. Since in the target domain, the data corresponding to components that are new or after reprofiling can be labelled as “intact”, the DA is modified to be semi-supervised rather than unsupervised. Two-level marginal and conditional DA is conducted in an adversarial manner, which can sufficiently eliminate the distribution discrepancy induced by the operational differences between two rail sections on which the train runs. Onboard monitoring data collected from the Lanxin high-speed rail section before and after wheel reprofiling is used as a case study. Results demonstrate the effectiveness of the approach as well as its superiority over three baseline models, and the underneath mechanisms are visualized. The study is expected to provide new thinking for the condition assessment for other key components when the train runs under various operational conditions.

1. Introduction

Wheel conditions are closely correlated with the dynamic behaviours of trains, influencing operation safety and ride comfort. In the era of high-speed rail (HSR), the health condition of wheels is particularly vital since impacts and vibrations induced by wheel defects and irregularities are more intense at higher running speeds and can lead to derailment under extreme circumstances. Therefore, regular wheel inspection and reprofiling are essential to maintain the integrity of wheel profiles. To acquire real-time information on

* Corresponding author at: Hong Kong Branch of National Transit Electrification and Automation Engineering Technology Research Center, Hung Hom, Kowloon, Hong Kong Special Administrative Region.

E-mail address: lu.lz.zhou@polyu.edu.hk (L. Zhou).

in-service wheels, structural health monitoring (SHM) methods have been increasingly adopted over the past decade. With sensors (strain gauges, accelerometers, etc.) implemented on either rail tracks or vehicles, wheel conditions including local defects, wear, and polygonization can be inferred by analysing the collected data such as wheel-rail forces [1–3] or accelerations [4,5].

Principally speaking, change of wheel conditions will be manifested as specific features in the monitoring vibration data of high-speed trains (HSTs) in either time domain [6] or frequency domain [7–9]. For data with relatively fewer uncertainties, traditional analytical methods, including a wheel flat index defined on time domain [6] or Short-time Fourier transform and wavelet transform [8,9], are adequate to identify the features and detect anomalies of interest. Hilbert-Huang transform (HHT) [10] or adaptive multiscale morphological filter [11] are also utilized, exhibiting more adaptability. However, HSTs are operating in an open environment. In most in-situ monitoring scenarios, the collected data is accompanied by great uncertainties, and the features are often obscured by noises and wheel-rail intense interactions, making it tricky to detect defects, let alone assessment of the evolving wheel conditions in the long term. To tackle the uncertainties and automatically extract features of interest, various machine learning approaches have been introduced, including random forest [12], Bayesian forecasting [13,14] and deep neural networks (DNNs) [15–17]. Specifically, DNNs prevent subjective judgments or labour-intensive feature handcrafting and thus have the best ability to separate the factors of variation. For example, a convolutional neural network (CNN) with multi-instance learning was proposed in [15] to improve the performance of identifying flat spots and non-roundness. Besides, [17] leveraged CNNs for wheel flat diagnosis based on images encoding the axle box acceleration signal and achieved a high separation index.

Owing to large-scale labelled data, DNNs exhibit excellent performance in solving a variety of engineering problems. However, as mentioned in [18], label information of structural conditions is not always known in SHM. For train wheels, normally, the only available condition information that can be labelled is “intact” when they are new or immediately after reprofiling. Unless in some rare cases, say, a well-controlled experiment, can the collected data correspond to different stages of the wheel wear. It would be highly helpful if the model for wheel condition assessment learned from a well-controlled experiment in one rail section could be used to aid model training in another rail section with limited label information.

On the other hand, although a deep learning model is expected to automatically eliminate the interference of irrelevant factors in the vibration data, the testing data may be collected in totally different operational conditions. As for the case in this study, distinguished environmental factors (temperature, soil condition, wind etc.) and rail infrastructures (trackbed, rail foundations, etc.) in different rail sections will alter the overall distribution of the collected vibration data even for the same wheel under the same health condition. This problem, in a machine learning sense, is called the distribution shift between training data and testing data [19]. As a result, when the model is directly applied to the testing data using conventional DNNs, the performance is far from satisfactory.

In view of the two concerns above, transfer learning (TL) [20] is considered to be a potentially efficient tool, which, crudely speaking, is to leverage the knowledge learned from one domain or task (source) to a different or related domain or task (target). To utilize the model from a different operational condition, we introduce domain adaptation (DA), which is a branch of TL. Its main philosophy is to find a transformation function to minimize or eliminate the distribution discrepancy between source and target domains (i.e., to narrow down their distance in a transformed metric space) [21–23].

DA was originally used in computer vision when images of the same object are collected under different conditions (such as light exposures) [24]. This strategy has been applied to some very recent research on fault diagnosis of machine elements (rolling element bearings and gearboxes) handling different kinds of distribution shifts. For example, the distribution of data collected under different working conditions for the machine fault diagnosis was synchronized in [25–28] by DA. Even when the data were not collected by a different sensor, DA could still be conducted [29]. In addition, some studies aimed to transfer DNN from one machine to another so that the latter can be diagnosed without any supervised training. For example, two methods were combined in [30] for bearing fault diagnosis, transferring among three machines, an electric motor, a shaft and a railway locomotive. Similarly, [31] adapted the model trained in the lab to locomotive bearings. Regarding transferring between civil structures, Worden et al. [32] have demonstrated three DA techniques on four case studies, providing a new framework. In another work by them, Transfer Component Analysis (TCA) was applied to transfer damage detectors between structures [33]. The operational conditions of trains suffer from much greater variations than the working conditions of other mechanical systems, making the investigation of DA worthwhile. There have been studies regarding applying DA to the SHM of railway engineering. For example, [30] involved the fault diagnosis of locomotive bearing and focused on the DA between different machines. Nevertheless, the studies that attempt to handle the operational condition variation of HSTs are very limited. The gap is to be bridged in this study.

In the present study, we propose a semi-supervised adversarial domain adaptation (SADA) approach for wheel condition assessment of HSTs operated in various scenarios. The study is an extension of the authors' previous work [34] in railway engineering, where we developed an acoustic-specific TL model for rail condition monitoring. In this study, the key point is to introduce a domain discriminator to the pre-trained model so that the features of all labelled data from one rail section (source domain) and all data from another (target domain) become indistinguishable through an adversarial process, and the discrepancy induced by operational condition difference can be eliminated. Compared with the artificially controlled operational conditions in lab in previous studies [25–28], the varying of operational conditions in field is more natural, making the integration of DA in this study more challenging and necessary. Moreover, apart from DA over the marginal distribution of all data, we leverage the part of the target data that are labelled as “intact” to conduct another conditional DA with the corresponding data category in the source domain to further refine the overall DA performance. As a result, the classifier that can assess wheel conditions can be transferred from the source domain to the target domain. To validate the proposed approach, axle box acceleration data collected from an in-situ monitoring test in Lanxin HSR line is used. The in-situ monitoring test contains well-annotated wheel condition information, and data collected from two rail sections with drastically different operational conditions in the Lanxin HSR line is chosen as source and target data, respectively. Three baseline models, without DA, without semi-supervision or without adversarial tactic, are also developed for comparison, and the underlying

mechanisms are presented in detail. This is an extension of our previous study [34], which transfers the knowledge learned from a large-scale acoustic dataset to our small-scale SHM dataset for rail condition assessment.

The rest of the paper is organized as follows. Section 2 introduces some basic concepts of TL and DA, formulates the real-scenario problem in this paradigm and proposes the SADA approach for solving the problem. Section 3 introduces the real onboard monitoring experiment for the validation, clarifies the source domain and target domain and lists the baseline methods. Results are shown and discussed in Section 4, with discussions and conclusions drawn in Section 5. For illustration and elaboration convenience, Table 1 lists the notations of math symbols and colours to be used in this paper.

2. Semi-supervised adversarial domain adaptation

Some critical concepts, including TL, feature space, label space, source domain, target domain and DA, are introduced before discussing the problem of wheel condition assessment in various operational conditions.

Technically, DA is a branch of TL, which might either transfer knowledge from a source domain to a different target domain or transfer knowledge from a source task to a different target task. For example, our previous study [34] transferred knowledge from one task (audio classification) to another task (rail condition assessment), which shares the same philosophy of TL but is subtly different from DA. Since this study focuses on DA, the detailed introduction of other branches is skipped.

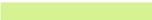
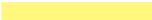
Probability distributions play a highly important role in many SHM applications [35]. Given a dataset $X = \{x^{(1)}, x^{(2)}, \dots, x^{(n)}\} \in X$ contains n feature samples and $Y = \{y^{(1)}, y^{(2)}, \dots, y^{(n)}\} \in Y$ consists of their corresponding labels. The task of supervised learning is to learn a model M for mapping $x^{(i)}$ to $y^{(i)}$ or to learn a conditional probability distribution $P(x)$. According to [36], the feature space X and the marginal distribution $P(x)$ constitute a **domain** $D = \{X, P(x)\}$. The label space Y and the relationship between x and y consists of a **task** denoted as $T = \{Y, P(x)\}$.

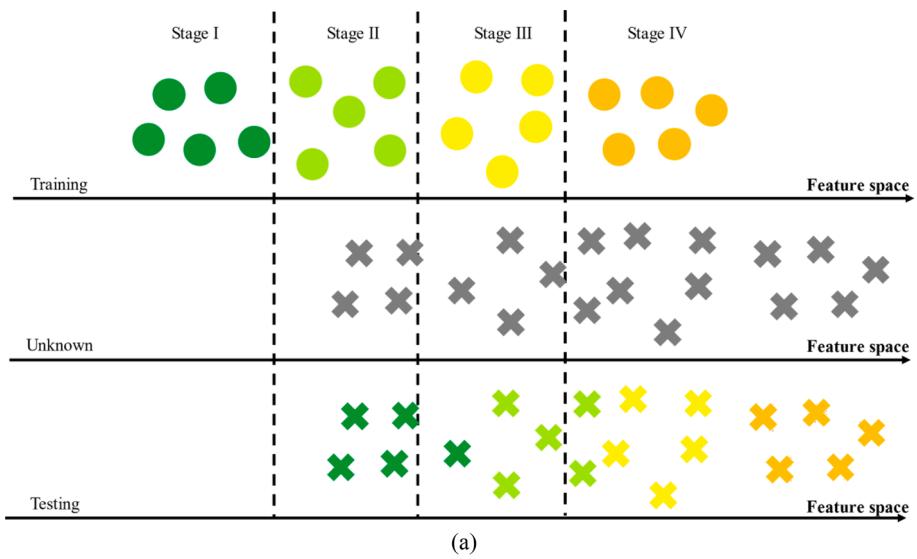
Traditionally, the model M , when well developed, will be directly applied to a real scenario, where all X , Y , $P(x)$ and $P(x)$ are assumed to remain the same. However, the real case can never be so ideal. Although the task remains the same, the $P(x)$ is likely to shift, and even the feature space itself can change [37]. In this case, the domain where the model is trained is called the **source domain**, while the new domain where the model is applied is denoted **target domain**. By definition, DA aims to adapt a model learned from the source domain to the target domain to maintain good performance on the task of interest [38].

2.1. Problem formulation and illustration

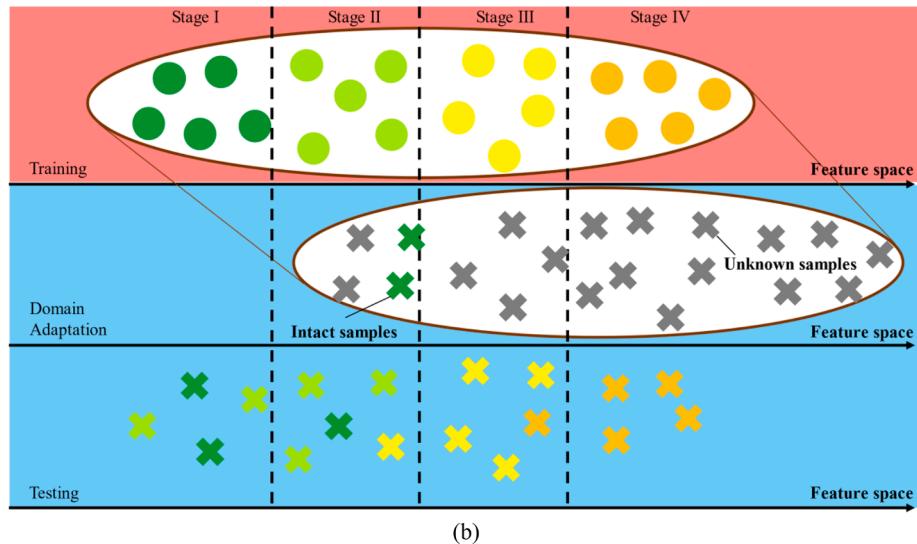
When an HST is running on one rail section, the vibration levels of its axle boxes can be continuously monitored. The vibration data can be divided into segments and form a dataset $Z_S = \{z_S^{(1)}, z_S^{(2)}, \dots, z_S^{(n)}\}$, where n is the number of segments. In some rare cases, for example, when the wheels are frequently examined, the wheel conditions can be known, forming $Y_S = \{y_S^{(1)}, y_S^{(2)}, \dots, y_S^{(n)}\}$. On this favourable basis, a model M , which is usually a CNN, can learn to extract feature $x_S^{(n)}$ from $z_S^{(n)}$ and map it to $y_S^{(n)}$ using a regressor or classifier C . These features $X_S = \{x_S^{(1)}, x_S^{(2)}, \dots, x_S^{(n)}\}$, in the feature space, are considered drawn from a distribution $P(x_S)$.

Table 1
Notations of math symbols and colours.

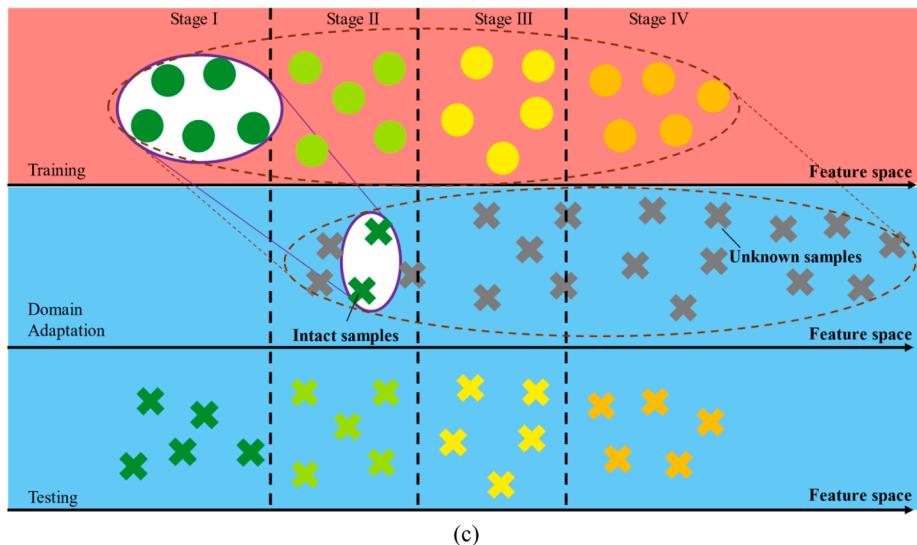
z_S, z_T	Original sample in source or target domain
x_S, x_T	Feature of sample in source or target domain
y_S, y_T	Label of sample in source or target domain
$P(x_S)$	Marginal distribution of source feature
$P(x_T)$	Marginal distribution of target feature
$P(x_S y_S = 0)$	Distribution of source feature condition on “intact”
$P(x_T')$	Distribution of target feature with labels
F	Feature extractor
C	Classifier for wheel condition assessment
D	Domain discriminator
	Source Domain
	Target Domain
	Stage I
	Stage II
	Stage III
	Stage IV
	Unlabelled Data
	SADA
	UADA
	DDC



(a)



(b)



(c)

(caption on next page)

Fig. 1. Illustration of (a) Domain shift leading to performance decay of classifier; (b) unsupervised DA keeping transferability of classifier; (c) semi-supervised DA further improving performance.

