A Global Adversarial and Local Contrastive Transfer Learning Approach for Remaining Useful Life Prediction

Zengwei Yuan, Shichang Du, Rui Wang, Xin Wang

***Abstract*—The Transfer learning method can effectively mitigate the reliance on an extensive amount of target domain data for the construction of remaining useful life (RUL) prediction models. While transfer learning has achieved remarkable achievements in numerous real-world applications, traditional methods often focus on aligning global domain features and extracting domain-invariant features, overlooking local and channel-specific characteristics. This limitation can result in suboptimal RUL predictions, as critical degradation-related features may be ignored. To address these challenges, this article proposes a global adversarial and local contrastive (GALC) transfer learning approach for predicting RUL in multiple domains. The proposed method employs adversarial learning to capture global domain-invariant features while incorporating channel-level and temporal-level contrastive modules to preserve local temporal patterns and channel uniqueness. This approach enables us to achieve domain-invariant feature learning while preserving channel distinctiveness across diverse domains, and its effectiveness has been validated using real-world datasets.**

***Index Terms*—Adversarial learning, contrastive learning, remaining useful life, transfer learning.**

I. Introduction

PROGNOSTICS and health management (PHM) is an innovative technology that plays a crucial role in achieving predictive maintenance in various industrial systems, such as semiconductor manufacturing [1], health management of lithium batteries [2], [3], and bearing fault diagnosis [4]. An essential component in PHM technology is precisely predicting the remaining useful life (RUL) of industrial systems. The RUL prediction offers crucial information that enables enterprises to determine the remaining operational time of industrial systems and facilitate maintenance decision-making before any potential failures, which plays a pivotal role in preventing unplanned equipment downtime and reducing maintenance costs [5].

In recent years, significant progress has been made in RUL prediction, with data-driven techniques, particularly deep learning approaches, gaining significant attention. These methods excel in leveraging deep network architectures to automatically extract meaningful degradation features from raw sensor data. However, RUL prediction often faces challenges arising from diverse operational states in industrial settings, which lead to distinct data distributions. For example, varying speed and load forces in bearing operations, different ion etching conditions during etching processes, or diverse lithium battery types and temperature conditions during failure processes all contribute to distributional differences. Moreover, the amount of labeled data across these operational states is often imbalanced, further complicating model performance under new conditions. As a result, previously trained models may fail to generalize effectively to changing working conditions. Training models for each scenario individually not only incurs substantial computational costs but also demands significant human resources.

To overcome this challenge, transfer learning methods are extensively employed in the development of RUL prediction models across multiple work scenarios. The objective of transfer learning is to enhance the performance of models in target domains by leveraging knowledge from distinct yet related source domains [6]. Various approaches, including fine-tuning [1], domain adaptation (DA) methods [7], and adversarial learning [8], have been proposed to minimize domain discrepancies. Fine-tuning is one of the most commonly used techniques in transfer learning, involving the optimization of pre-trained models on new tasks. However, its success heavily depends on the selection of appropriate initial weights. DA methods have proven to be valuable in enhancing the generalization performance of models and mitigating the need for labeled data in RUL prediction. Notwithstanding its usefulness, DA methods are susceptible to discrepancies between domains. The adversarial transfer learning method proves to be highly efficient in addressing the challenge of limited data availability, and it notably diminishes the training time and computational resources demanded by the model in the target domain. Nevertheless, sole reliance on adversarial transfer learning approaches for achieving model transfer raises concerns regarding excessive dependency on the quality of source domain data.

Despite the recent advancements in transfer learning, as highlighted above, there remain two significant challenges in applying transfer learning for RUL prediction in industrial settings: 1) Insufficient migration of local critical information. Localized critical information under varying operating con-

This work was supported in part by the National Natural Science Foundation of China under Grant No. 72101065 and Grant No. 92467101, and in part by the Shenzhen Science and Technology Program under Grant No. RCBS20221008093124063. (*Corresponding author: Rui Wang, e-mail: r.wang@hit.edu.cn*)

Zengwei Yuan, Rui Wang and Xin Wang are with the Department of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen, China.

Shichang Du is with the Department of Industrial Engineering and Management, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China.

ditions is crucial, as it often encapsulates key details about specific stages of the failure process. However, conventional methods fail to adequately capture and transfer this information. 2) Neglect of domain-specific information in the target domain. In industrial scenarios, RUL predictions are highly sensitive to the characteristics of specific sensor channels, which vary depending on working conditions and specific fault types. The inability to account for this domain-specific information compromises the accuracy and reliability of the prediction model.

To address the challenges of transferring RUL prediction models across diverse operational states, we propose a novel Global Adversarial and Local Contrastive (GALC) transfer learning approach. This method integrates adversarial and contrastive learning at both global and local levels, effectively utilizing unlabeled target domain data to enhance model performance. Specifically, the approach is divided into local and global parts. In the global part, adversarial learning is applied to align feature representations between the target and source domains, ensuring consistency in the overall degradation process. In the local part, contrastive learning is incorporated to capture localized features from two perspectives: Temporal-Level Contrastive Module, which is a kernel-based method, is used to align temporal features at specific time steps, while Channel-Level Contrastive Module preserves the distinctiveness of individual sensor channels within the target domain. The GALC approach achieves robust domain adaptation while retaining target-specific information, thereby improving RUL prediction accuracy across varying operational states. The contributions of this paper can be summarized as follows:

1) We propose a global adversarial and local contrastive (GALC) transfer learning model, combining global and local perspectives with adversarial and contrastive learning, to enhance RUL prediction across diverse operating conditions.
2) We introduce a channel-level contrastive module to preserve domain-specific information in the target domain and a temporal-level contrastive module that incorporates position information to effectively utilize domain specific information of the target domain.
3) We conduct comparative experiments to validate the effectiveness of the proposed method, and satisfactory performance is obtained for knowledge transfer across various operating conditions.

The rest of this paper is organized as follows. In Section II, we provide an overview of related works concerning adversarial and contrastive learning methods applied to RUL prediction. Section III details the proposed global adversarial and local contrastive transfer learning approach. Section IV presents the experimental setup and analyzes the results. Finally, Section V concludes the paper, offering insights and potential directions for future research.

II. Related Works

A. Deep Learning Approaches for RUL Prediction

Researchers have made advancements in data-driven RUL prediction techniques of fault prognosis over the past few years. These methods have demonstrated outstanding performance in extracting meaningful features for RUL prediction. The data-driven RUL prediction model based on convolutional neural networks (CNN) [9], recurrent neural networks (RNN) [10], and attention-based Transformer models [11] is undergoing a gradual evolution from relying on single structures like convolutional and recurrent structures to adopting composite models (such as CNN-RNN [12]). Ma and Mao [13] introduced a convolution-based long short-term memory network for the prediction of RUL in rotating machinery. Zraibi *et al*. [14] propose a hybrid approach called CNN-long short term memory network (LSTM)-deep neural network (DNN) that c ombines CNN, LSTM, and DNN to estimate the RUL of lithium-ion batteries.

However, existing models still encounter challenges in capturing global dependencies. To address this limitation, the RUL prediction model incorporates the attention mechanism, which is inspired by Transformer. Zhang *et al*. [15] propose dual-aspect self-attention based on transformer for RUL prediction, which effectively addresses the issue of information interference between the sensor and time step aspects during prediction. Wang *et al*. [16] presented a multiscale convolutional attention network aimed at accurately forecasting the RUL of machinery. Furthermore, Liu *et al*. [17] propose an innovative feature-attention-based approach for predicting the RUL, while Chen *et al*. [18] introduce a deep learning framework based on attention mechanisms for machine's RUL prediction. The attention mechanism enhances model performance by enabling dynamic weighting of input features, allowing for improved focus on relevant information and thereby facilitating better contextual understanding in temporal tasks.

B. Adversarial Network

Adversarial networks facilitate transfer learning by minimizing distribution disparities between source and target domains. They have gained prominence in deep learning, particularly with the introduction of generative adversarial networks (GANs) [19]. In RUL prediction, adversarial networks integrate an encoder, which extracts features across domains, and a discriminator, which distinguishes these features [20]. This interaction enables the encoder to learn domain-invariant features, enhancing feature extraction and improving prediction models for unlabeled datasets. Ragab *et al*. [21], [22] propose two adversarial domain adaptation methods to incorporate target-specific information. Lu *et al*. [8] introduce a deep adversarial LSTM framework to address prediction error superposition.

These methods effectively extract domain-invariant features from both source and target domains on a global scale. However, the performance of adversarial networks heavily relies on the similarity between source and target data distributions. Temporal sensor data, influenced by varying operating conditions and time-dependent characteristics [23], often exhibits significant distributional discrepancies. Consequently, adversarial networks alone may fail to fully leverage target domain information. Moreover, equipment degradation is typically a gradual process, where local time intervals reveal subtle but

critical changes and trends. While global information provides a broad historical context, it often includes redundant data that can obscure key degradation signals. Thus, relying solely on global adversarial feature extraction is insufficient for accurate RUL prediction. To enhance transfer learning, it is crucial to integrate both global and local information from time series data, ensuring a comprehensive and effective model adaptation.

C. Contrastive Learning

Contrastive learning aims to map samples from the same category into similar embedding spaces by evaluating their pairwise similarity, while ensuring that samples from different categories are projected into more distant embedding spaces [24]. This approach has proven particularly effective for extracting meaningful feature representations from unlabeled sensor data, a critical requirement for analyzing variations in equipment or systems operating under diverse conditions. Recent advancements have further demonstrated the versatility of contrastive learning in addressing challenges within predictive maintenance and fault diagnosis. Mao et al. [25] proposed a deep transfer learning-based online remaining useful life (RUL) prediction method for rolling bearings, effectively mitigating biases introduced by shifts in working conditions. Similarly, Zhuang et al. [26] developed a cross-domain adaptation framework that integrates contrastive loss with multi-kernel maximum mean discrepancy (MK-MMD) to enhance domain generalizability. Building on these contributions, Wang et al. [27] introduced a contrastive generative replay technique tailored for continuous RUL prediction, which ensures model performance when incorporating new data.

Contrastive learning organizes data into pairs, enabling it to capture local dynamic changes in time series data, such as trends and anomalies. This capability is particularly valuable for extracting critical features while addressing common challenges, including sample imbalance and missing values. Even with sparse or incomplete data, contrastive learning enhances model transferability, thereby improving prediction robustness and reliability across various operating conditions.

III. Methodology

A. Problem Definition

To formulate our problem, $\mathcal{D}^s$ and $\mathcal{D}^t$ refer to multivariate time-series data collected under different working conditions. We have abundant labeled instances from the source domain, but we lack labeled instances from the target domain. The goal of out transfer learning method is to learn a more accurate RUL prediction model with unlabeled data under different working conditions. We denote the source domain $\mathcal{D}^s = \{(X_i^s, y_i^s)\}_{i=1}^{n_s}$, where $X_i^s \in \mathbb{R}^{M\times W}$ denotes the ith input source sample with M sensors and W time steps, $y_i^s \in \mathbb{R}$ represents the corresponding RUL value, and n_s is the total number of samples in source domain data. Similarly, the unlabeled target domain $\mathcal{D}^t = \{X_i^t\}_{i=1}^{n_t}$, where $X_i^t \in \mathbb{R}^{M\times W}$ represents the ith sample of target domain dataset, and n_t is the number of target domain samples.

B. Overall Framework

The process of this method is illustrated in Fig. 1., and can be separated into three main steps. The initial step involves preparing data for both the source and target domains. In the subsequent step, The source encoder E^s and the RUL predictor R are pre-trained using annotated data in the source domain. Then, the source and target encoders extract features from the respective domains. With the target features in hand, the target encoder E^t is updated through both the global adversarial part and the local contrastive part (including channel-level contrastive module and temporal-level contrastive module). In the final step, the trained target encoder and the trained source RUL predictor are merged to predict the RUL for the target domain. The model is separated into three main parts: (1) RUL prediction, (2) global adversarial learning, (3) local contrastive learning (including channel-level contrastive module and temporal-level contrastive module). Detailed explanations of each part will be provided in the following subsections.

C. RUL Prediction Model on Source Domain

In this section, the RUL prediction model is first proposed for supervised pretraining on source domain with annotated data. Based on our previous work [28], we propose a multi-scale Transformer network that builds on the original Transformer encoder, as illustrated in Fig. 2. Transformer first uses the positional encoder capture temporal information of sequences, with

$$p_t(2i) = \sin(t/10000^{2i/d_{model}}), \tag{1}$$

and

$$p_t(2i+1) = \cos(t/10000^{2i/d_{model}}), \tag{2}$$

where t is the time step position of entry in the sequence, d_{model} is the model dimension, and and $2i$ and $2i+1$ are the odd and even numbered dimension, respectively. There is a linear dependency between position embeddings of different samples, and this allows the model to learn the positional information of the sequence. The positional encoding is then added to the original sample according to its corresponding element.

The network uses a multi-scale feature extraction structure, which simultaneously captures both coarse and fine features at the same granularity level. Specifically, the input $X_i^s + P^s \in \mathbb{R}^{M\times W}$ is mapped to a new token through a 1D-convolution function $\mathrm{F}_{C\times k}(\cdot)$ across the temporal dimension W, yielding the embedded features at each scale, where P^s denotes the position embedding of X_i^s. The convolution kernel size is k, which depends on the scale, and C is the number of filters that remains fixed across different scales.

These features are then processed through multi-layer stacked Transformer blocks, where the attention mechanism plays a crucial role [11]. The attention process involves mapping the input to Queries (Q), Keys (K), and Values (V) using weight matrices, followed by calculating the correlation scores between Q and K. The scores are normalized and