Nonetheless, it is a very practical scenario that the wheels are known to be intact when they are brand new or immediately after reprofiling. The target dataset is $Z_T = \{z_T^{(1)}, z_T^{(2)}, \dots, z_T^{(m)}\}$ and a small portion of them have labels denoted as $Y_T' = \{y_T'^{(1)} = 0, y_T'^{(2)} = 0, \dots, y_T'^{(m')} = 0\}$. The task in this study is to develop a model that can perform well on the remaining $m - m'$ target samples.

M cannot be directly applied to the target domain because of the difference of operational conditions between two rail sections. To be more specific, M can extract features $X_T = \{x_T^{(1)}, x_T^{(2)}, \dots, x_T^{(m)}\}$ that are drawn from $P(x_T)$, which is very likely to be different from $P(x_S)$. The potential consequence of this distribution shift is illustrated in Fig. 1(a). Each colour represents the health condition label $y^{(i)}$ of the wheel, and its location on the axis indicates the signal in the feature space $x^{(i)}$. The black dotted lines represent the classifier C learned from and applicable to the source domain. It is shown that, in the feature space, the well-trained C may misclassify many samples due to the distribution shift and is no longer sufficiently reliable.

DA, in this scenario, is to adapt the model M learned from $\{Z_S, Y_S\}$ for Z_T and overcome the interferences of operational conditions. As shown in Fig. 1(b), the feature distribution $P(x_T)$ is somehow aligned to the distribution $P(x_S)$ and the discrepancy between source and target domain is narrowed down. As a result, C become applicable in various operational conditions. The adversarial tactic involved will be introduced in Section 2.2.

Moreover, note that this is unsupervised DA since the labels Y_T' in the target domain are not leveraged. In other words, the information that wheels are intact at the beginning is wasted. To overcome this drawback, the semi-supervised DA scheme aims to align $P(x_T')$ to the conditional distribution $P(x_S|y_S = 0)$ where $P(x_T')$ is the distribution of feature with labels, besides aligning the marginal distribution $P(x_T)$ to $P(x_S)$. As shown in Fig. 1(c), this operation may help two domains to reach a consensus on the intact condition of wheels and thus improve the effect of DA.

It should be emphasized that the preliminary training of M , mentioned in the 1st paragraph of Section 2.1, is fully supervised since labels are available in the source domain. Regarding unsupervised DA and semi-supervised DA, they are all in terms of DA. In addition, Fig. 1 is merely a schematic illustration of the phenomenon of domain shift and the idea of SADA. The actual DA visualization will be presented in Section 4.

2.2. Procedures of preliminary training and SADA

To achieve the alignment of two domains in the feature space, as shown in Fig. 1, the adversarial tactic is adopted. The scheme of pre-training and SADA is illustrated in Fig. 2, and the details are explained as follows.

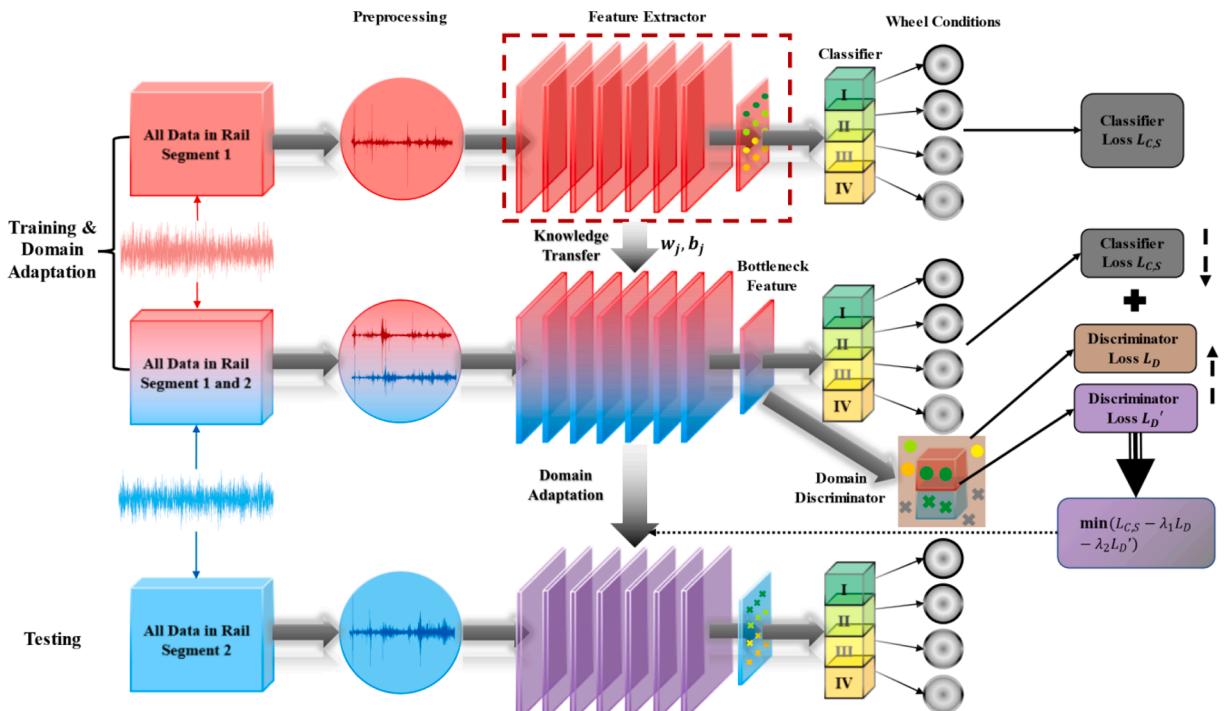


Fig. 2. Scheme of preliminary training and SADA.

2.2.1. Preliminary training

Before SADA, preliminary training on the source domain is necessary. The model M for wheel condition assessment is a CNN consisting of a feature extractor F and a wheel condition classifier C . Specifically, the F contains 7 convolutional layers, 7 activation layers, 7 pooling layers, 1 flatten layer and 1 fully-connected layer, while C consists of 2 fully connected layers and 2 activation layers. To avoid losing focus, the architectures of model will be skipped here and presented in Section 2.3.

F can transform the signal $z^{(i)}$ into feature $x^{(i)}$, based on which, C can obtain wheel conditions $y^{(i)}$. Specifically, it is found that the signal after discrete cosine transform (DCT) [39] is more suitable than the original time series as input to the model. Therefore, the i^{th} sample $z^{(i)}$ here represents the DCT form of the original signal.

In the beginning, the parameters of M , including those of F and C , are initialized by Xavier initialization [40], then M is preliminarily trained on the source domain. For each $z_S^{(i)}$ in the source dataset input to M , it outputs a prediction $\hat{y}_S^{(i)} = C(F(z_S^{(i)}))$, which is the vector that contains the probability of each class. There is also a corresponding one-hot vector $y_S^{(i)}$, such as $[0100]^T$, indicating the true class. The loss of classification $L_{C,S}$ in the source domain is cross entropy:

$$L_{C,S} = \frac{1}{n} \sum_{i=1}^n -[y_S^{(i)} \log \log \hat{y}_S^{(i)} + (1 - y_S^{(i)}) \log \log (1 - \hat{y}_S^{(i)})] \quad (1)$$

where n is the size of the source dataset Z_S .

$L_{C,S}$ is back-propagated according to the chain rule, and the parameters of both F and C are updated to gradually reduce $L_{C,S}$ in the process of traversing all mini-batches within the training set for many epochs.

2.2.2. SADA

DA is usually reduced to matching the feature distributions of the source and target domains. According to [41,42], if a feature can enable an algorithm to learn to identify its original domain, it is good for DA. In the deep learning paradigm, this can be done by updating the feature extractor F so that the features it extracts from the source and target domains are not distinguishable in terms of a distance measuring metric such as maximum mean discrepancy (MMD) [21,22] or by a discriminator [23]. The latter solution is selected, while the former one is used in the Baseline 3 approach as introduced in Section 3.3 and compared in Section 4.3. It should be emphasized that $P(z_S)$ and $P(z_T)$ cannot be changed but $P(x_S)$ and $P(x_T)$ can be aligned since the samples are transformed into the feature space X .

After preliminary training, a temporal network is created by attaching a domain discriminator D to M . As shown in Fig. 2, this network consists of three parts: F , D and C .

Based on the feature $x^{(i)} = F(z^{(i)})$ extracted by F , D can predict the domain where the sample is from. The discriminator loss is measured by binary cross entropy:

$$L_D = \frac{1}{n+m} \sum_{i=1}^{n+m} -[d^{(i)} \log \log \hat{d}^{(i)} + (1 - d^{(i)}) \log \log (1 - \hat{d}^{(i)})] \quad (2)$$

where m is the size of the target dataset Z_T ; $d^{(i)}$ is either 0 or 1, indicating source domain or target domain, respectively; $\hat{d}^{(i)} = D(F(z^{(i)}))$ is the predicted probability of target domain between 0 and 1.

The model training in terms of L_D is adversarial [43]. On the one hand, the discriminator D is updated to minimize L_D . On the other hand, the feature extractor F is trained to confuse D by maximizing L_D . As a result of this two-player game, the features generated by F will be increasingly domain invariant.

Since the label of a small portion of target samples are known, their counterpart in the source domain can be used, and the discriminator loss is:

$$L'_D = \frac{1}{n'+m'} \sum_{i=1}^{n'+m'} -[d^{(i)} \log \log \hat{d}^{(i)} + (1 - d^{(i)}) \log \log (1 - \hat{d}^{(i)})] \quad (3)$$

where m' is the number of samples in Z_T that are labelled as intact condition and n' is the number of samples in Z_S that are labelled as intact condition.

Table 2

Procedures of model training in SADA.

-
- For each epoch i :
- For each mini-batch j in Z_1 :
- Obtain a mini-batch from Z_2
 - Calculate L_D and its gradient in terms of parameters in D ; Update D to decrease L_D
 - Obtain a mini-batch from Z_3
 - Calculate L_D' and its gradient in terms of parameters in D ; Update D to decrease L_D'
 - Calculate $L_{C,S}$ and its gradient in terms of parameters in C ; Update C to decrease $L_{C,S}$
 - Calculate the gradient of $L_{C,S} - \lambda_1 L_D - \lambda_2 L_D'$; Update F to decrease it.
-

Meanwhile, the parameters of F and C are updated to maintain the ability to classify wheel condition by minimizing $L_{C,S}$. The purpose of SADA is to gradually increase the domain confusion L_D supplemented with L_D' while maintaining a low $L_{C,S}$. Finally, the adapted model denoted as M_{SADA} can be obtained by removing the domain predictor D , then be applied to unknown data in the target domain.

In practice, three datasets are involved in SADA: $Z_1 = \{Z_S, Y_S\}$ contains the source data and labels; $Z_2 = \{Z_T, Z_S\}$ contains both target data and source data; $Z_3 = \{Z_T', Z_S'\}$ contains those target data and source data corresponding to the intact stage. $L_{C,S}$, L_D and L_D' are calculated for mini-batches obtained from them rather than for the whole dataset. The training of D and the training of F and C are conducted in an alternating manner, and the procedures are summarized in [Table 2](#).

The mini-batch size n_b , λ_1 and λ_2 are set as 32, 0.1 and 0.05, respectively, after trial and error. The details of data for training, DA as well as testing will be introduced in [Tables 5, 6 and 7](#). The architecture of D , F and C will be introduced immediately.

2.3. Architecture of the network

The architecture of M (including F and C) and D are summarized in [Table 3](#) and [Table 4](#), respectively.

F consists of 8 blocks, and the main operations include Convolution (Conv), Restricted Linear Unit (ReLU) activation, maximum pooling (MaxPool), Fully Connected (FC) feedforward and Softmax activation.

A Conv layer consists of a set of learnable filters and is used to extract local feature maps [44]. Each filter is spatially small but extends through the full depth of the input volume. During the forward propagation, each filter slides across the width and height of the input volume and compute dot products between the weights of the filter and the entries of the receptive field (the region that the filter is looking at). This convolution can be considered as feature extraction and finally produces a 2D feature map containing the activations of that filter at every spatial position. The set of filters generates a number of feature maps. In summary, in a Conv layer numbered $[l]$ with $n_D^{[l]}$ learnable filters, the j^{th} filter generates a feature map $x_j^{[l]}$ from an input volume $x^{[l-1]}$:

$$x_j^{[l]} = w_j^{[l]} * x^{[l-1]} + b_j^{[l]} \quad (4)$$

where $*$ represents the convolutional operator; $w_j^{[l]}$ and $b_j^{[l]}$ are the weight volume and bias volume of the j^{th} filter, respectively. The stacked $n_D^{[l]}$ feature maps give the activations $x^{[l]}$.

The feature maps are passed through a nonlinear activation function, ReLU [45], which is elementwise and remains the size of the volume:

$$\text{ReLU}(x) = \max(x, 0) \quad (5)$$

A pooling layer can be used to shrink the volume of representation and reduce the number of parameters in the next layer to train. In this network, MaxPool is used, which takes the max over 4 numbers in every 2×2 region of the input volume. The layer can maintain the depth dimension n_D and disregard 75% of the previous activations with a stride of 2 on every depth slice.

At the beginning of Block 8, the feature maps are flattened into a vector (16384) and input to an FC layer. For the input volume $x^{[l-1]}$, the output of one FC layer is:

$$x^{[l]} = w^{[l]} \times x^{[l-1]} + b^{[l]} \quad (6)$$

To be specific, the input to F is the DCT of a vibration segment, denoted as $z = x^{[0]}$ and the output is the feature of this segment $x = F(x^{[0]}) = F(z)$, which is then fed to the classifier C .

C is a multi-layer perceptron (MLP) consisting of two Fully Connected (FC) layers with ReLU or Softmax as the activation function. The final score for four classes is given by:

$$\hat{y} = \sigma(w^{[L]} \times x^{[L-1]} + b^{[L]}) \quad (7)$$

where σ is the Softmax function, and L is the number of layers of M .

Table 3
Architecture of M .