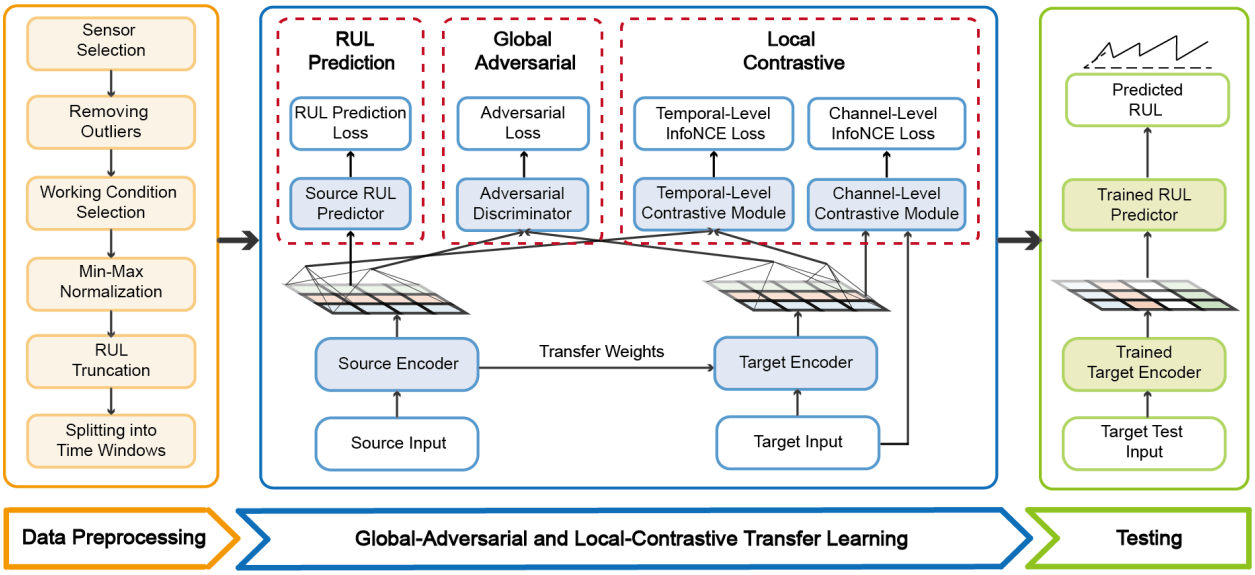
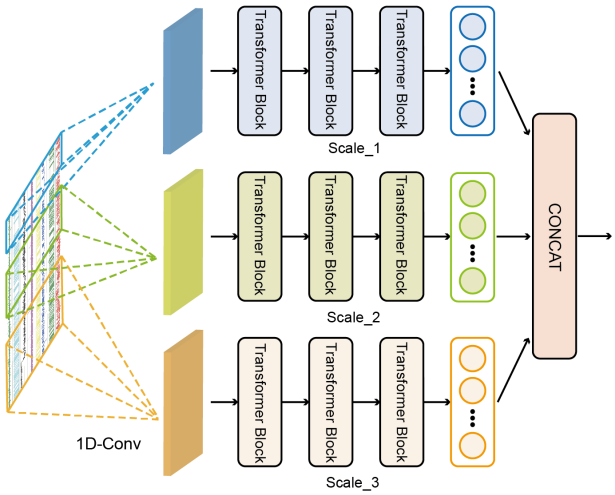
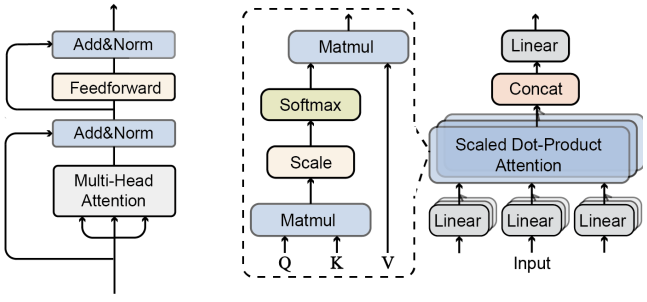


Fig. 1. Flowchart of the GALC transfer learning method. The data preprocessing stage involves sensor selection, removing outliers, working condition selection, Min-Max normalization, RUL truncation, and splitting into time windows. The Global Adversarial and Local Contrastive Transfer Learning Approach is separated into three main parts: (1) RUL prediction, (2) global adversarial learning, (3) local contrastive learning (including channel-level contrastive module and temporal-level contrastive module).



(a) Multi-scale Transformer.



(b) Transformer block. (c) Multi-head attention.

Fig. 2. Overall structure and details of multi-scale Transformer based encoder E^s and E^t.

used to produce the output Z, implemented through multi-head attention as:

$$Z = \text{Attention}(Q, K, V) = \text{Softmax}(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V, \quad (3)$$

where $\sqrt{d_k}$ is the dimension of K. Each Transformer block also includes a feedforward network, consisting of two fully-connected layers with ReLU activation in the first layer. Moreover, residual connections are incorporated into each subblock of the encoder, followed by LayerNorm, ensuring stability in the model's training process. The entire calculation process can be expressed as follows:

$$X_{out} = \text{LayerNorm}(X_{in} + \text{SubBlock}(X_{in})). \quad (4)$$

The final output feature F_i^s of the multi-scale feature extraction is obtained by concatenating the extracted features across different scales along the feature dimension, allowing for a comprehensive feature map that captures information from receptive fields of varying sizes, expressed as:

$$F_i^s = \text{Concat}(\mathbf{f}_i^{s,(1)}, \mathbf{f}_i^{s,(2)}, ..., \mathbf{f}_i^{s,(N)}), \quad F_i^s \in \mathbb{R}^{N \cdot C \times W'}, \quad (5)$$

where F_i^s represents the final output of the multi-scale feature extraction, and N denotes the number of scales, and W' denotes the length of the time series after convolution.

Given the extracted features F_i^s from the multi-scale Transformer network, The RUL predictor is a multi-layer network $R : \mathbb{R}^{N \cdot C \times W'} \rightarrow \mathbb{R}$ that maps the latent features into the corresponding RUL value. The RUL predictor R and the source encoder E^s are trained by minimizing the mean square error (MSE) loss function:

$$\mathcal{L}_{mse} = \frac{1}{n_s} \sum_{i=1}^{n_s} (\hat{y}_i^s - y_i^s)^2, \quad (6)$$

where $\hat{y}_i^s = R(E^s(X_i^s))$ is the predicted RUL value, y_i^s is the actual RUL value, and n_s is the number of samples in the source domain.

D. Global Adversarial Learning

In the global adversarial learning, a domain discriminator D is used to achieve domain-invariant features extraction between the source and target domains at a global level. Different systems or the same system under different working conditions may have similarities in failure modes. Furthermore, the complete time series data of the system's failure cycle contains all the relevant information about the failure. Therefore, adversarial methods at the global level enable the encoder and discriminator to better understand the global structure and temporal dependencies in time series data, thereby improving the encoder's ability to capture system failure modes and extract domain-invariant features.

Let E^s and R represent the Transformer-based feature extractor trained on the source domain and the predictor for RUL, respectively. Based on the aforementioned analysis,the domain discriminator network D is trained to differentiate between the source and target features. Simultaneously, the target encoder E^t is trained to generate target features that cannot be distinguished from the source features by the domain discriminator network. The loss function for the adversarial training process between the discriminator network D and the target encoder E^t is expressed as follows:

$$\min_{E^t}\max_{D}\mathcal{L}_{adv} = \frac{1}{n}\sum_{i=1}^{n}[\log D(E^s(X_i^s))] + \frac{1}{n}\sum_{i=1}^{n}[\log(1 - D(E^t(X_i^t)))]. \tag{7}$$

Here, n denotes the number of samples in the minibatch, X_i^s and X_i^t denote the source and target samples, respectively. The target encoder E^t is iteratively refined to minimize the loss function $\mathcal{L}_{adv}$, while the discriminator network D is trained to maximize the same loss. Ultimately, the trained target encoder E^t will be capable of extracting features F_i^t that exhibit minimal differences compared to the source features.

E. Local Contrastive Learning

The local contrastive learning part consists of the dual aspects of channel-level and the temporal-level contrastive module. The channel-level contrastive module is used to maximize the mutual information between the target domain channel weights and the target domain channel weights, preserving task-specific channel information. Meanwhile, the temporal-level contrastive module aims to maximize the mutual information between the target domain features and the source domain features to capture local domain-invariant information.

1) Channel-level Contrastive Module: To preserve the sensitivity of the target domain to specific feature channels, we construct a channel-level contrastive learning module based on noise contrastive estimation for information maximizing self-supervised learning (InfoNCE) [24]. This module is shown in Fig. 3.

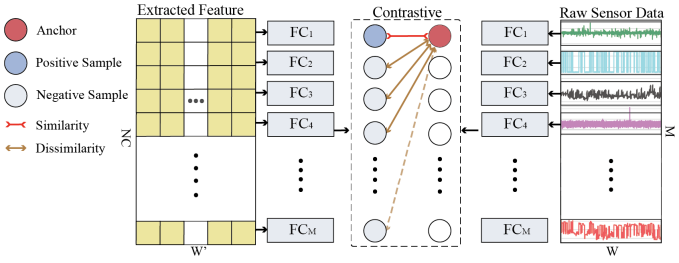


Fig. 3. Channel-level contrastive learning module.

For the given input from target domain $\mathcal{D}^t$, we can get target features $F_i^t = E^t(X_i^t)$ through E^t. Let $X_i^t = \{\mathbf{x}_{i,1}^t, \mathbf{x}_{i,2}^t, ..., \mathbf{x}_{i,M}^t\}$ denote the sequences of the input channel features, with $\mathbf{x}_{i,j}^t \in \mathbb{R}^W$ being the feature from the jth sensor of the ith input data. To maximize the mutual information between target domain features F_i^t and target domain raw data X_i^t, we apply fully-connected networks $\Theta : \mathbb{R}^W \to \mathbb{R}^W$ to map each $\mathbf{x}_{i,j}^t$ to dimension W. The transformed input data is $U_i^t = \{\Theta_1(\mathbf{x}_{i,1}^t), \Theta_2(\mathbf{x}_{i,2}^t), ..., \Theta_M(\mathbf{x}_{i,M}^t)\}$. To compare the input and the target features, M fully-connected are applied to guarantee that the mapped features and $\mathbf{u}_i^t$ should have the same dimension, where $V_i^t = \{\Theta_1^{'}(F_i^t), \Theta_2^{'}(F_i^t), ..., \Theta_M^{'}(F_i^t)\}$ is the transformed features which has the dimension $M \times W$, and $\Theta' : \mathbb{R}^{N \cdot C \times W'} \to \mathbb{R}^W$ is the fully-connected layers to map F_i^t to dimension W.

Contrastive learning aligns the representations of all positive samples, while repelling the representations of the negative ones. The positve and negative representations should be defined for the feature in a certain channel m. After the transformation, $\mathbf{x}_{i,m}^t \in \mathbb{R}^{N \cdot C}$ is mapped through the mapping layer Θ_m^s to obtain the anchored sample $\mathbf{u}_{i,m}^s$. The positive sample can be found of $\mathbf{v}_{i,m}^t$, which has the same position with the anchor sample, where $\mathbf{v}_{i,m}^t = \Theta_m^{'}(F_i^t)$. The rest of the transformed representations of F_i^t, i.e., $\{\mathbf{v}_{i,j}^t | j = 1, 2, ..., M; j \neq m\}$, are defined as negative samples. The positive and negative samples for $m = 1$ is illustrated in Fig. 3.

The InfoNCE loss is applied to maximize the mutual information by contrasting between positive and negative samples, which is shown as follows:

$$\min_{E^t,\Theta,\Theta'}\mathcal{L}_{clc} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{m=1}^{M}\log\frac{\exp(\mathbf{u}_{i,m}^t \cdot \mathbf{v}_{i,m}^t)}{\sum_{\mathbf{u}_{i,j}^t \in U_i^t}\exp(\mathbf{u}_{i,m}^t \cdot \mathbf{v}_{i,j}^t)}. \tag{8}$$

The algorithm of local contrastive learning is shown in Algorithm 1.

2) Temporal-Level Contrastive Module: Localized critical information under varying operating conditions is essential, as it may capture key details about specific stages of the failure process. In order to effectively identify effective information in local data while preserving temporal correlations, we introduce temporal-level contrastive learning into transfer learning. A dynamic temperature coefficient is designed using the kernel method, so that the model optimization can be dynamically adjusted according to the distance and similarity between

Algorithm 1: Channel-Level Contrastive Loss

Input: Target features: $F_i^t = E^t(X_i^t)$
Output: Channel-level contrastive loss $\mathcal{L}_{\text{clc}}$

Channel weights of inputs:
$U_i^t = \{\Theta_1(\mathbf{x}_{i,1}^t), \Theta_2(\mathbf{x}_{i,2}^t), ..., \Theta_M(\mathbf{x}_{i,M}^t)\}$
Channel weights of features:
$V_i^t = \{\Theta_1^{'}(F_i^t), \Theta_2^{'}(F_i^t), ..., \Theta_M^{'}(F_i^t)\}$
Compute $\mathcal{L}_{\text{clc}}$ by (8)
return $\mathcal{L}_{\text{clc}}$

samples in different time steps. The details of the temporal-level contrastive module is shown in Fig. 4.

Let G_i^s and G_i^t be the transpose of source feature F_i^s and target feature F_i^t, respectively. Then $G_i^s = \{\mathbf{g}_{i,1}^s, \mathbf{g}_{i,2}^s, ..., \mathbf{g}_{i,W'}^s\}$ and $G_i^t = \{\mathbf{g}_{i,1}^t, \mathbf{g}_{i,2}^t, ..., \mathbf{g}_{i,W'}^t\}$ denote the sequences of temporal features in the source and target domains. For contrastive learning, fully-connected networks $\Omega : \mathbb{R}^{N \cdot C} \rightarrow \mathbb{R}^{M'}$ are applied to obtain transformed features for comparison. To be specific, the feature at a certain time point $\mathbf{g}_{i,k}^s \in \mathbb{R}^{N \cdot C}$ is mapped through the source domain mapping layer Ω_k^s to obtain the anchored sample $\mathbf{h}_{i,k}^s$. Similarly, the features $\mathbf{g}_{i,k}^t$ at the same time point are mapped through the target domain mapping layer Ω_k^t to obtain positive samples $\mathbf{h}_{i,k}^t$. Features at other time points in the target domain are mapped to negative sample sets $\{\mathbf{h}_{i,j}^t \mid j = 1, 2, ..., W'; j \neq k\}$ through the target domain mapping layers.

For temporal-level contrastive learning, the the degree of "positiveness" is determined based on the similarity between the representations of the samples and the anchor to capture spatial correlations. The kernel function is applied to compute the similarity [29], [30], as temporal correlations should be considered for different time steps. To be specific, the weight factor $\boldsymbol{\tau}_i^k$ for the kth anchored sample is computed by the kernel function to act like a temperature parameter, by giving less weight to the samples which are farther away from the anchor in the kernel space [11]. For a proper kernel choice, samples closer than $\mathbf{f}_{i,k}^t$ will be repelled with very low strength ($\sim$0). The Gaussian kernel function is applied, which is defined as follows:

$$\boldsymbol{\tau}_i^k = \mathcal{K}(\mathbf{l}_i^k) = \exp(-\gamma ||\mathbf{l}_i^k||^2), \tag{9}$$

where γ represents the hyperparameter, and $\mathbf{l}_i^k = [l_{i,1}, l_{i,2}, ..., l_{i,W'}]$ is the vector which calculates the Euclidean distance $l_{i,j}^k = ||\mathbf{p}_{i,j}^t - \mathbf{p}_{i,k}^s||_2$ of the position embeddings between positive/negative sample $\mathbf{h}_{i,j}^t$ and anchor sample $\mathbf{h}_{i,k}^s$.