Block		Operations in Block	Conv filters	Size of output representation
Input F	1	Conv → ReLU → MaxPool	8 1 × 3 filters	[1 × 4096 × 1]
	2	Conv → ReLU → MaxPool	16 1 × 3 filters	[1 × 2048 × 8]
	3	Conv → ReLU → MaxPool	32 1 × 3 filters	[1 × 1024 × 16]
	4	Conv → ReLU → MaxPool	64 1 × 3 filters	[1 × 512 × 32]
	5	Conv → ReLU → MaxPool	128 1 × 3 filters	[1 × 256 × 64]
	6	Conv → ReLU → MaxPool	256 1 × 3 filters	[1 × 128 × 128]
	7	Conv → ReLU → MaxPool	512 1 × 3 filters	[1 × 64 × 256]
	8	Flatten → FC		[1 × 32 × 512]
C	9	FC → ReLU		512
	10	FC → Softmax		64
				4

Table 4
Architecture of D .

Layer	Operations in Layer	Size of output representation
Input		512
1	FC → ReLU	64
2	FC → Sigmoid	1

Table 5
Data size for DA in Setting HS → MM.

Source domain (Hami-Shanshan, HS)		Target domain (Menyuan-Minle, MM)		
Stage I (Z_S')	Training set (Z_S) and Evaluation set	8052	Stage I	Adaptation set (Z_T')
		6589	Stage II	Testing set (Z_T)
Stage II		10,248	Stage III	
Stage III		10,248	Stage IV	
Stage IV				4389

Table 6
Data size for DA in Setting MM → HS.

Source domain (Menyuan-Minle, MM)		Target domain (Hami-Shanshan, HS)		
Stage I (Z_S')	Training set (Z_S) and Evaluation set	4758	Stage I	Adaptation set (Z_T')
		4758	Stage II	Testing set (Z_T)
Stage II		4758	Stage III	
Stage III		4758	Stage IV	
Stage IV		4392		10,248

Table 7
Data size for DA in Setting HS → ST.

Source domain (Hami-Shanshan, HS)		Target domain (Menyuan-Minle, MM)		
Stage I (Z_S')	Training set (Z_S) and Evaluation set	8052	Stage I	Adaptation set (Z_T')
		6589	Stage II	Testing set (Z_T)
Stage II		10,248	Stage III	
Stage III		10,248	Stage IV	
Stage IV				10,248

Some critical hyper-parameters of the architecture, including the number of layers of F and the size of each hidden layer of C , are selected by cross-validation. Specifically, the source dataset is randomly partitioned into a training set and an evaluation set. For one hyper-parameter setting, M is first trained on the training set then tested on the evaluation set. After iterating all the settings, the architecture with the lowest $L_{C,S}$ was selected.

D is also an MLP consisting of two FC layers. Unlike M whose architecture can be optimized on the Evaluation set, D has nothing to rely on. Thus, its number of layers and size of hidden layer follows C . It intakes the feature from F , and the output is:

$$\hat{d} = \text{Sigmoid}(w^{[2]} \times x^{[1]} + b^{[2]}) \quad (8)$$

3. Case study

3.1. Onboard monitoring test in Lanxin HSR line

The authors' research team conducted a one-month onboard monitoring test in the Lanxin HSR line in the Chinese Mainland from 15th December 2015 to 15th January 2016. The Lanxin HSR line starts from Lanzhou in Gansu Province to Urumqi in Xinjiang Province with a total length of 1776 km. As it can be seen in Fig. 3 (geometry retrieved from Google Earth, Google LLC.), the surrounding environment of the Lanxin HSR line is complicated and volatile. The Lanxin HSR line passes through Qilian Mountains and Turpan Basin, making it the rail line with the largest elevation difference in the world. As stated in Section 1 and Section 2, the complex environment would lead to a distribution shift of the collected data by directly altering the dynamic responses or indirectly influencing the infrastructure design and construction (trackbed, rail foundations, etc.). Hence, the onboard monitoring test can be an excellent case to validate the proposed DA model.

A comprehensive monitoring system was deployed on two coaches of an in-service CRH-V type passenger HST. The sensors include accelerometers, strain sensors, temperature sensors, and wind pressure sensors which were deployed on bogie frames, bump stops, axle



Fig. 3. The Lanxin HSR line (in blue mark) and surrounding terrain.

boxes, and car bodies, respectively. Previous research work by the authors on ride comfort using accelerometers mounted on car bodies can be found in [14]. Since the vibration responses of axle boxes can effectively reflect the interaction between wheel and track without the involvement of primary and secondary suspension systems [11], in this study, we use the acceleration data of two axle boxes for wheel condition assessment.

In terms of specific sensor positions, as shown in Fig. 4, one accelerometer was installed on the axle box of the trailing bogie of the power car, and three were on the axle boxes of the leading bogie of the driven car respectively, each sensor collected vibration responses of one wheel correspondingly. The dynamic range of the accelerometer is ± 1000 g, and the sampling rate was 5000 Hz.

The system installation and the onboard monitoring test are shown in Fig. 5. The data acquisition system was placed in one of the tested coaches with encapsulated wires connecting the sensors and the equipment. The monitoring period lasted one consecutive month except for the wheel reprofiling day on 31st December 2015. The train departed from Lanzhou to Urumqi on the first day and

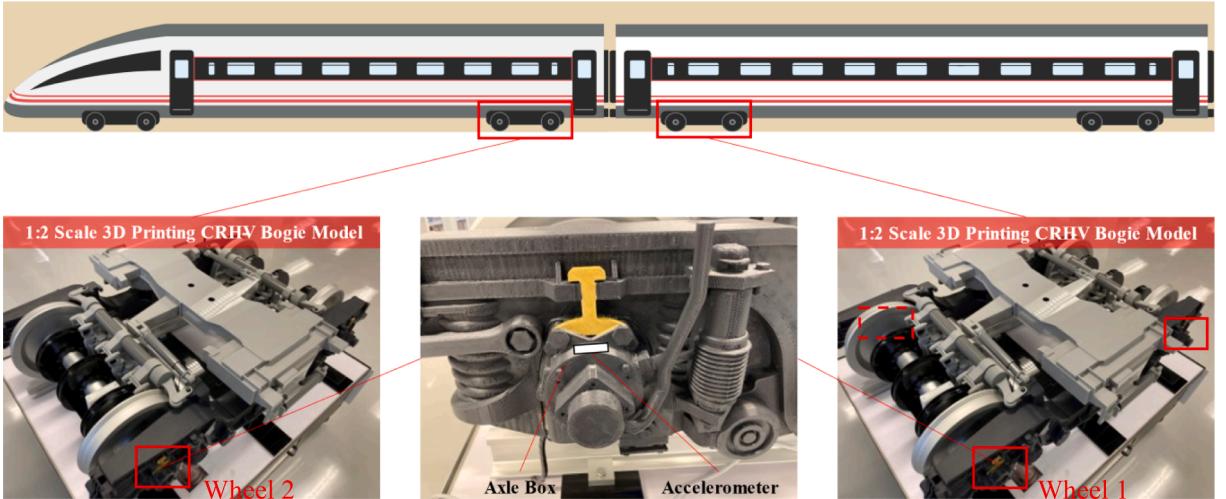


Fig. 4. Positions of accelerometers on axle boxes.

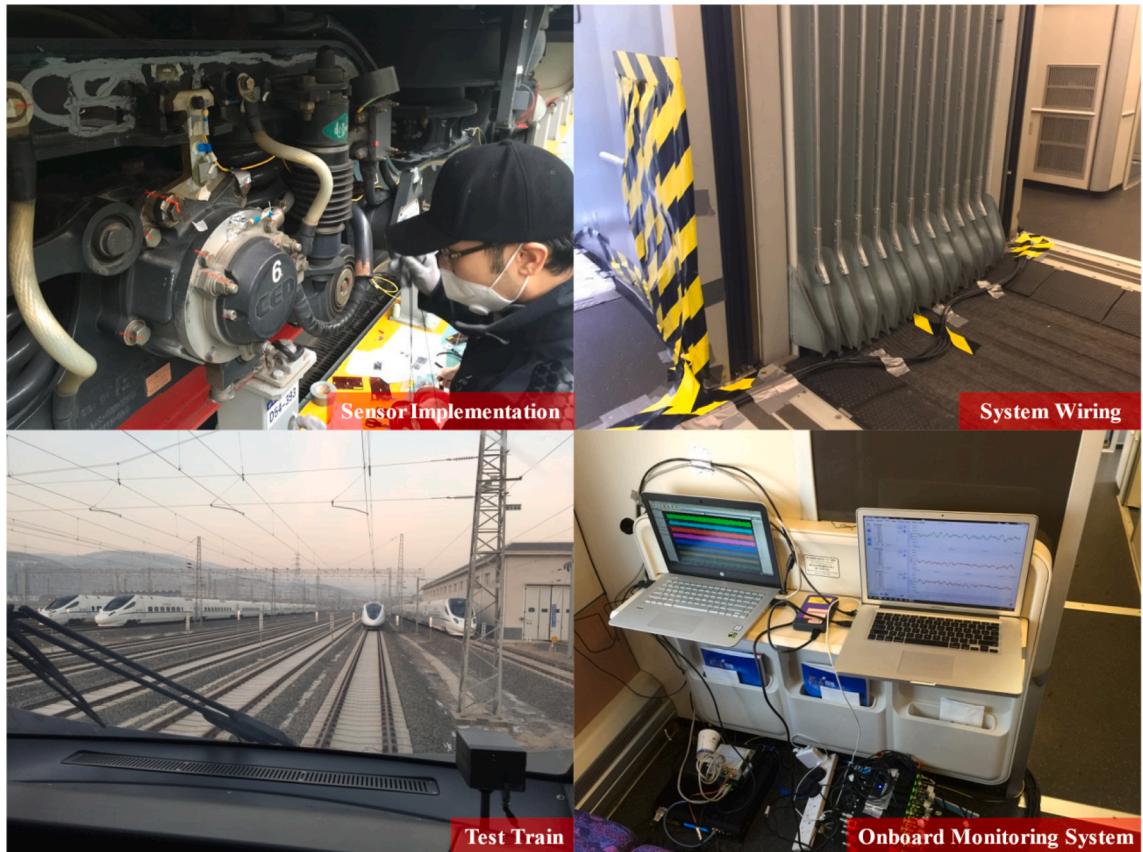


Fig. 5. Implementation of monitoring system and onboard testing.

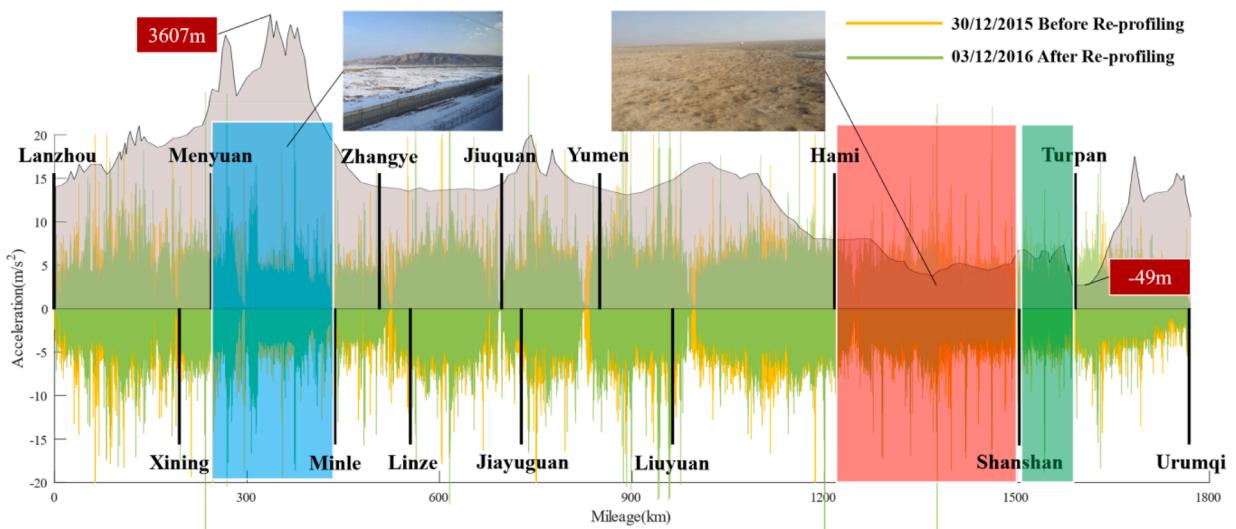


Fig. 6. Acceleration data series before and after reprofiling and the elevation profile.

returned to Lanzhou on the second day to cover a round trip.

3.2. Data for preliminary training, DA, and testing

Two acceleration data series collected before and after wheel reproving throughout the entire running length are plotted together with the elevation profile with respect to sea level. As can be seen in Fig. 6, the highest point is 3607 m in the section near Qilianshan No. 2 Tunnel, while the lowest point in Turpan Basin is -49 m. As seen in the pictures taken during the monitoring period, the surrounding environments in these two segments are obviously different. The great elevation difference in two rail sections causes a great difference in temperature, lateral winds, soil condition, etc. Specifically, temperature change alters wheel-rail contact conditions [46–48], lateral winds induce additional lateral creepage [49,50] in wheel-rail contact. For high-elevation rail sections in Lanxin rail line, seasonal tundra is a common factor to be considered [51]. All these factors either directly alter the dynamic responses or indirectly influence the infrastructure design and construction (trackbed, rail foundations, etc.) and eventually result in a distribution shift of the monitoring data in a comprehensive manner.

Therefore, we choose data collected from the same sensor reflecting conditions of the same wheel in the Menyuan-Minle (MM) section (highest average elevation) and data collected in the Hami-Shanshan (HS) section (lowest average elevation) to validate our model. The Shanshan-Turpan (ST) section was also selected, which shares a similar elevation as the HS section. Totally, there were three DA settings, HS → MM, MM → HS and HS → ST. Setting HS → MM and Setting MM → HS represent a large variation of operational conditions for HSTs, while Setting HS → ST represents a small operational condition change. The data sizes of these settings are summarised in Tables 5, 6 and 7, and the results are presented in Section 4.1 and Section 4.4.

In Fig. 6, despite the acceleration level before reproving is significantly higher than that after reproving and the difference can be observed directly, the change in wheel conditions is actually an evolutionary process. Our primary target is to assess the evolving wheel conditions in a regressive manner using our learning model rather than providing a binary classifier.

According to the maintenance strategy provided by the rail operator, the train wheels operating in the Lanxin HSR line were reproved every 150,000 km, which is less than the reproving mileage of conventional rail lines due to the harsh and high-speed operational conditions. Although over the one-month monitoring period, the train passed approximately 55,000 km mileage, covering only 1/3 of one full-service life cycle between two reproving points, it is confident that the monitoring data is sufficient to cover all critical stages of wheel conditions considering the mechanism of wheel wear, which is the primary manifestation of wheel conditions. Referring to a series of theoretical and experimental research work on wheel wear evolution [52], the wheel wear mechanism is three-fold, as schematically illustrated in Fig. 7. When a wheel is newly reproved, the rate of material removal is relatively high (*intermediate* wear); As the mileage increases, a strain hardening layer (normally 1 mm thick) is formed, and the wear rate drops dramatically, entering a long *mild* wear period; In the final part of one service life cycle, the wear rate increases critically again (*intermediate* and *severe* wear) as the strain hardening layer is abraded gradually. By measuring the change of wheel flange thickness and wheel tread profile with a mini-profilometer out of operating hours every day, it was confirmed by the rail operator that axle box vibration data collected before reproving covers period of *mild-intermediate* wear and *intermediate-severe* wear, and data collected after reproving covers period of *intermediate* wear and *intermediate-mild* wear.

Based on this, we adopt a data labelling strategy similar to our previous work in [34] and label the wheel into four discrete stages as a compromise to demonstrating the regression capability of the proposed model with SHM data collected in a limited period, specifically, two days. Since the reproving day of the test lies exactly in the middle of the monitoring period, Stage I and II are selected from the data after reproving, and stage III and IV are from the data before reproving with a gap period left in between to avoid label

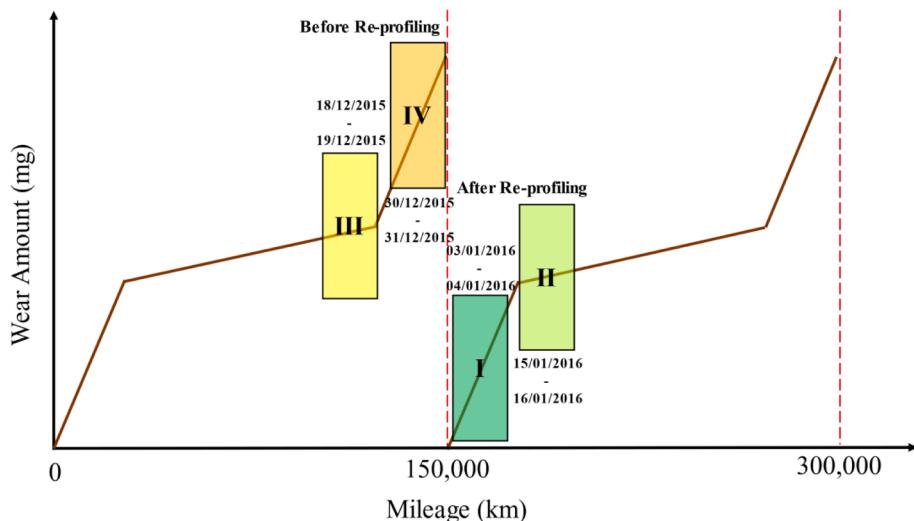


Fig. 7. Schematic of wheel condition (wear) evolution and data labelling.

overlapping, as shown in Fig. 7. It should be emphasized that the wear amount of one wheel was very slight within two days. Therefore, the health condition of one wheel was considered consistent in these two days, and the corresponding data were tagged as the same labels.

Under this labelling strategy, the entire Stage II to III interval should lie in the *mild* wear zone following a slow wear evolution mechanism and no additional label is needed in between. In this regard, we are able to fully reveal the progressive wheel conditions even though we do not have monitoring data of a complete life cycle.

We verify the proposed approach on two of the monitored wheels, denoted as Wheel 1 and Wheel 2, respectively. One is on an axle box of the power car, and the other is on an axle box of the driven car. The investigation was based on axle box acceleration data, and only data collected during cruising speed are used. They are divided into segments, each of which covering 0.8192 s and at least 10 circles of wheel rotation under an operation speed of 200 km/h. Each sample contains 4096 data points. As mentioned in Section 2.2, DCT is used since this simple transformation proves to be a better representation than the original time series segments. Totally, there were three DA settings, HS → MM, MM → HS and HS → ST. The data sizes of these settings are summarized in Tables 5, 6, and 7, which respond to the SADA strategy presented in Table 2. 10% of data from the source domain are separated for model selection of M (Section 2.2) and Baseline 0. A small portion of data from the target domain are separated as Z_T' .

It should be noted that in real practice, we can only access the training set, the adaptation set, and the unlabelled testing set to be evaluated, as one can trace back to the problem description in Section 2.1. In this case, fully supervised learning in the target domain is not possible and motivates the SADA approach. From a research perspective, the well-controlled experiment provides a chance to validate the SADA approach because a wheel with the same health condition works in quite different operational conditions. In this regard, the labels of the testing set are only available for research purposes and no longer exist in real practice.

3.3. Baseline methods for comparison

With the data collected from field monitoring, it is possible to validate the proposed approach. Following Section 2.2, model M is preliminary trained on the Training set for 20 epochs. Then, five different procedures distinguish different approaches, including SADA, and four baseline methods are used to solve the same task for comparison.

SADA: M is adapted following Table 2 by SADA for another 20 epochs, and the model after SADA is tested on the Testing set (Stage I, II, III, IV) from the target domain. This process corresponds to Fig. 1(c) and Fig. 2.

Baseline 0 (Evaluation): The pre-trained model M is trained for another 20 epochs on the Training set and then tested on the Evaluation set.

Baseline 1 (No DA): The pre-trained model M is trained for another 20 epochs on the Training set without DA and then directly testing on the Testing set, which corresponds to Fig. 1(a). By comparison, one can see how DA can significantly avoid performance decay.

Baseline 2 (UADA): A classical method based on adversarial training is used [23]. The pre-trained model M is adapted for the combined Adaptation set and Testing set by adversarial training with no semi-supervision involved. In other word, L_D is not included in the Loss function for updating F . This process corresponds to Fig. 1(b).

Baseline 3 (DDC): The idea of deep domain confusion (DDC) [21] is adopted with no adversarial training involved, but semi-supervision is utilised. The features become domain invariant in the process of maximizing the domain invariance measured by MMD:

$$L_D = MMD_{S,T} = \left[\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n k(x_{S(i)}, x_{S(j)}) + \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m k(x_{T(i)}, x_{T(j)}) - \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m k(x_{S(i)}, x_{T(j)}) \right]^{\frac{1}{2}} \quad (9)$$

In this study, two kinds of kernels were used, including linear kernel $k(x^{(i)}, x^{(j)}) = \langle x^{(i)}, x^{(j)} \rangle$, and radial basis function (RBF) kernel:

Table 8

Confusion matrix by M_{SADA} (HS → MM): (a) Wheel 1; (b) Wheel 2.

(a)					
Prediction Label	Stage I	Stage II	Stage III	Stage IV	Recall
Stage I	1975	151	31	38	90.0%
Stage II	279	4256	199	22	89.5%
Stage III	22	107	4310	317	90.6%
Stage IV	36	125	419	3809	86.8%
Precision	85.4%	91.7%	86.9%	91.0%	F1: 89.0%

(b)					
Prediction Label	Stage I	Stage II	Stage III	Stage IV	Recall
Stage I	2025	110	38	22	92.3%
Stage II	340	4206	147	63	88.4%
Stage III	94	200	4298	164	90.4%
Stage IV	22	2	631	3734	85.1%
Precision	81.6%	93.1%	84.0%	93.7%	F1: 88.4%

$$k(x^{(i)}, x^{(j)}) = \exp\left(-\|x^{(i)} - x^{(j)}\|^2 / (2\sigma^2)\right) \quad (10)$$

where σ is the kernel width parameter and selected as 1 here.

Model preliminary training, DA and testing process was conducted on a workstation with an Intel(R) Core(TM) i7-7700HQ 2.8 GHz processor, 12 GB RAM, and an Nvidia GTX2070 Graphic Card.

4. Results and discussions

4.1. Performance of SADA compared with baselines

The model after SADA is denoted as M_{SADA} and used for the wheel condition assessment of Wheel 1 and Wheel 2 based on the Testing set in setting HS → MM. The inferred wheel condition is the category with the highest probability. For each category, the recall and precision are calculated, and the F1 score is:

$$F1_i = 2 \frac{recall_i \times precision_i}{recall_i + precision_i} \quad (11)$$

The mean of F1 score for four conditions, called macro-F1, is used as a summarized metric:

$$F1 = \sum_{i=1}^4 F1_i \quad (12)$$

[Table 8\(a\)](#) and (b) show the confusion matrix of predictions by M_{SADA} (HS → MM) on Wheel 1 and Wheel 2, respectively. The macro F1 score is 89.0% and 88.4%, respectively. It should be noted that the testing data, based on which the inference is made, are collected under an operational condition different from the training data and are unseen by M_{SADA} until the very last moment.

Similarly, the confusion matrix by M_{SADA} (MM → HS) is summarized in [Table 9\(a\)](#) and (b). The macro F1 score is 79.4% and 82.2%, respectively. This performance decay compared with the previous setting should be ascribed to the size of source dataset. The dataset of Hami-Shanshan is about twice larger than that of Menyuan-Minle, as noted in [Table 5](#) and [Table 6](#), and a larger source dataset can nurture more transferable features by better disentangling explanatory factors of variations [53,54] for the following wheel condition assessment.

Four models developed by four baseline methods (Evaluation, No DA, UADA and DDC) were tested on the same Testing sets. The results in terms of the F1 score are compared in [Fig. 8](#) and [Fig. 9](#).

Take Wheel 1 in setting HS → MM for example. When the model is trained on the training set and tested on the Evaluation set, which are both in the source domain, the performance is quite good, with an F1 score of 98.9%. However, when directly applied to the Testing set, it exhibits a serious performance decay, F1 score dropping to 74.8%. As mentioned in the Introduction, this should be ascribed to the different environmental and operational conditions. Fortunately, if SADA is applied, the F1 score can be recovered to nearly 90%, with a confusion matrix shown in [Table 8\(a\)](#). In comparison, when UADA is applied, the F1 score only improves to 85.1%. Finally, when the normal DDC approach is applied, the F1 score can be improved from 74.8% to 81.0%, meaning that DDC is also effective in our scenario, although not as effective as SADA and UADA.

Similar phenomena can be observed for Wheel 2 in setting HS → MM and for Wheel 1 and Wheel 2 in setting MM → HS. To reveal the underneath mechanism, the setting HS → MM will be emphasized and discussed. [Section 4.2](#) explains how SADA make the assessment under various operational conditions possible and why it exhibits superiority to UADA. [Section 4.3](#) shows the advantages of SADA in comparison with DDC.

Table 9

Confusion matrix by M_{SADA} (MM → HS): (a) Wheel 1; (b) Wheel 2.

(a)					
Prediction Label	Stage I	Stage II	Stage III	Stage IV	Recall
Stage I	2448	354	26	96	83.7%
Stage II	1080	4914	16	570	74.7%
Stage III	5	158	8077	2000	78.9%
Stage IV	6	91	1605	8538	83.4%
Precision	69.2%	89.1%	83.1%	76.2%	79.4%

(b)					
Prediction Label	Stage I	Stage II	Stage III	Stage IV	Recall
Stage I	2736	106	80	2	93.6%
Stage II	676	5293	300	311	80.4%
Stage III	286	284	9151	519	89.4%
Stage IV	252	299	2124	7565	73.9%
Precision	69.3%	88.5%	78.5%	90.1%	82.2%

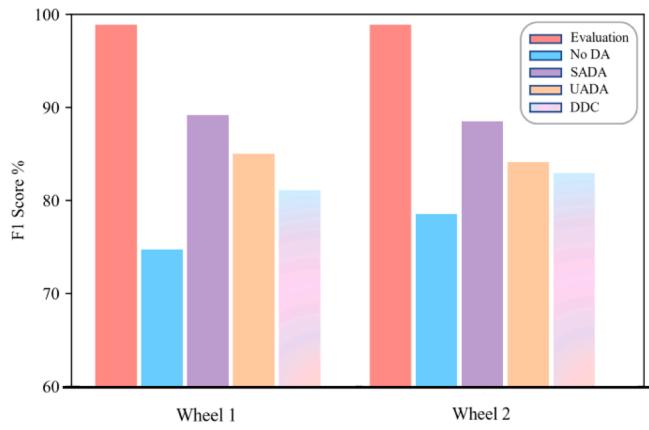


Fig. 8. Performance comparison in terms of F1 score in setting HS → MM.

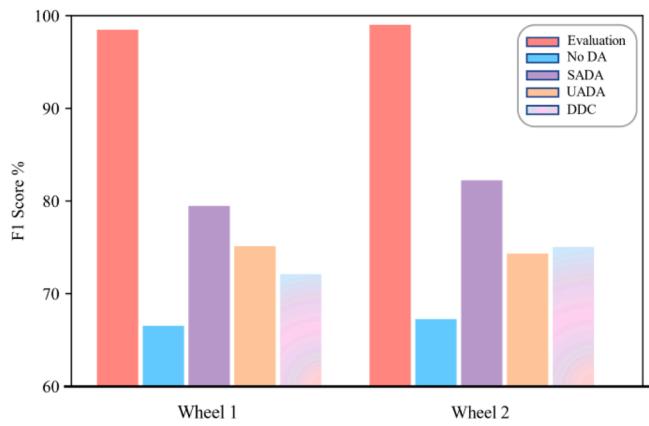


Fig. 9. Performance comparison in terms of F1 score in setting MM → HS.

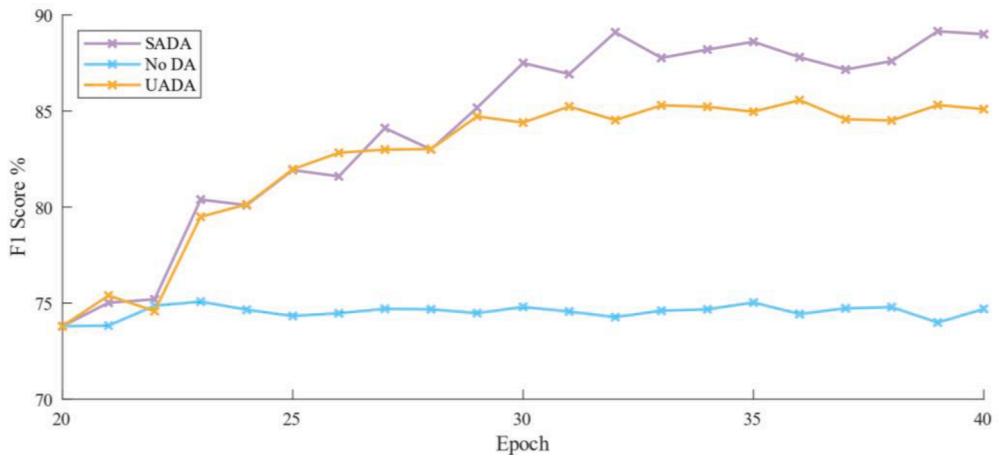


Fig. 10. The change of F1 score for Wheel 1 during training or adaptation.

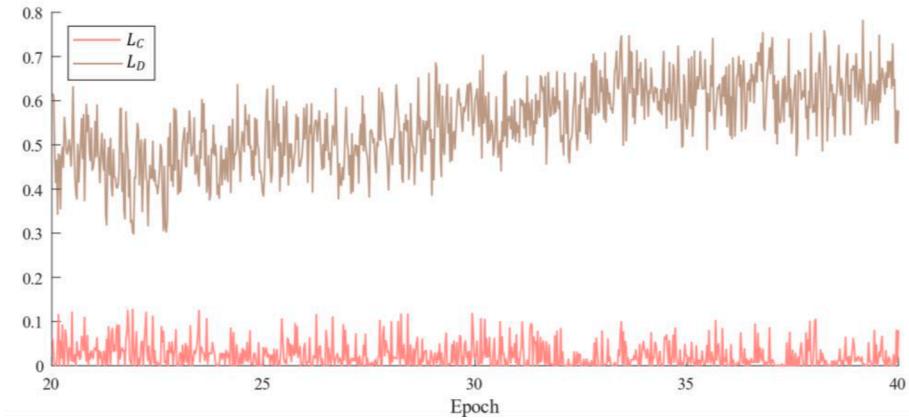


Fig. 11. Classifier loss and discriminator loss during UADA for Wheel 1.