By introducing the temperature parameter which strengthens the repulsion of samples based on their distances from the anchor in the kernel space, the infoNCE loss can be defined as:

$$\min_{E^t,\Omega} \mathcal{L}_{tlc} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{W'} \log \frac{\exp(\mathbf{h}_{i,k}^s \cdot \mathbf{h}_{i,k}^t / \tau_{i,k}^k)}{\sum_{\mathbf{h}_{i,j}^t \in H_i^t} \exp(\mathbf{h}_{i,k}^s \cdot \mathbf{h}_{i,j}^t / \tau_{i,j}^k)}. \tag{10}$$

The algorithm process is shown in Algorithm 2.

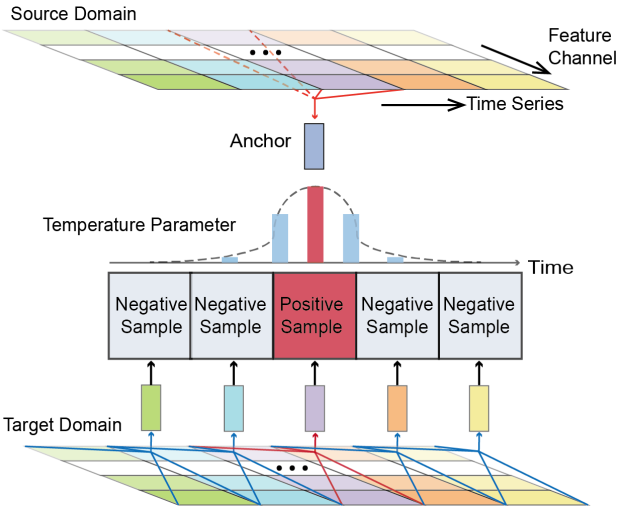


Fig. 4. Temporal-level contrastive module.

Algorithm 2: Temporal-Level Contrastive Loss

Input: Source feature: $F_i^s = E^s(X_i^s)$, Target feature: $F_i^t = E^t(X_i^t)$
Output: Temporal-level contrastive loss $\mathcal{L}_{\text{tl}}$

Compute transformed feature sequences:
$H_i^s = [\Omega_k^s(\mathbf{g}_{i,1}^s), \Omega_k^s(\mathbf{g}_{i,2}^s), ..., \Omega_k^s(\mathbf{g}_{i,W'}^s)]$,
$H_i^t = [\Omega_k^t(\mathbf{g}_{i,1}^t), \Omega_k^t(\mathbf{g}_{i,2}^t), ..., \Omega_k^t(\mathbf{g}_{i,W'}^t)]$
Compute the temperature value $\boldsymbol{\tau}_i^k$ by (9)
Compute temporal-level contrastive loss $\mathcal{L}_{\text{tlc}}$ by (10)

3) Overall Loss Function: In this study, the transfer learning method was jointly optimized through end-to-end approach. The overall loss function is shown as follow:

$$\min_{E^t,\Theta,\Theta',\Omega} \max_{D} J(E^t, D, \Theta, \Theta', \Omega) = \mathcal{L}_{adv} + \alpha \mathcal{L}_{clc} + \beta \mathcal{L}_{tlc} \tag{11}$$

where α and β represent the parameters that control the weights among learning of globally invariant features, retaining unique information in the local domain, and learning local invariant features.

IV. Experiments

To assess the effectiveness and performance of the proposed GALC method, we employ an ion mill etching (IME) dataset obtained from an ion etching process for our experimental analysis. The IME process is shown in Fig. 5. The IME dataset is a real-world dataset examining the fault behavior of multiple ion mill etching tools utilized in an IME process. This dataset was made available through the 2018 PHM data challenge [1].

In our study, the degradation patterns across different fault modes exhibit similarities. This is due to the common physical and operational processes driving system degradation, resulting in comparable temporal patterns of performance decline. The systems used operate in similar environments and share functional principles, further supporting the assumption of similar degradation patterns. While minor differences in

Algorithm 3: GALC Transfer Learning Method

Input: Source Domain:$\mathcal{D}^s = \{X_i^s, y_i^s\}_{i=1}^{n_s}$
Target Domain:$\mathcal{D}^t = \{X_j^t\}_{i=1}^{n_t}$
Initialize E^t with E^s parameters
Output: Trained target encoder E^t
Trained source encoder E^s
Domain Discriminator D

for *number of iterations* **do**
  Sample minibatch of n source samples X_i^s
  Sample minibatch of n target samples X_i^t
  Source features: $F_i^s = E^s(X_i^s)$
  Target features: $F_i^t = E^t(X_i^t)$
  Feed F_i^s and F_i^t to D
  Compute adversarial loss $\mathcal{L}_{\text{adv}}$ by (7)
  Compute $\mathcal{L}_{clc}$ based on Algorithm 1
  Compute $\mathcal{L}_{tlc}$ based on Algorithm 2
  Update E^t, D by (11)
end

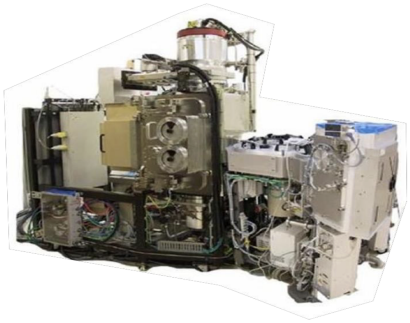


Fig. 5. Ion mill etching (IME) system. Three failure modes: Flowcool pressure dropped below limit (F1), Flowcool pressure too high check flowcool pump (F2), and Flowcool leak (F3). The process of ion mill etching typically consists of the following steps: Inserting a wafer into the mill, Configure wafer settings, Processing the wafer for a set amount of time, Remove wafer from mill. [1]

degradation behavior may arise from hardware or operational variations, the fundamental causes of degradation are consistent across systems within the same domain.

A. Experimental Settings

The raw IME data are collected from 20 distinct IME tools, comprising 5 classification variables and 19 numerical variables. Among the numerical variables, 9 data types are chosen as inputs for our proposed model. Besides, we are interested in three failure modes of the flowcool systems: when the flowcool pressure drops below the limit (F1), when the flowcool pressure is excessively high and requires checking the flowcool pump (F2), and when there is a flowcool leak (F3). These selected variables provide a multi-faceted description (voltage, electric potential, current velocity and velocity of flow) of the overall failure process.

During the model training and testing phases, we include all fault modes F1, F2, and F3, as well as five tools (01_M02, 02_M01, 02_M02, 04_M01, 05_M01) that exhibit the highest number of failure cycles. Tool 05_M02 is designated as the validation dataset for this method. The dataset is randomly partitioned into training (70%), validation (20%), and testing (10%) sets based on periodicity. Furthermore, we construct a sliding window with a length of 500 ($W = 500$) and a step size of 1 to segment the data within one cycle.