4.2. Illustration of adaptation: UADA vs SADA

Wheel 1 will be focused on the following illustrations. After 20 epochs of preliminary training on domain HS, the model can achieve an F1 score of 73.8% on domain MM, which is the initial point in Fig. 10. Then, as shown by the blue curve, if another 20 epochs are conducted without any DA, the F1 score has little improvement to 74.8%. In comparison, significant and continuous improvement of F1 score (from 73.9% to 88.9%) can be observed when SADA is applied. Although UADA is able to improve the F1 score in the first several approaches, it converges earlier than SADA and ends up with an F1 score of 85.1%.

The change of discriminator loss L_D and the classifier loss L_C during the 20 epochs of UADA are presented in Fig. 11, where L_C remains at a low level, smaller than 0.1, while L_D turbulently and gradually rises and begins to converge at around 0.65 at about epoch 32. The implication is that the classifier C maintains its ability to identify four wheel conditions while the feature extractor F attempts to learn features that are simultaneously domain-confusing and wheel condition-sensitive. The correspondence between Figs. 10 and 11

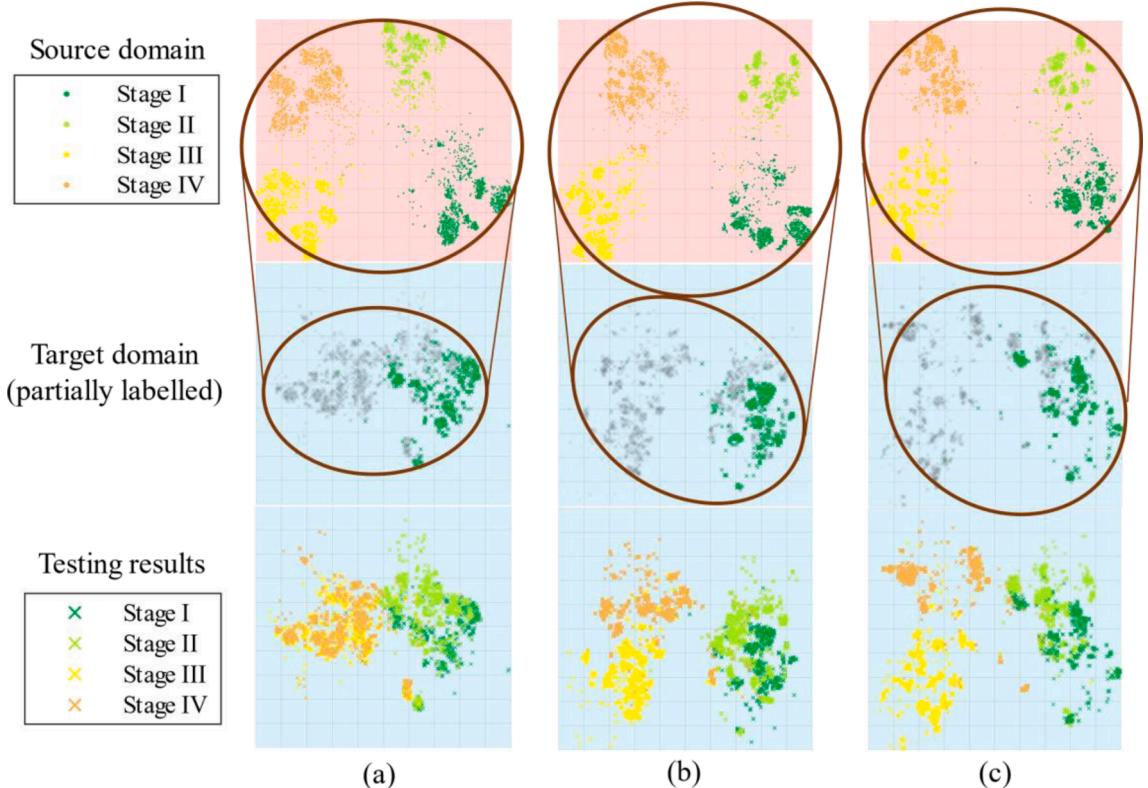


Fig. 12. Evolvement of features for Wheel 1 during UADA: (a) epoch 20, immediately after preliminary training; (b) epoch 30; (c) epoch 40.

verifies the claim in [41,42] that if a feature can enable an algorithm to identify its original domain, it is good for DA.

To intuitively illustrate how UADA and SADA change the domain discrepancy, T-distributed Stochastic Neighbour Embedding (t-SNE) [55], as a nonlinear dimensionality reduction technique for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions, is used to visualize the features generated by the feature extractor F . The feature visualizations are displayed in a way the same as what has been illustrated in Fig. 1(b) and (c), respectively, to provide an impression from strategy to results.

The evolution of features during UADA is presented in Fig. 12. In (a), after preliminary training, the features in the source domain can be well separated (upper subfigure). However, in the target domain (middle subfigure), the features are clustered, and the feature distribution is different from that in the source domain. If the unknown samples are coloured according to their labels (which are usually not available in real cases), as shown in the below subfigure, the separation rule encoded in the classifier C can be found to fail to distinguish features in the target domain.

As shown in (b), after 10 epochs of UADA, the features in the target domain (middle) gradually diffuse so that their distribution can become closer to that in the sourced domain (upper). It should be noted that the feature distribution in the source domain does not remain unchanged because both the feature extractor F and the classifier C are updating during UADA.

Finally, after 20 epochs of UADA, the feature distribution in the target domain is closer to that in the source domain, as shown in (c). Meanwhile, the updated classifier C can be transferred from the source domain to the target domain and remain a relative well performance. However, one can observe from the bottom figure that the features from Stage I and Stage II do not end up well separable, since only the marginal distribution $P(x_T)$ is aligned to the distribution $P(x_S)$ and the information that wheels are intact at the beginning is wasted, as shown in the middle subfigure.

The change in L_D , $L_{D'}$ and L_C during the 20 epochs of SADA are presented in Fig. 13. Similar to what happens in SADA, L_C remains at a low level. Regarding discriminator loss, on one hand, L_D does not converge at a level as high as that in the USDA case. On the other hand, $L_{D'}$ exhibits a significant improvement, from 0.3 to around 0.65, indicating that the features from datasets Z_T' and Z_S' are gradually mixed up during the SADA.

To more concretely figure out what is happening, one can turn to Fig. 14. As shown in the first row and second row, in addition to aligning the marginal distribution $P(x_T)$ and $P(x_S)$, SADA also leverages the Stage I features from the source domain. The distribution of them $P(x_T')$ is gradually close to its counterpart in the source domain $P(x_S|y_S = 0)$, which does not happen in Fig. 12. Note that in this process, the features of other unknown samples are also better diffuse. Consequently, as shown in Fig. 13(c), they end up being better separable, and the C suitable for source features is now applicable to them too. In this way, the effectiveness of SADA can be intuitively shown.

A similar phenomenon can be observed when the preliminary trained model is adapted for Wheel 2. To be concise, for Wheel 2, we just present visualization results at the beginning (epoch 20) and the end (epoch 40) of the SADA or UADA process in Fig. 15 and Fig. 16. Similarly, after UADA, the features in the source domain can well be undisguisable with the features in the target domain, but features from different stages do not diffuse well. In contrast, if SADA is applied, the conditional distribution can also be well aligned. As a result, the classifier can be well transferred.

4.3. Superiority of adversarial strategy: SADA vs DDC

As shown in Fig. 8 and Fig. 9, the SADA perform significantly better than DDC. The suggested reason is that a fixed metric (i.e. MMD) usually fails to capture the subtle discrepancy between source and target domains because mean embedding matching is sensitive to the Kernel choices [22].

To prove this hypothesis and provide a more comprehensive comparison, four DDC schemes with different settings are applied. In

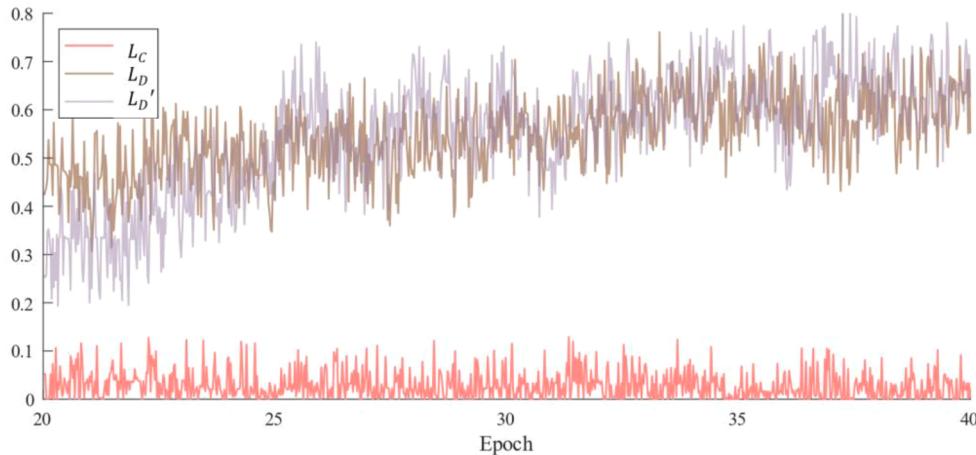


Fig. 13. Classifier loss and discriminator loss during SADA for Wheel 1.

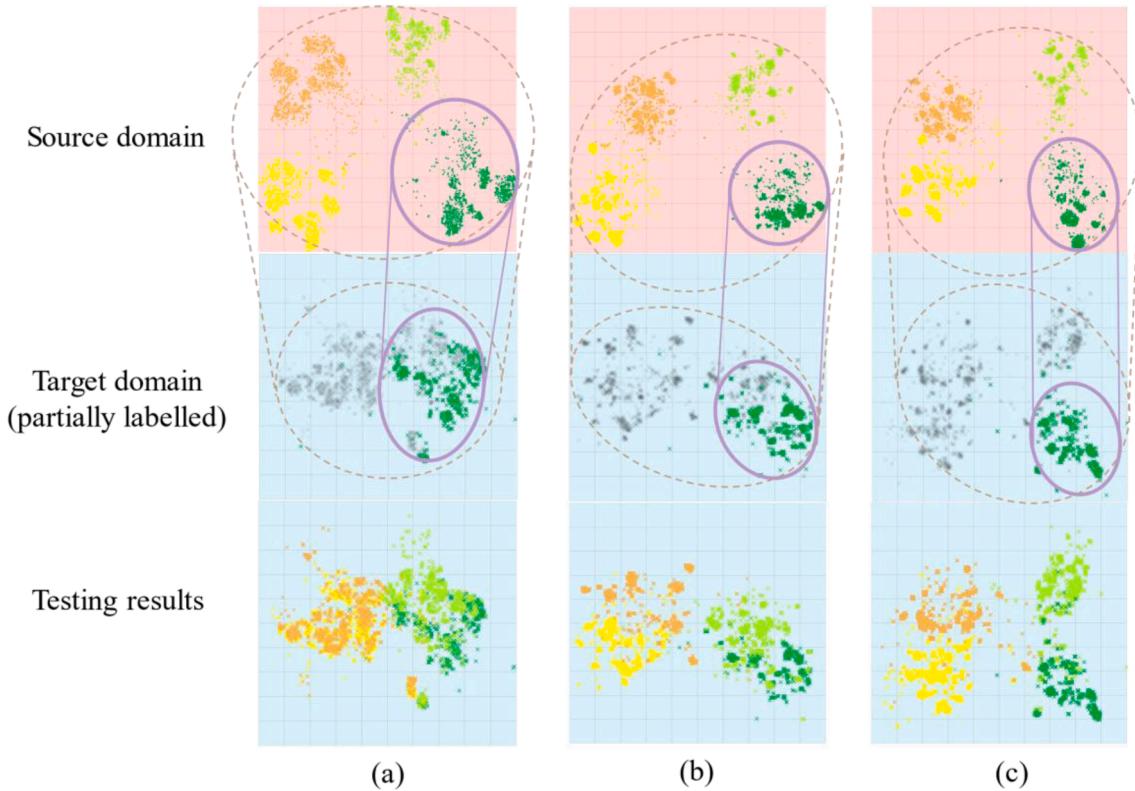


Fig. 14. Evolvement of features for Wheel 1 during SADA: (a) epoch 20, immediately after preliminary training; (b) epoch 30; (c) epoch 40.

addition to the DDC using MMD with RBF kernel ($\sigma = 1$) in Baseline 3, RBF kernel ($\sigma = 0.5$), linear kernel, and multiple kernels (MK) [56] are also used to measure L_D and L_D' based on features generated by F . As shown in Fig. 17, these variants do enable the DDC approach to perform more nearly as good as the SADA approach. However, even the strongest MK scheme cannot achieve F1 scores as high as SADA.

The results imply that the adversarial strategy enables the discriminator D , as a universal approximator [57], to learn to implicitly encode a metric that is more domain-discriminative than MK MMD. As a result, the feature extractor F , as the other role of this two-player adversarial game, learns to better mix up the features and thus enable the well-trained classifier C transferable between different operational conditions.

It should be noted that in the real practice of DA, the labels of target data are not available, unlike in research, making this kind of Kernel selection impossible. In comparison, the architecture of D in the SADA scheme is determined in a relatively arbitrary without model selection, as mentioned in Section 2.3. Therefore, the latter scheme seems to be more reliable and robust in real practice.

4.4. Adapting the model to a similar elevation

As mentioned in Section 3.2, Setting HS → MM represents a large variation of operational conditions for HSTs. All the factors either directly alter the dynamic responses or indirectly influence the infrastructure design and construction. As a result, even when wheel 1 is completely intact, the distribution of monitoring data exhibits a significant shift. Fig. 18(a) illustrates this shift using the 2D visualization method as in Section 4.2. Each pink dot in this figure represents the feature of one data segment collected from the HS section, while each blue dot represents that from the MM section. Although all the presented data belongs to the same label (Stage I), there is a significant shift caused by the change of operational conditions. In comparison, Setting HS → ST represents a small operational condition change, and the data segments from ST seem to still fall in the same distribution as those from HS. A similar pattern of data shift can be found for data in Stages II, III and IV, but they are not presented due to the space limitation.

Therefore, applying the model learned from HS exhibits a smaller performance decay on the data from ST, as shown in Fig. 18(b). The F1 score falls from 98.9% to 89.0%. This fall is much smaller than that on MM (from 98.9% to 74.8%). When SADA was applied to adapt the model from HS to ST, the performance on ST can be improved to 92.1%.

The above discussion is for Wheel 1. An investigation was conducted for Wheel 2, as shown in Fig. 19, and the findings are similar. The data segments from the ST section are closer to those from the HS section. Thus, the performance decay is not so significant and can be compromised by SADA.