[1]https://www.phmsociety.org/events/conference/phm/18/data-challenge

TABLE I
HYPER-PARAMETER CONFIGURATIONS OF E^s, E^t, D AND R

Multi-scale Transformer Network E^s **and** E^t	No. of scales	3
	No. of conv blocks	1
	kernel size	[2, 5, 10]
	No. of heads of attention	4
	Dropout	0.1
Domain Discriminator D	No. of hidden nodes1	128
	Dropout	0.4
	No. of hidden nodes2	64
	ReLU	/
RUL predictor R	No. of hidden nodes1	128
	Dropout	0.2
	No. of hidden nodes2	64
	ReLU	/
Mapping Layer Θ	No. of hidden nodes	128
	ReLU	/
Mapping Layer Θ'	No. of hidden nodes	64
	ReLU	/
Mapping Layer Ω	No. of hidden nodes	128
	ReLU	/

Regarding the experimental setup, the GALC method comprises several modules: source domain feature extractor E^s, target domain feature extractor E^t, RUL predictor R, domain discriminator D, channel-level contrastive module, and temporal-level contrastive module. The network architecture and hyperparameter configurations of E^s, E^t, R and D are detailed in Table I. Grid search algorithms are employed in the fully connected layers and mapping layers of the channel-level contrastive module and temporal-level contrastive module to optimize model hyper-parameters.

To enhance feature learning, our channel-level contrastive learning module defines features from different channels as negative pairs, while a signal and its high-level representation form positive pairs. Given the nonlinear dependencies and latent correlations in industrial sensor data, distinguishing these pairs in the raw feature space is challenging. To address this, we employ projection heads to map features into a high-dimensional space, preserving intrinsic feature associations and enhancing feature separability.

During the training process of the model, Xavier's normal initializer is used to initialize the weights and biases of the network. The optimizer Nadam is used for weight updates, with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-7}$. The initial values of learning rate for E^t, D and R are set to 2×10^{-3}, 2×10^{-3} and 5×10^{-2} respectively. If the loss function value of the validation dataset does not decrease after 20 epoches, the learning rate becomes half of the previous epoch. In addition, the batch size is set to 256 and the maximum training epoch is set to 500.

TABLE II
PERFORMANCE COMPARISON OF FOUR CROSS-DOMAIN TOOLS

Tansfer Learning Method	01_M02 → 02_M01			01_M02 → 05_M01			04_M01 → 05_M02			05_M01 → 05_M02		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
GALC	1297.71	**24.33**	1576.58	**1134.75**	24.68	**1207.88**	1136.62	**23.63**	**1341.37**	**1238.13**	**23.65**	**1454.43**
DAN [31]	**1103.64**	25.06	**1026.72**	1215.09	24.94	1274.21	1321.11	24.07	1543.04	1291.52	25.06	1502.12
DANN [32]	1387.67	25.37	1702.17	1429.34	26.55	1649.22	**1103.82**	24.79	1384.33	1492.14	24.27	1548.23
DAAN [33]	1591.35	26.15	1890.23	1163.01	24.51	1328.72	1483.99	25.28	1714.21	1407.74	24.76	1504.41
BNM [34]	1737.38	27.73	2197.23	1329.41	25.09	1549.27	1328.28	25.28	1532.52	1424.29	23.87	1537.32
MDD [35]	1729.48	26.61	1994.07	1203.37	**24.32**	1388.64	1239.44	23.99	1377.23	1243.31	24.22	1481.14
WDGRL [36]	2139.01	29.46	2731.84	1244.95	24.64	1431.29	1329.21	24.10	1578.79	1253.43	24.11	1456.82

TABLE III
PERFORMANCE COMPARISON OF FOUR CROSS-DOMAIN FAILURE MODES

Tansfer Learning Method	F1 → F2			F1 → F3			F2 → F3			F3 → F1		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
GALC	**1731.13**	**26.79**	**1835.79**	**1704.82**	**26.96**	**1923.71**	**2047.27**	**27.80**	**2251.30**	**1812.96**	**27.17**	**1994.26**
DAN	1695.47	27.50	1865.02	1707.64	27.13	1981.10	2232.46	29.04	2554.06	1988.70	28.46	2227.35
DANN	2027.79	28.09	2252.80	1928.57	28.21	2140.72	2119.36	30.74	2691.59	1917.08	28.24	2319.67
DAAN	1898.63	27.77	2104.40	2154.95	28.44	2284.25	2216.05	31.94	2858.25	2060.37	27.41	2184.59
BNM	2371.04	28.24	2236.17	2177.35	29.04	2395.09	2624.24	29.99	2781.70	1993.36	27.67	2252.80
MDD	2209.96	29.16	2453.06	1854.37	27.22	2076.89	2454.86	30.62	2798.54	2268.02	30.67	2828.10
WDGRL	2180.65	28.92	2398.71	2144.10	28.28	2358.51	2565.71	30.40	2745.31	2373.72	29.91	2741.91

In the stage of evaluating model performance, we apply the mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE) as metrics [1] to assess the effectiveness of the proposed method as well as that of other competing methods:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{12}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}, \tag{13}$$

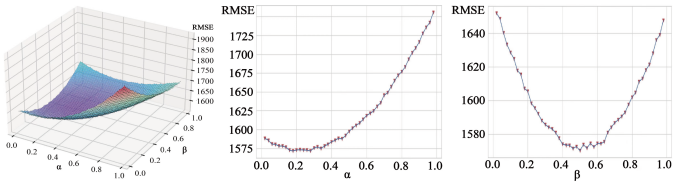
$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{y}_i - y_i}{y_i}\right|, \tag{14}$$

where N denotes the total number of samples, y_i is the actual RUL value, and $\hat{y}_i$ is the predicted RUL value. This experiment was conducted on Intel (R) Xeon (R) Gold6226R@2.90 Development of GHz CPU and NVIDIA RTX A6000 GPU devices. In order to reduce the impact of randomness, the prediction result of the proposed method in this experiment is the average value after 10 runs.

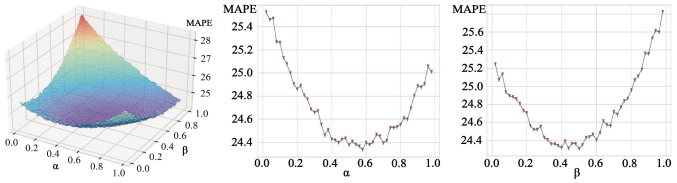
B. Comparative Experiments

In order to evaluate the effectiveness of the transfer learning method in predicting the failure RUL of ion erosion systems, we constructed 23 transfer learning modes using data from three fault modes (F1, F2, F3) and six different ion erosion systems (01_M02, 02_M01, 02_M02, 04_M01, 05_M01, 05_M02). This includes migration patterns between 17 different systems and migration patterns between 6 different failure modes. In these 23 transfer scenarios, we compared our method with seven other transfer learning methods. 1) Deep adaptation networks (DAN) [31], 2) Generative adversarial networks (DANN) [32], 3) Dynamic adversarial adaptation network (DAAN) [33], 4) Batch nuclear-norm maximization (BNM) [34], 5) Margin disparity discrepancy (MDD) [35], 6) Wasserstein distance guided representation learning (WDGRL) [36].

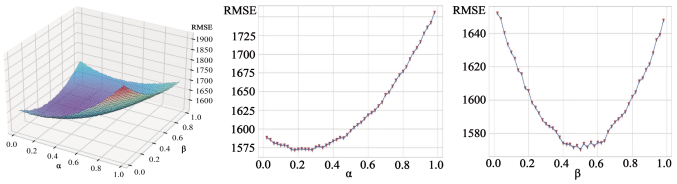
The results of the comparative analysis of transfer learning across different tools are presented in TABLE II, while TABLE III illustrates the findings of transfer learning across different fault modes. Compared with other methods, our proposed method demonstrates excellent performance. The experimental results indicate a significant reduction in RMSE, averaging 5.32% for cross-domain IME tools and 6.39% for cross-domain fault modes. For the transfer learning across different tools, four cross-domain scenarios is presented, and our proposed method performs the best except the except the 01_M02 → 05_M01 mode considering the MAPE score. When calculating the RMSE score, except for the scenario 01_M02 → 02_M01, our model achieved the optimal results. However, our model's performance is better than the rest comparable results and close to the optimal transfer effect, showing competitive results with existing studies. For the transfer learning across different failure modes, our model performs the best in various scenarios in terms of MAE, MAPE and RMSE. The comparison result show that the proposed method not only facilitates effective knowledge transfer at both global and local levels but also adeptly maintains the sensitivity of distinct domain feature channels within the target domain, thereby demonstrating strong performance in transfer learning throughout ion etching processes.