What Fig. 18(a) and Fig. 19(a) present are the results after dimension reduction. As presented in Table 3, each feature extracted

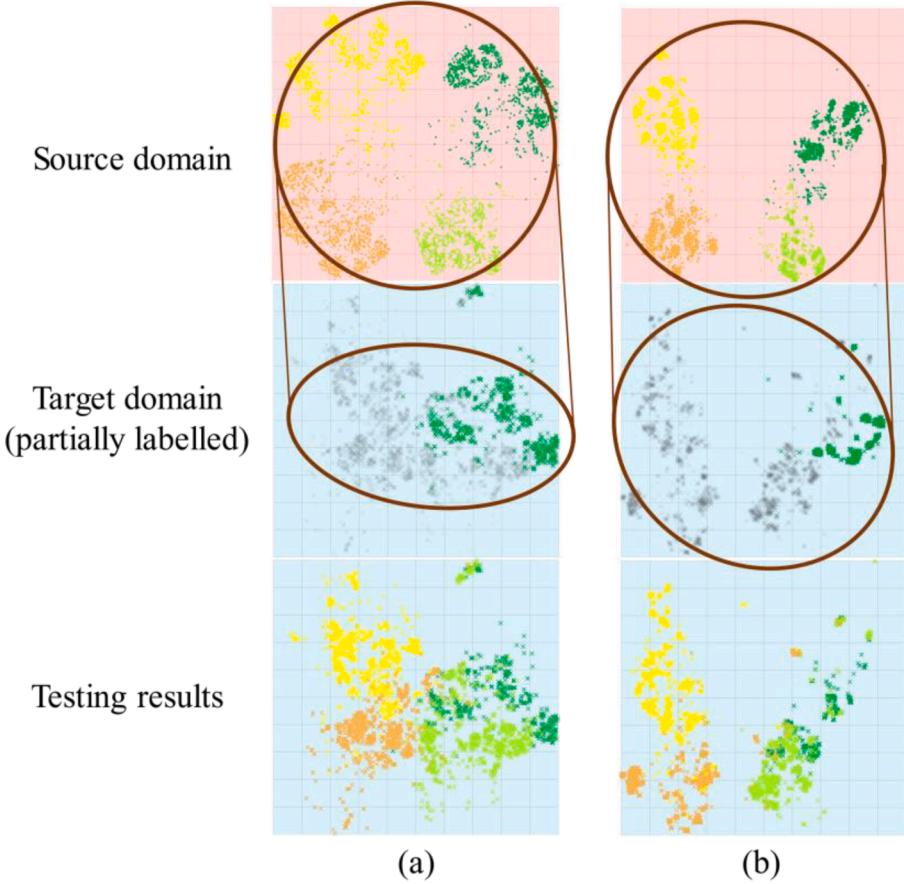


Fig. 15. Evolvement of features for Wheel 2 during UADA: (a) epoch 20, immediately after preliminary training; (b) epoch 40.

from a vibration data segment has 512 elements. To more directly demonstrate the data distribution shift, we also randomly picked up 5 elements and presented their distribution when the trained was operated on rail sections MM, HS and SS. The results for Wheel 1 are presented in Fig. 20, and those for Wheel 2 are in Fig. 21. Take the first line of Fig. 20 for example. When the HST is running on the HS section, this feature element, which is extracted from axle box vibration data of Wheel 1, mainly appears between -1 and 6 . When the HST is running on ST, the distribution of this feature element is basically the same. By contrast, when the HST was running on MM, the feature element mainly distributes between -6 and 3 , which exhibits a significant distribution shift. Similar patterns can be found in other rows and Fig. 21.

4.5. Adapting the model to a new wheel

In the research presented in previous sections, we focused on one wheel at a time. It is emphasized that the health condition of one wheel was considered unchanged in these two days, and the corresponding data were tagged as the same labels. Therefore, for one specific wheel, the labels are consistent during the training and validation.

In real rail engineering applications, however, the monitoring system normally cannot be implemented in all coaches, and there are multiple trains. In this case, the wheel to be inferred is definitely different from the wheel on which the model is trained. For the sake of completeness and further research, this study investigates the performance of a model adapted for not only a new operational condition but also for a new wheel. This is an attempt to conduct DA across sensors [29].

Two new settings are evaluated. One is to transfer from Wheel 2 on domain HS to Wheel 1 on domain MM. The other is to transfer from Wheel 1 on domain HS to Wheel 2 on domain MM. The confusion matrices are shown in Table 10(a) and Table 10(b), respectively.

For wheel 1, the F1 score is 85.8%, while that without SADA is only 69.5%. For wheel 2, it is improved from 71.8% without SADA to 84.9% with SADA. Although performance decay unavoidably exhibits when models are adapted to cross wheels, the F1 scores remain acceptable. In addition, an interesting phenomenon can be observed. As shown in Table 10(a) (Wheel 2 \rightarrow Wheel 1), the precision loss is significant for Stage IV, compared with that in Table 8(a), while the precision for other stages remains basically the same. In contrast, in Table 10(b) (Wheel 1 \rightarrow Wheel 2), the precision has a large negative change for Stage I.

This symmetry is not occasional but due to the gap between the universal labels and the slightly different ground truths of health

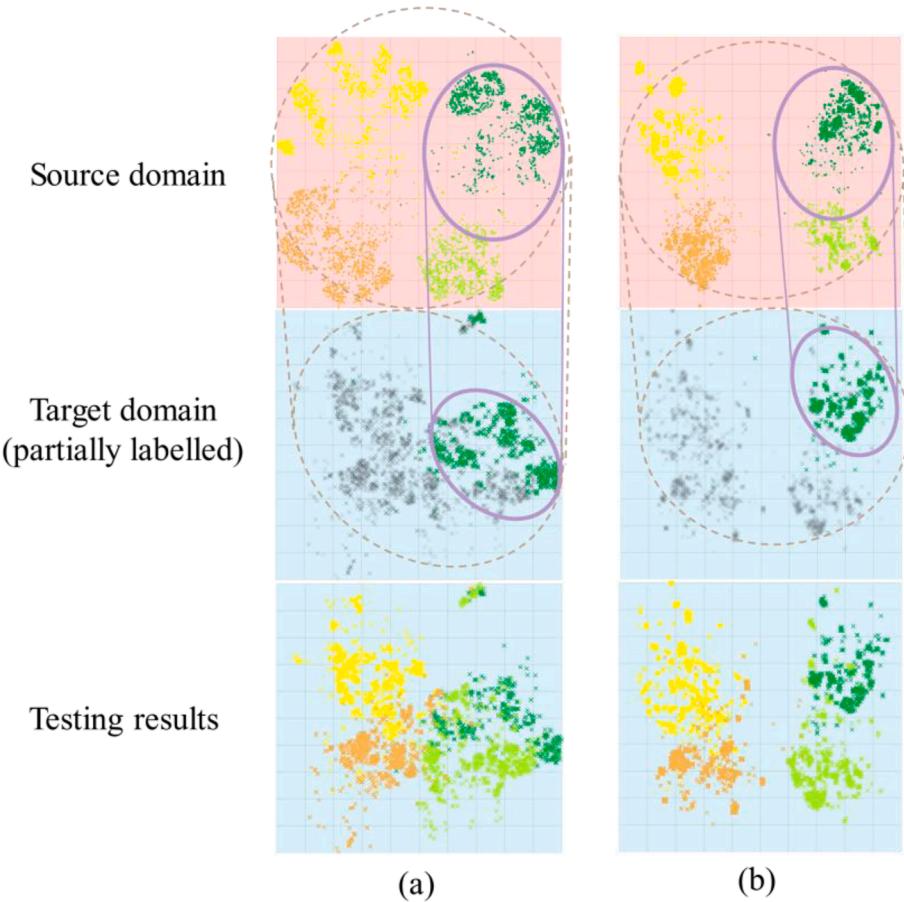


Fig. 16. Evolvement of features for Wheel 2 during SADA: (a) epoch 20, immediately after preliminary training; (b) epoch 40.

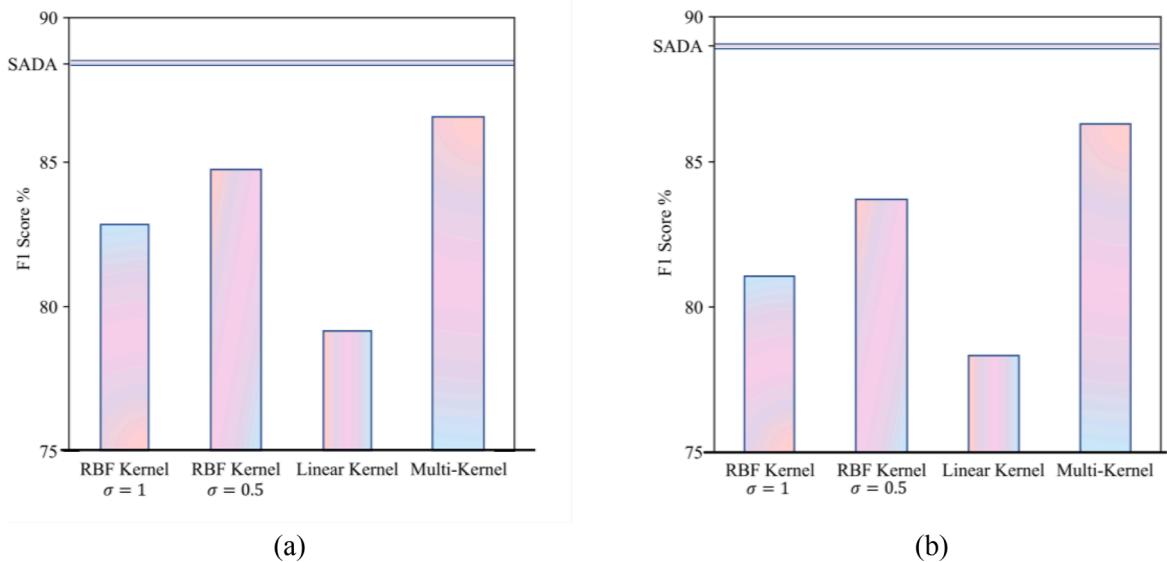


Fig. 17. SADA vs DDC with different schemes: (a) Wheel 1; (b) Wheel 2.

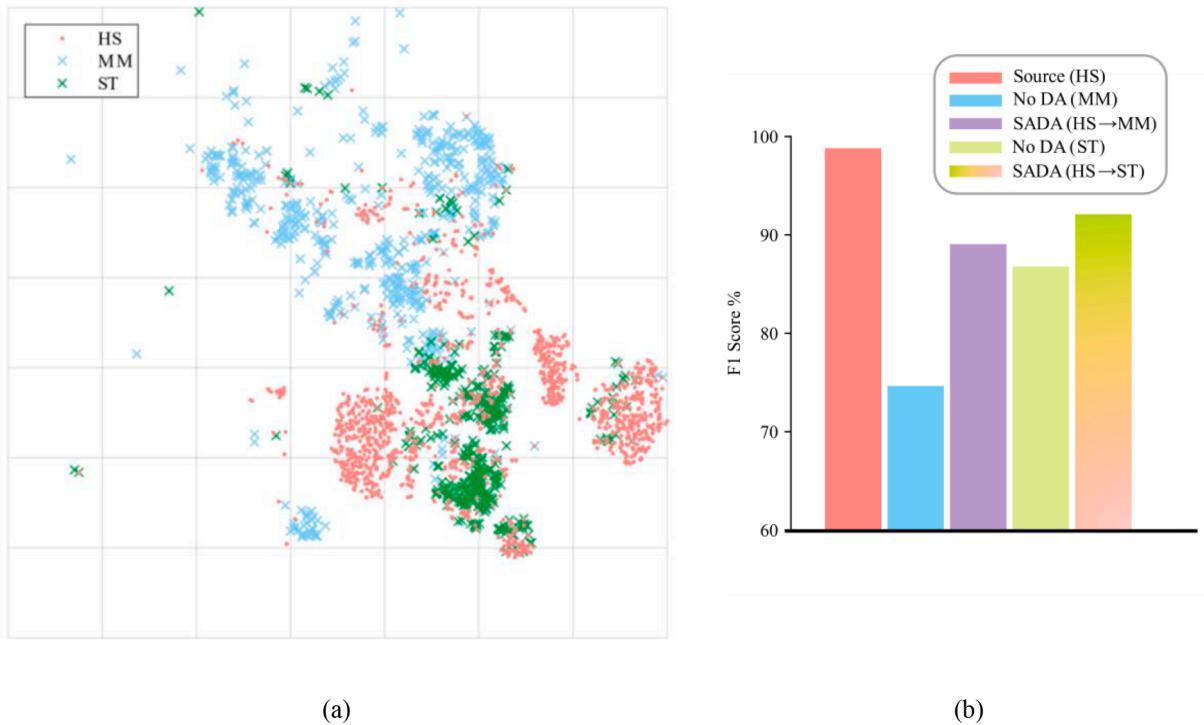


Fig. 18. Comparison between large and small operational condition variation for Wheel 1: (a) Feature visualization; (b) Performance of SADA.

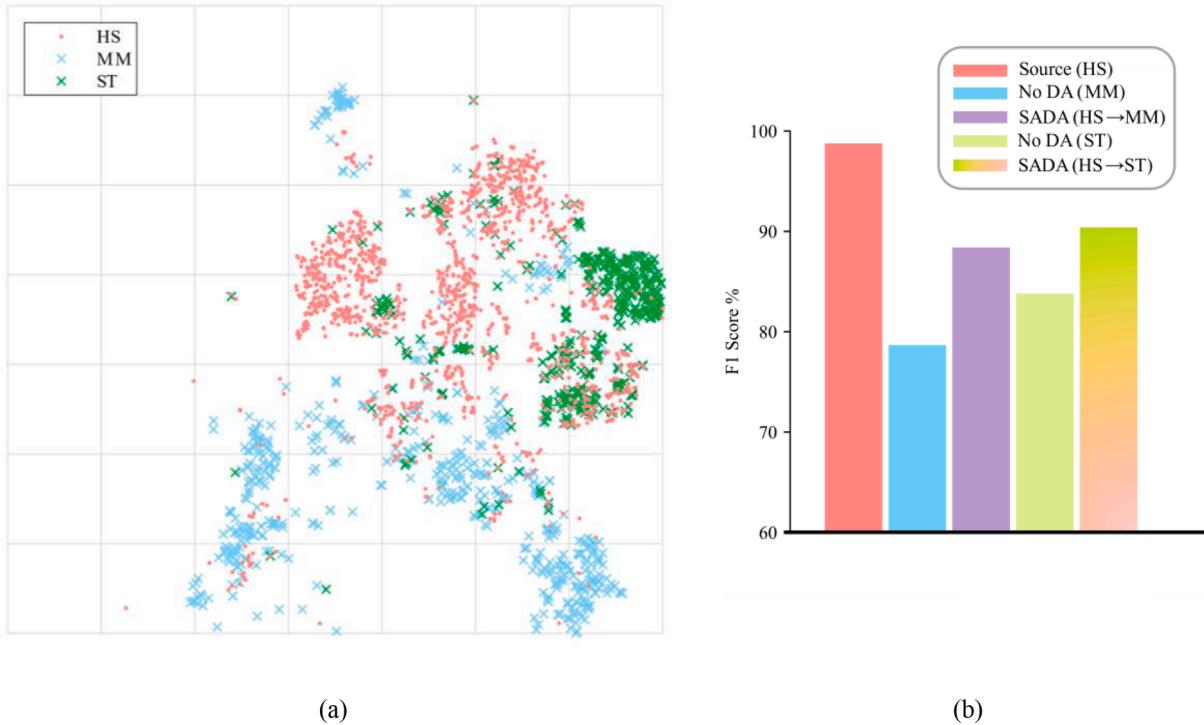


Fig. 19. Comparison between large and small operational condition variation for Wheel 2: (a) Feature visualization; (b) Performance of SADA.