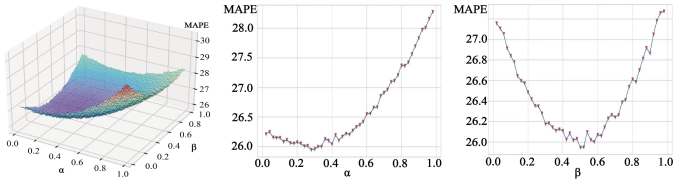
(a) Variations of RMSE in 01_M02 → 02_M01



(b) Variations of MAPE in 01_M02 → 02_M01



(c) Variations of RMSE in 01_M02 → 05_M01



(d) Variations of MAPE in 01_M02 → 05_M01

Fig. 6. Variations of RMSE and MAPE with respect to α and β in two different transfer modes.

C. Ablation Study

To validate the effectiveness of each module within the global adversarial and local contrastive transfer learning method, we utilized five transfer scenarios as exemplars and assessed each module's efficacy under two evaluation metrics (RMSE and MAPE). The specific ablation experimental results are depicted in TABLE IV, where GA represents the global adversarial module, LC represents the local channel-level contrastive module, and LT represents the local temporal-level contrastive module. The prediction results of the methods is the average value (mean ± std) after 10 runs.

The findings clearly indicate that the global adversarial part is critical for effective knowledge transfer, as configurations lacking this part demonstrated the lowest performance levels. Furthermore, the integration of either a local channel-level or temporal-level contrastive module alongside the global adversarial part resulted in improved performance without compromising overall effectiveness. Notably, employing both the channel-level and temporal-level contrastive modules in conjunction with the global adversarial part produced substantial enhancements in model performance, highlighting the importance of a multi-faceted approach to transfer learning that leverages complementary strengths of each module to optimize predictive accuracy in complex environments.

TABLE IV
THE RESULTS OF ABLATION EXPERIMENTS

Group	Transfer Mode	MAE	RMSE	MAPE
GA+LC +LT	01_M02 → 02_M01	**1285.02**±2.52	**1426.37**±2.04	**24.33**±0.03
	01_M02 → 02_M02	**1214.45**±2.12	**1587.44**±2.31	**24.82**±0.04
	01_M02 → 04_M01	**1503.71**±2.27	**1732.19**±1.74	**25.82**±0.02
	01_M02 → 05_M01	**946.44**±1.93	**1207.88**±2.07	**24.68**±0.01
	01_M02 → 05_M02	**1542.73**±2.41	**1804.92**±3.03	**26.03**±0.01
GA+LT	01_M02 → 02_M01	1809.09±2.79	1990.94±2.42	27.34±0.06
	01_M02 → 02_M02	1783.37±2.79	1965.67±2.93	27.11±0.10
	01_M02 → 04_M01	1586.32±2.47	1729.09±2.55	27.21±0.05
	01_M02 → 05_M01	1586.79±2.88	1682.19±3.91	25.54±0.8
	01_M02 → 05_M02	1676.58±2.57	1861.50±2.63	26.38±0.6
GA+LC	01_M02 → 02_M01	1568.69±2.07	1709.88±2.31	25.20±0.10
	01_M02 → 02_M02	1610.39±2.05	1787.54±2.37	25.07±0.03
	01_M02 → 04_M01	1657.75±2.77	1856.68±2.99	26.74±0.04
	01_M02 → 05_M01	1428.65±2.78	1614.38±2.97	24.82±0.05
	01_M02 → 05_M02	1619.30±2.39	1846.43±2.45	26.27±0.03
LC+LT	01_M02 → 02_M01	2542.94±2.01	2898.96±1.71	29.64±0.09
	01_M02 → 02_M02	2481.42±3.81	2804.64±3.74	29.60±0.06
	01_M02 → 04_M01	2401.39±4.04	2689.56±3.92	30.88±0.04
	01_M02 → 05_M01	2652.21±3.81	2943.95±3.91	29.20±0.04
	01_M02 → 05_M02	2717.71±4.81	2989.48±5.12	29.22±0.09
GA	01_M02 → 02_M01	1713.68±3.97	1867.92±2.81	27.61±0.07
	01_M02 → 02_M02	1977.78±5.27	2136.66±4.07	27.76±0.05
	01_M02 → 04_M01	2115.74±5.02	2263.85±3.85	28.29±0.03
	01_M02 → 05_M01	1913.77±4.28	2028.60±4.74	27.60±0.03
	01_M02 → 05_M02	2074.71±4.71	2199.19±4.62	27.25±0.04

D. Sensitivity Analysis

1) Coefficient of the GALC Loss (α and β): This section examines the sensitivity of the proposed GALC model to variations in the GALC loss coefficient, α and β. Experiments were conducted in 2 different transfer scenarios (01_M02 → 02_M01 and 01_M02 → 05_M01) with α and β ranging from 0.02 to 1.0, as shown in Fig. 6. Among these, (a) depict the three-dimensional variations of RMSE with respect to α and β, cross-sectional views of the optimal point along the α axis and cross-sectional views of the optimal point along the β axis in 01_M02 → 02_M01 mode, while figure (b) shows the three-dimensional variations of MAPE with respect to α and β. Figure (c) depicts the three-dimensional variations of RMSE

TABLE V
PERFORMANCE COMPARISON OF FIVE TRANSFER MODES

Method	01_M02 → 02_M01		01_M02 → 02_M02		01_M02 → 04_M01		01_M02 → 05_M01		01_M02 → 05_M02		Number of parameters	Training Time
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE		
Our Encoder	24.33	1426.37	24.82	1587.44	25.79	1732.19	24.68	1207.88	26.03	1804.92	1,382,472	2072.428
Transformer Encoder	27.34	1990.94	27.11	1965.67	27.21	1901.84	25.54	1682.19	26.38	1861.50	1,425,882	2243.213
TCN	25.20	1709.88	25.07	1787.54	26.74	1856.68	24.82	1614.38	26.27	1861.50	38,457,835	1974.055
LSTM	27.61	1867.92	27.76	2136.66	28.29	2263.85	27.60	2028.60	27.25	2989.48	1,194,497	2318.109
TCN-LSTM	29.64	2898.96	29.60	2804.64	30.88	2689.56	29.20	2943.95	29.22	2989.48	39,062,406	1903.734

with respect to α and β, cross-sectional views of the optimal point along the α axis and cross-sectional views of the optimal point along the β axis in 01_M02 $\rightarrow$ 05_M01 mode. Figure (d) show the three-dimensional variations of MAPE with respect to α and β in the same mode. Our model effectively resolves the potential conflict between global adversarial learning and local contrastive learning and finds optimal α and β values in different scenarios through experiments, ensuring that these two seemingly competing objectives, $\mathcal{L}_{adv}$ and $\mathcal{L}_{clc}$, are harmonized rather than interfering with each other.

2) Encoder of the GALC Method: To assess the sensitivity of the proposed transfer learning method, we also explored the impact of different encoders on performance. Specifically, we compared the Transformer-based multi-branch encoder, selected for this study, against other commonly used architectures, including the Transformer encoder, TCN [1], LSTM [37], and TCN-LSTM [1] model. Experiments were conducted across five distinct transfer modes, with performance evaluated using the RMSE and MAPE metrics, as summarized in TABLE V. The experimental results indicate that, with the exception of LSTM, the differences in RMSE and MAPE metrics among the remaining models are relatively minor. This suggests that, given a well-chosen encoder, the proposed GALC method can effectively achieve robust transfer performance. Furthermore, the Transformer-based multi-branch encoder consistently outperforms the alternative models, highlighting its superior ability to extract informative features for accurate RUL prediction.

In addition, we compare our encoder's parameter count and training time with other models, as shown in Table V. Our encoder trains faster than Transformer Encoder and LSTM while being comparable to TCN and TCN-LSTM. Although our model has slightly higher complexity, it requires fewer parameters and converges more quickly. Considering both accuracy and computational efficiency, our encoder demonstrates superior performance.

V. CONCLUSION

This article proposes a novel GALC transfer learning approach for RUL prediction across diverse domains. The global adversarial part aligns feature representations between source and target domains, while the local contrastive part captures domain-specific information, preserving both temporal and channel-level features. By integrating global domain-invariant feature learning with channel-specific contrastive learning and local temporal feature alignment, GALC effectively captures both domain-invariant and domain-specific feature, significantly enhancing the generalization and adaptability of RUL prediction models. This methodology provides a robust foundation for handling the complexity of real-world, multi-domain operational conditions.

In future work, we plan to validate the effectiveness of our approach in more complex and diverse industrial scenarios to further demonstrate its practicality and robustness. Additionally, we aim to enhance the computational efficiency of our model while maintaining high predictive performance, ensuring its practical applicability in real-world deployments, particularly in resource-constrained environments.

REFERENCES

[1] C. Liu, L. Zhang, J. Li, J. Zheng, and C. Wu, "Two-stage transfer learning for fault prognosis of ion mill etching process," *IEEE Transactions on Semiconductor Manufacturing*, vol. 34, no. 2, pp. 185–193, 2021.

[2] M. Kurucan, M. Özbaltan, Z. Yetgin, and A. Alkaya, "Applications of artificial neural network based battery management systems: A literature review," *Renewable and Sustainable Energy Reviews*, vol. 192, p. 114262, 2024.

[3] L. Wu, W. Guo, Y. Tang, Y. Sun, and T. Qin, "Remaining useful life prediction of lithium-ion batteries based on neural network and adaptive unscented kalman filter," *Electronics*, vol. 13, no. 13, p. 2619, 2024.

[4] J. Zhang, K. Zhang, Y. An, H. Luo, and S. Yin, "An integrated multitasking intelligent bearing fault diagnosis scheme based on representation learning under imbalanced sample condition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, pp. 6231–6242, 2023.

[5] R. Jin, Z. Chen, K. Wu, M. Wu, X. Li, and R. Yan, "Bi-lstm-based two-stream network for machine remaining useful life prediction," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–10, 2022.

[6] I. Misbah, C. Lee, and K. Keung, "Fault diagnosis in rotating machines based on transfer learning: literature review," *Knowledge-Based Systems*, p. 111158, 2023.

[7] X. Li, W. Zhang, X. Li, and H. Hao, "Partial domain adaptation in remaining useful life prediction with incomplete target data," *IEEE/ASME Transactions on Mechatronics*, 2023.

[8] B.-L. Lu, Z.-H. Liu, H.-L. Wei, L. Chen, H. Zhang, and X.-H. Li, "A deep adversarial learning prognostics model for remaining useful life prediction of rolling bearing," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 4, pp. 329–340, 2021.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[10] W. Zaremba, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[12] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2285–2294.

[13] M. Ma and Z. Mao, "Deep-convolution-based lstm network for remaining useful life prediction," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1658–1667, 2020.
[14] B. Zraibi, C. Okar, H. Chaoui, and M. Mansouri, "Remaining useful life assessment for lithium-ion batteries using cnn-lstm-dnn hybrid method," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4252–4261, 2021.
[15] Z. Zhang, W. Song, and Q. Li, "Dual-aspect self-attention based on transformer for remaining useful life prediction," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.
[16] B. Wang, Y. Lei, N. Li, and W. Wang, "Multiscale convolutional attention network for predicting remaining useful life of machinery," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 8, pp. 7496–7504, 2020.
[17] H. Liu, Z. Liu, W. Jia, and X. Lin, "Remaining useful life prediction using a novel feature-attention-based end-to-end approach," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1197–1207, 2020.
[18] Z. Chen, M. Wu, R. Zhao, F. Guretno, R. Yan, and X. Li, "Machine remaining useful life prediction via an attention-based deep learning approach," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 3, pp. 2521–2531, 2020.
[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.
[20] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3313–3332, 2021.
[21] M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, and X. Li, "Adversarial transfer learning for machine remaining useful life prediction," in *2020 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 2020, pp. 1–7.
[22] M. Ragab, Z. Chen, M. Wu, C. S. Foo, C. K. Kwoh, R. Yan, and X. Li, "Contrastive adversarial domain adaptation for machine remaining useful life prediction," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5239–5249, 2020.
[23] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
[24] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.
[25] W. Mao, J. Chen, J. Liu, and X. Liang, "Self-supervised deep domain-adversarial regression adaptation for online remaining useful life prediction of rolling bearing under unknown working condition," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1227–1237, 2022.
[26] J. Zhuang, M. Jia, Y. Ding, and P. Ding, "Temporal convolution-based transferable cross-domain adaptation approach for remaining useful life estimation under variable failure behaviors," *Reliability Engineering & System Safety*, vol. 216, p. 107946, 2021.
[27] T. Wang, D. Guo, and X.-M. Sun, "Contrastive generative replay method of remaining useful life prediction for rolling bearings," *IEEE Sensors Journal*, 2023.
[28] Z. Yuan and R. Wang, "Multi-scale and multi-branch transformer network for remaining useful life prediction in ion mill etching process," *IEEE Transactions on Semiconductor Manufacturing*, vol. 37, no. 1, pp. 67–75, 2024.
[29] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 661–18 673, 2020.
[30] C. A. Barbano, B. Dufumier, E. Tartaglione, M. Grangetto, and P. Gori, "Unbiased supervised contrastive learning," *arXiv preprint arXiv:2211.05568*, 2022.
[31] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International Conference on Machine Learning*. PMLR, 2015, pp. 97–105.
[32] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.
[33] C. Yu, J. Wang, Y. Chen, and M. Huang, "Transfer learning with dynamic adversarial adaptation network," in *2019 IEEE international Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 778–786.
[34] S. Cui, S. Wang, J. Zhuo, L. Li, Q. Huang, and Q. Tian, "Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3941–3950.
[35] Y. Zhang, T. Liu, M. Long, and M. Jordan, "Bridging theory and algorithm for domain adaptation," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7404–7413.
[36] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
[37] S. Wu, Y. Jiang, H. Luo, and S. Yin, "Remaining useful life prediction for ion etching machine cooling system using deep recurrent neural network-based approaches," *Control Engineering Practice*, vol. 109, p. 104748, 2021.