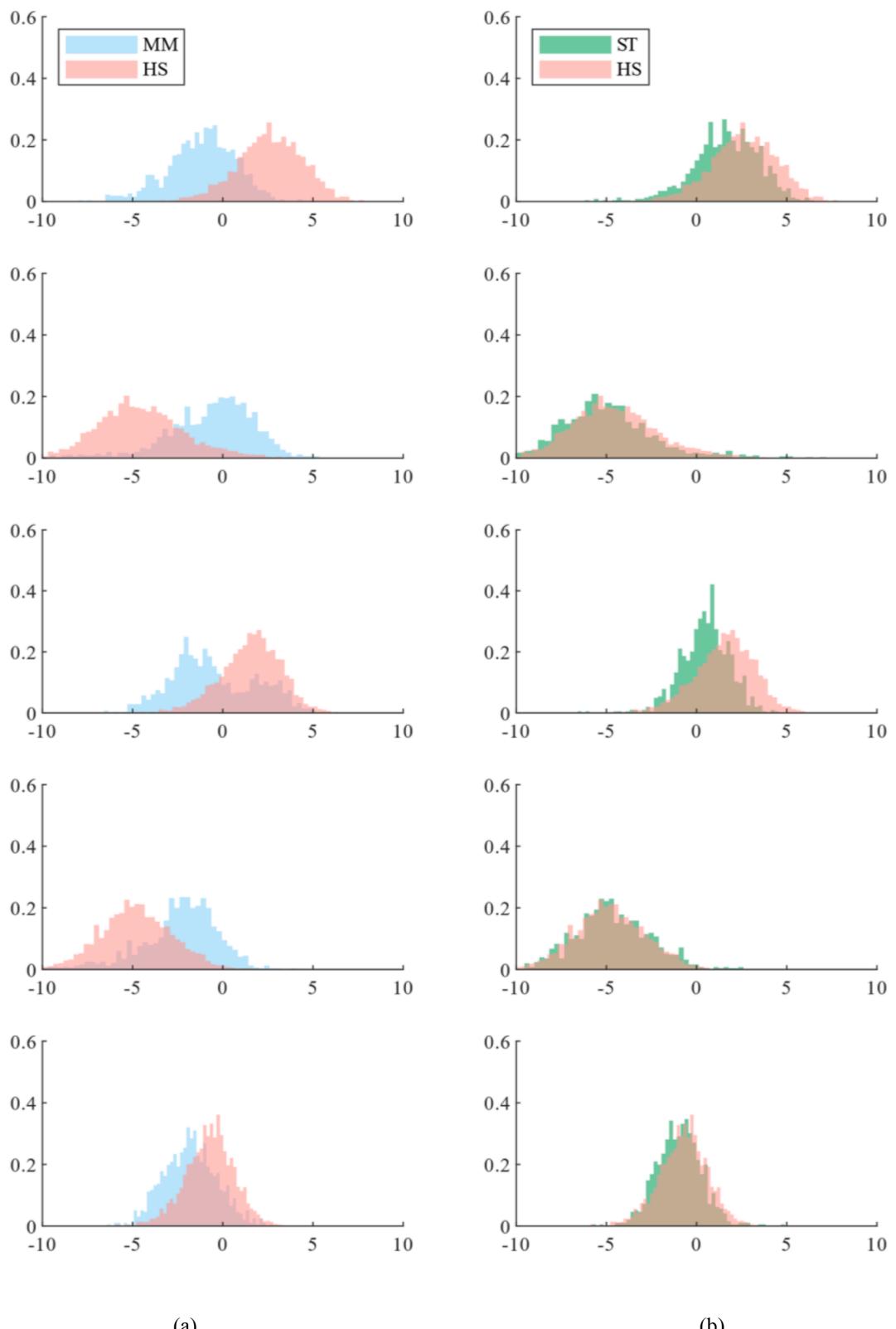


Fig. 20. Distribution comparison between elements of feature of Wheel 1: (a) HS vs MM (five different elements); and (b) HS vs ST (five different elements).

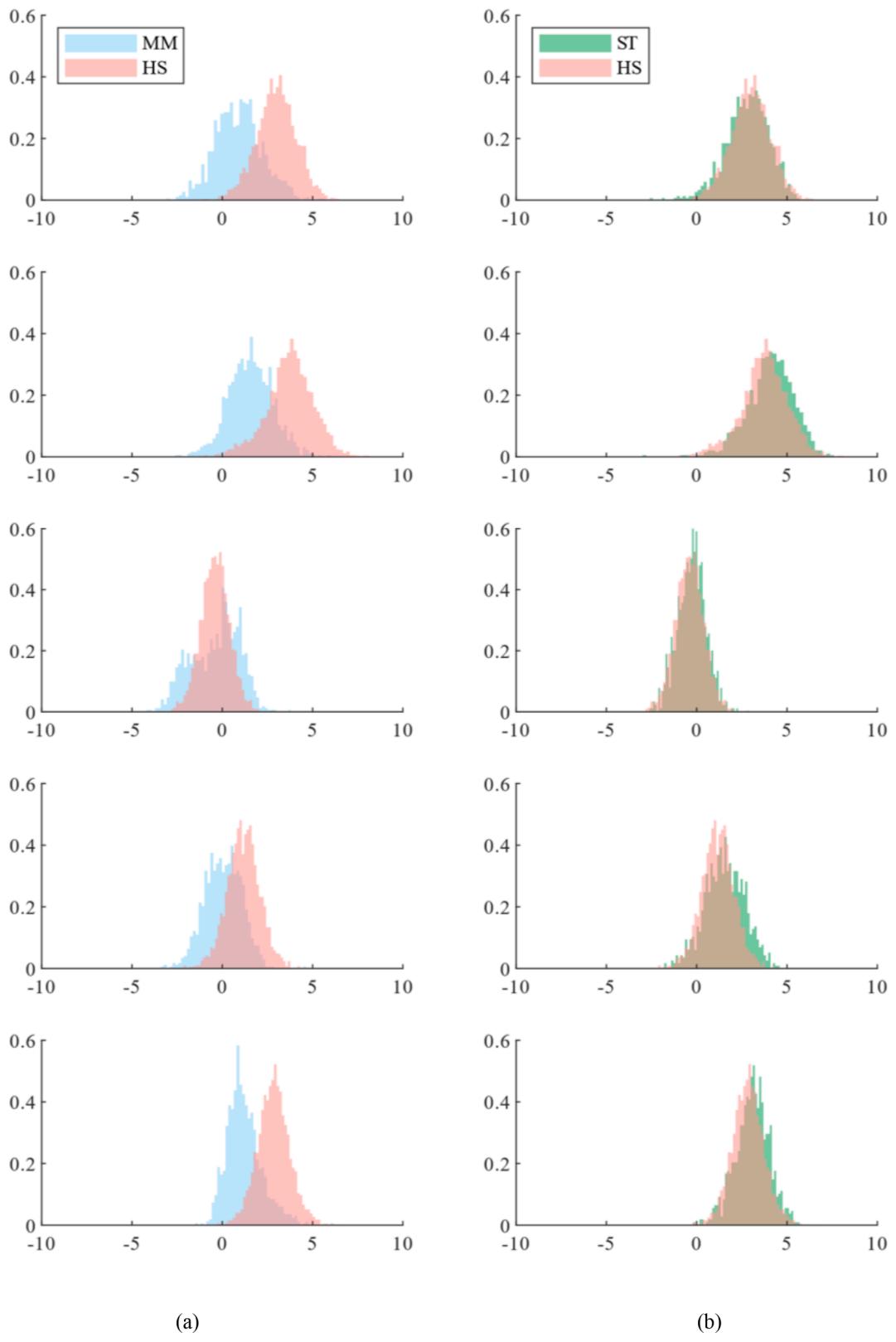


Fig. 21. Distribution comparison between elements of feature of Wheel 2: (a) HS vs MM (five different elements); and (b) HS vs ST (five different elements).

Table 10Confusion matrix by M_{SADA} (HS → MM): (a) Wheel 2 → Wheel 1; (b) Wheel 1 → Wheel 2.

(a)					
Prediction Label	Stage I	Stage II	Stage III	Stage IV	Recall
Stage I	1923	203	27	42	86.4%
Stage II	275	4058	404	19	87.8%
Stage III	20	107	3908	721	83.0%
Stage IV	39	122	323	3905	86.1%
Precision	85.2%	90.4%	83.8%	83.3%	F1: 85.8%
Precision change	+0.2%	-1.3	-3.1	-7.7%	

(b)					
Prediction Label	Stage I	Stage II	Stage III	Stage IV	Recall
Stage I	2078	56	37	24	94.7%
Stage II	693	3860	150	53	81.2%
Stage III	97	348	4150	161	87.3%
Stage IV	22	4	728	3635	82.8%
Precision	71.9%	90.4%	81.9%	93.9%	F1: 84.9%
Precision change	-9.7	-2.7	-2.1%	+0.2	

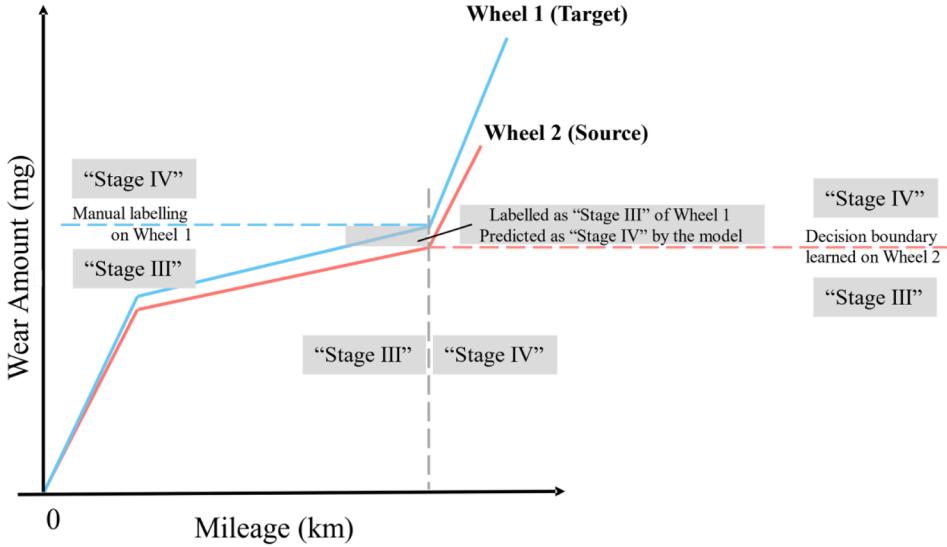


Fig. 22. Labelling deviation between Wheel 2 (source domain) and Wheel 1 (target domain).

conditions of two wheels. As mentioned in [Section 3.2](#), the labelling is based on millage and manual inspection and represent four stages. Namely, within the same two days, the ground truth health conditions of Wheel 1 and Wheel 2 were not exactly the same, although they had the same labels. In our case, it seems that the wear development of Wheel 1 is slightly faster than that of Wheel 2. As shown in [Fig. 22](#), in Setting Wheel 2 → Wheel 1, the decision boundary between “Stage III and Stage IV” is first learned on Wheel 2. Therefore, those samples from Wheel 1 exceeding this threshold (in the decision space) will be predicted as “Stage IV”. However, these samples are labelled as “Stage III” because the wear amount of Wheel 1 is slightly larger than Wheel 2 with the same millage. Therefore, these samples, labelled as “Stage III” for Wheel 1, will be predicted as “Stage I” by M_{SADA} , which explains the precision decay in [Table 10\(a\)](#). Similarly, as shown in [Fig. 23](#), some labels labelled as “Stage II” of Wheel 2 are predicted as “Stage I” by M_{SADA} , which explains the precision decay in [Table 10\(b\)](#).

To summarize, the cross-wheel scheme is possible, but the gap between labels and ground truths may be a hinder to further application. The preliminary plan will be presented in [Section 5](#). However, this gap does not influence the validation of SADA in [Section 4.1](#).

5. Conclusions and future work

In this study, a SADA approach is proposed to enable assessment of evolving wheel conditions for HSTs under different operational conditions by transferring knowledge from a model trained on monitoring data collected in one rail section (source data) to the model on data collected in another rail section (target data). The development of a two-level DA strategy over both the marginal distribution

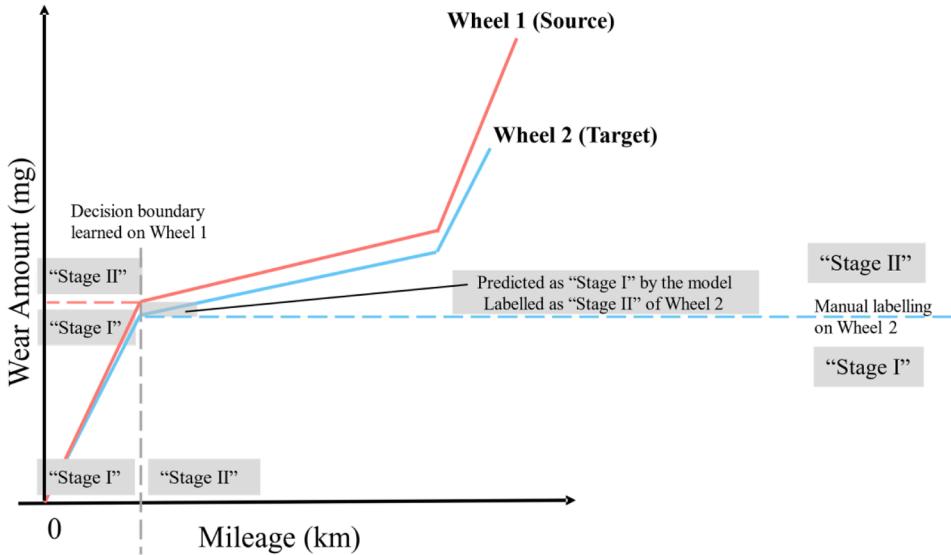


Fig. 23. Labelling deviation between Wheel 1 (source domain) and Wheel 2 (target domain).

of all data and the conditional distribution of some labelled data, together with the adversarial tactic, guarantee the thorough distribution fusion between source and target data so that the operational effects can be sufficiently eliminated. The effectiveness of approach and its superiority over other baseline methods have been well demonstrated through an in-situ monitoring case study on an operating HSR line. Specific conclusions are listed as follows.

- A significant drop can be found when the model is transferred from one rail section to another, and UADA approach can effectively compromise this drop. The F1 score has an improvement of around 5%~10%. Further, the involvement of semi-supervision can introduce another increase of around 5%. The size of source dataset also matters—the more abundant the source dataset, the more transferable the learned features.
- According to the discriminator loss, the classifier loss as well as the feature 2D representation, it is found that UADA can successfully help feature extractor to generate domain-invariant features, and SADA can further align the features from two domains condition on “intact”. As a result, the learned classifier for wheel condition assessment become transferable and generalizable.
- By showing that a carefully crafted metric of domain discrepancy (MK MMD) cannot outperform a learned discriminator without strict model selection, the superiority of adversarial training is also proved.
- The variation of operational conditions between HS and ST is smaller than that between HS and MM. Therefore, the data distribution shift is smaller, and the performance decay is less significant.
- Cross-wheel SADA is feasible, but the performance is influenced by the gap between universal labels and slightly different ground truths of different wheels.

This study shows the feasibility to assess the health condition of the same wheel even if it works under varying operational conditions. It is expected that this strategy can be generalized for the monitoring of other critical components of HST running in any rail section, given only the healthy data in this section and a well-studied case in another rail section. Nowadays, there have been 4 east-west and 4 north-south HSR lines in China, making the operational conditions of HSTs highly various. Thus, the exhibited generalizability and flexibility are valuable. Accordingly, some limitations of this study motive the following further work.

- For further validation, the proposed approach will be applied to the evaluation of another component of HST running on another rail line.
- Our approach will be further refined to enable more precise and flexible knowledge transfer from one wheel to another. As mentioned in Section 4.5, the wheel condition labelling based on mileage and manual inspection may induce skewness of prediction precision when the model is transferred to a new wheel. For this challenge, one can take both wheels as source domains by multi-source DA [58,59] so that the slight deviation of labelling from “ground truth” can be cancelled out.
- In this study, the UADA and SADA in the target domain are possible only when all the data are available, including those with wear. This hinders the wheel condition assessment conducted in real-time. In the future, we will explore DA in a progressive manner [60] so that a train can be monitored and assessed at the very beginning.
- The influence of model architecture, including feature extractor F , classifier C as well as discriminator D , are not discussed as it is not the focus of this study, and detailed discussion may lead to confusion. This exploration will be conducted in future work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was funded by a grant (RIF) from the Research Grants Council of the Hong Kong Special Administrative Region, China, grant number R5020-18. This research was also funded by the grants from the Ministry of Science and Technology of China and the Innovation and Technology Commission of Hong Kong SAR Government to the Hong Kong Branch of Chinese National Rail Transit Electrification and Automation Engineering Technology Research Center, grant number K-BBY1.

References

- [1] P. Gullers, P. Dreik, J.C.O. Nielsen, A. Ekberg, L. Andersson, Track condition analyser: identification of rail rolling surface defects, likely to generate fatigue damage in wheels, using instrumented wheelset measurements, *Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit.* 225 (2011) 1–13.
- [2] B. Stratman, Y. Liu, S. Mahadevan, Structural health monitoring of railroad wheels using wheel impact load detectors, *J. Fail. Anal. Prev.* 7 (3) (2007) 218–225.
- [3] D. Milković, G. Simić, Ž. Jakovljević, J. Tanasković, V. Lučanin, Wayside system for wheel–rail contact forces measurements, *Measurement* 46 (9) (2013) 3308–3318.
- [4] W. Zhai, P. Liu, J. Lin, K. Wang, Experimental investigation on vibration behavior of a CRH train at speed of 350 km/h, *Int. J. Rail Transp.* 3 (2015) 1–16.
- [5] T. Vanhoenacker, M. Laeremans, E. De Donder, Low-cost on-line wheelset condition monitoring, *Rail Eng. Int.* (2012).
- [6] N. Bosso, A. Gugliotta, N. Zampieri, Wheel flat detection algorithm for onboard diagnostic, *Measurement* 123 (2018) 193–202.
- [7] A. Bracciali, G. Cascini, Detection of corrugation and wheelflats of railway wheels using energy and cepstrum analysis of rail acceleration, *Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit.* 211 (2) (1997) 109–116.
- [8] S. Jia, M. Dhanasekar, Detection of rail wheel flats using wavelet approaches, *Struct. Heal. Monit.* 6 (2) (2007) 121–131.
- [9] B. Liang, S.D. Iwnicki, Y. Zhao, D. Crosbee, Railway wheel-flat and rail surface defect modelling and analysis by time–frequency techniques, *Veh. Syst. Dyn.* 51 (9) (2013) 1403–1421.
- [10] Z.J. Li, L. Wei, H.Y. Dai, J. Zeng, Y.J. Wang, Identification method of wheel flat based on Hilbert-Huang transform, *Jiaotong Yunshu Gongcheng Xuebao.* 12 (2012) 33–41.
- [11] Y. Li, M.J. Zuo, J. Lin, J. Liu, Fault detection method for railway wheel flat using an adaptive multiscale morphological filter, *Mech. Syst. Signal Process.* 84 (2017) 642–658.
- [12] L.H. Zhang, Y.Q. Ni, S.K. Lai, S. Wang, A Novel Machine Learning Technique for Online Health Monitoring of High-speed Trains, in: *Proc. 2nd Int. Work. Struct. Heal. Monit. Railw. Syst.*, Qingdao, 2018.
- [13] L.H. Zhang, Y.W. Wang, Y.Q. Ni, S.K. Lai, Online condition assessment of high-speed trains based on Bayesian forecasting approach and time series analysis, *Smart Struct. Syst.* 21 (2018) 705–713.
- [14] H. Wan, Y.Q. Ni, Binary segmentation for structural condition classification using structural health monitoring data, *J. Aerosp. Eng.* 32 (2019) 04018124.
- [15] G. Krummenacher, C.S. Ong, S. Koller, S. Kobayashi, J.M. Buhmann, Wheel defect detection with machine learning, *IEEE Trans. Intell. Transp. Syst.* 19 (4) (2018) 1176–1187.
- [16] Y. Bai, J. Yang, J. Wang, Y. Zhao, Q. Li, Image representation of vibration signals and its application in intelligent compound fault diagnosis in railway vehicle wheelset-axlebox assemblies, *Mech. Syst. Signal Process.* 152 (2021) 107421.
- [17] Y. Bai, J. Yang, J. Wang, Q. Li, Intelligent diagnosis for railway wheel flat using frequency-domain Gramian angular field and transfer learning network, *IEEE Access* 8 (2020) 105118–105126.
- [18] F. Charles R., W. Keith, Supervised Learning – Classification and Regression, in: *Struct. Heal. Monit. A Mach. Learn. Perspect.*, 2013: p. 398.
- [19] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: *Proc. Int. Conf. Mach. Learn.*, Lille, France, 2015: pp. 1180–1189.
- [20] K. Weiss, T.M. Khoshgoftaar, D. Wang, A survey of transfer learning, *J. Big Data.* 3 (2016) 1–40.
- [21] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, T. Darrell, Deep Domain Confusion: Maximizing for Domain Invariance, (2014). <http://arxiv.org/abs/1412.3474>.
- [22] M. Long, Y. Cao, J. Wang, M.I.I. Jordan, Learning Transferable Features with Deep Adaptation Networks, in: *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, 2015: pp. 97–105.
- [23] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, U. Dogan, M. Kloft, F. Orabona, T. Tommasi, Domain-Adversarial Training of Neural Networks, *The journal of machine learning research.* 17(2016) 2096–2030.
- [24] V.M. Patel, R. Gopalan, R. Li, R. Chellappa, Visual domain adaptation: a survey of recent advances, *IEEE Signal Process. Mag.* 32 (3) (2015) 53–69.
- [25] J. Singh, M. Azamfar, A. Ainaipure, J. Lee, Deep learning-based cross-domain adaptation for gearbox fault diagnosis under variable speed conditions, *Meas. Sci. Technol.* 31 (5) (2020) 055601.
- [26] X. Yu, Z. Zhao, X. Zhang, C. Sun, B. Gong, R. Yan, X. Chen, Conditional Adversarial Domain Adaptation with Discrimination Embedding for Locomotive Fault Diagnosis, *IEEE Trans. Instrum. Meas.* 9456 (2020) 1–1.
- [27] J. Jiao, M. Zhao, J. Lin, C. Ding, Classifier inconsistency-based domain adaptation network for partial transfer intelligent diagnosis, *IEEE Trans. Ind. Informatics.* 16 (9) (2020) 5965–5974.
- [28] P. Lei, C. Shen, D. Wang, L. Chen, Z. Zhou, Z. Zhu, A new transferable bearing fault diagnosis method with adaptive manifold probability distribution under different working conditions, *Measurement* 173 (2021), 108565.
- [29] X. Li, W. Zhang, N.-X. Xu, Q. Ding, Deep learning-based machinery fault diagnostics with domain adaptation across sensors at different places, *IEEE Trans. Ind. Electron.* 67 (8) (2020) 6785–6794.
- [30] L. Guo, Y. Lei, S. Xing, T. Yan, N. Li, Deep convolutional transfer learning network: a new method for intelligent fault diagnosis of machines with unlabeled data, *IEEE Trans. Ind. Electron.* 66 (9) (2019) 7316–7325.
- [31] B. Yang, Y. Lei, F. Jia, S. Xing, An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings, *Mech. Syst. Signal Process.* 122 (2019) 692–706.
- [32] P. Gardner, X. Liu, K. Worden, On the application of domain adaptation in structural health monitoring, *Mech. Syst. Signal Process.* 138 (2020) 106550, <https://doi.org/10.1016/j.ymssp.2019.106550>.
- [33] L.A. Bull, P.A. Gardner, N. Derviliis, E. Papatheou, M. Haywood-Alexander, R.S. Mills, K. Worden, On the transfer of damage detectors between structures: an experimental case study, *J. Sound Vib.* 501 (2021) 116072, <https://doi.org/10.1016/j.jsv.2021.116072>.
- [34] S.X. Chen, L. Zhou, Y.Q. Ni, X.Z. Liu, An acoustic-homologous transfer learning approach for AE-based rail condition evaluation, *Struct. Heal. Monit.* (2020).
- [35] Z. Chen, Y. Bao, H. Li, B.F. Spencer, LQD-RKHS-based distribution-to-distribution regression methodology for restoring the probability distributions of missing SHM data, *Mech. Syst. Signal Process.* 121 (2019) 655–674.
- [36] S.J. Pan, Q. Yang, A survey on transfer learning, *IEEE Trans. Knowl. Data Eng.* 22 (10) (2010) 1345–1359.
- [37] Y. Yao, Y. Zhang, X. Li, Y. Ye, Discriminative distribution alignment: a unified framework for heterogeneous domain adaptation, *Pattern Recognit.* 101 (2020), 107165.

- [38] G. Wilson, D.J. Cook, A survey of unsupervised deep domain adaptation, *ACM Trans. Intell. Syst. Technol.* 11 (5) (2020) 1–46.
- [39] N. Ahmed, T. Natarajan, K.R. Rao, Discrete cosine transform, *IEEE Trans. Comput. C-23* (1) (1974) 90–93.
- [40] S.K. Kumar, On weight initialization in deep neural networks (2017), <https://arxiv.org/abs/1704.08863>.
- [41] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, Analysis of representations for domain adaptation, *Adv. Neural Inf. Process. Syst.* 19 (2007) 137.
- [42] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J.W. Vaughan, A theory of learning from different domains, *Mach. Learn.* 79 (1-2) (2010) 151–175.
- [43] G. Fan, J. Li, H. Hao, Y. Xin, Data driven structural dynamic response reconstruction using segment based generative adversarial networks, *Eng. Struct.* 234 (2021), 111970.
- [44] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, in: Proc. 3rd Int. Conf. Learn. Repr., San Diego, CA, 2015.
- [45] V. Nair, G.E. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines, in: Proc. 27th Int. Conf. Mach. Learn., 2010: pp. 807–814.
- [46] C. Tomberger, P. Dietmaier, W. Sextro, K. Six, Friction in wheel–rail contact: a model comprising interfacial fluids, surface roughness and temperature, *Wear* 271 (2011) 2–12.
- [47] J.P. Srivastava, P.K. Sarkar, V. Ranjan, Effects of thermal load on wheel–rail contacts: A review, *J. Therm. Stress.* 39 (11) (2016) 1389–1418.
- [48] L.B. Shi, L. Ma, J. Guo, Q.Y. Liu, Z.R. Zhou, W.J. Wang, Influence of low temperature environment on the adhesion characteristics of wheel–rail contact, *Tribol. Int.* 127 (2018) 59–68.
- [49] D. Liu, Q. Wang, M.u. Zhong, Z. Lu, J. Wang, T. Wang, S. Lv, Effect of wind speed variation on the dynamics of a high-speed train, *Veh. Syst. Dyn.* 57 (2) (2019) 247–268.
- [50] D. Liu, T. Wang, X. Liang, S. Meng, M. Zhong, Z. Lu, High-speed train overturning safety under varying wind speed conditions, *J. Wind Eng. Ind. Aerodyn.* 198 (2020), 104111.
- [51] C. Wang, Z. Zhang, H. Zhang, Q. Wu, B.o. Zhang, Y. Tang, Seasonal deformation features on Qinghai-Tibet railway observed using time-series InSAR technique with high-resolution TerraSAR-X images, *Remote. Sens. Lett.* 8 (1) (2017) 1–10.
- [52] F. Braghin, R. Lewis, R.S. Dwyer-Joyce, S. Bruni, A mathematical model to predict railway wheel profile evolution due to wear, *Wear* 261 (11-12) (2006) 1253–1264.
- [53] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, in: Proc. Adv. Neural Inf. Process. Syst., 2014: pp. 3320–3328.
- [54] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, in: Proc. 31st Int. Conf. Mach. Learn., 2014: pp. 647–655.
- [55] L. Van Der Maaten, G. Hinton, Visualizing data using t-SNE, *J. Mach. Learn. Res.* 9 (2008) 2579–2605.
- [56] A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, B.K. Sriperumbudur, Optimal kernel choice for large-scale two-sample tests, in: Adv. Neural Inf. Process. Syst., Citeseer, 2012: pp. 1205–1213.
- [57] B.C. Csáji, Approximation with artificial neural networks, *Fac. Sci. Etvs Lornd Univ. Hungary.* 24 (2001) 7.
- [58] H. Zhao, S. Zhang, G. Wu, J.M.F. Moura, J.P. Costeira, G.J. Gordon, Adversarial multiple source domain adaptation, *Adv. Neural Inf. Process. Syst.* 31 (2018) 8559–8570.
- [59] H. Zheng, R. Wang, Y. Yang, Y. Li, M. Xu, Intelligent fault identification based on multisource domain generalization towards actual diagnosis scenario, *IEEE Trans. Ind. Electron.* 67 (2) (2020) 1293–1304.
- [60] J. Li, K. Lu, Z. Huang, L. Zhu, H.T. Shen, Heterogeneous domain adaptation through progressive alignment, *IEEE Trans. Neural Networks Learn. Syst.* 30 (5) (2019) 1381–1391.