Debiased Contrastive Learning With Supervision Guidance for Industrial Fault Detection

Rongyao Cai, *Graduate Student Member, IEEE*, Wang Gao, Linpeng Peng, Zhengming Lu, Kexin Zhang, *Graduate Student Member, IEEE*, and Yong Liu, *Member, IEEE*

***Abstract*—The time series self-supervised contrastive learning framework has succeeded significantly in industrial fault detection scenarios. It typically consists of pretraining on abundant unlabeled data and fine-tuning on limited annotated data. However, the two-phase framework faces three challenges: Sampling bias, task-agnostic representation issue, and angular-centricity issue. These challenges hinder further development in industrial applications. This article introduces a debiased contrastive learning with supervision guidance (DCLSG) framework and applies it to industrial fault detection tasks. First, DCLSG employs channel augmentation to integrate temporal and frequency domain information. Pseudolabels based on momentum clustering operation are assigned to extracted representations, thereby mitigating the sampling bias raised by the selection of positive pairs. Second, the generated supervisory signal guides the pretraining phase, tackling the task-agnostic representation issue. Third, the angular-centricity issue is addressed using the proposed Gaussian distance metric measuring the radial distribution of representations. The experiments conducted on three industrial datasets (ISDB, CWRU, and practical datasets) validate the superior performance of the DCLSG compared to other fault detection methods.**

***Index Terms*—Data augmentation, debiased contrastive learning, fault detection, representation extraction, similarity metric, weak supervision guidance.**

Manuscript received 22 April 2024; revised 13 June 2024 and 28 June 2024; accepted 1 July 2024. Date of publication 18 July 2024; date of current version 5 November 2024. This work was supported in part by the National Key R&D Program of China under Grant 2021YFB2012300, and in part by the Postdoctoral Fellowship Program of CPSF under Grant GZC20241491. Paper no. TII-24-1896. *(Rongyao Cai and Wang Gao contributed equally to this work.) (Corresponding authors: Kexin Zhang; Yong Liu.)*

Rongyao Cai, Linpeng Peng, Zhengming Lu, and Yong Liu are with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China (e-mail: rycai@zju.edu.cn; penglinpeng@zju.edu.cn; lukelu@zju.edu.cn; yongliu@iipc.zju.edu.cn).

Wang Gao is with the Science and Technology on Complex System Control and Intelligent Agent Cooperation Laboratory, Beijing 1000191, China (e-mail: gaowang_fly@163.com).

Kexin Zhang is with the Huzhou Institute of Zhejiang University, Huzhou 313000, China, and also with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China (e-mail: zhangkexin@zju.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2024.3424561.

Digital Object Identifier 10.1109/TII.2024.3424561

I. Introduction

PROGNOSTICS and health management (PHM) methods aim to monitor the operating status of machinery, diagnose faults, and schedule maintenance to address anomalies using available information [1]. Deep neural networks are widely used in PHM due to their powerful feature extraction capabilities. However, reliable deep neural network models often require substantial annotated data, which can be constrained by factors like domain expertise, increasing labor costs, and data privacy concerns. To address this challenge, promising contrastive learning has emerged, consisting of a pretraining and fine-tuning phase [2].

Current research of fault detection based on contrastive learning focuses on developing intricate similarity architectures [2] and mining multigranularity representations [3], [4], [5]. Zhang et al. [6] simultaneously extracts signal temporal and frequency domain representations and allows the two representations to be matched in high dimensional space. Zhang et al. [1] translated the relative relationships between analog signals into images, and extracted image features for detecting valve stiction. These research have fully exploited the potential of contrastive learning and achieved good results in industrial fault detection [7]. However, this two-phase contrastive learning framework suffers three critical issues: 1) the task-agnostic representation issue; 2) sampling bias; and 3) angular-centricity issue.

In a two-phase contrastive learning framework, the feature extractor refines high-quality representations from unlabeled data during pretraining. In the subsequent fine-tuning phase, a smaller network, in conjunction with a small-scale labeled dataset (the annotated fine-tuning dataset), is employed to address downstream tasks. This two-phase framework effectively harnesses labeled data and consistently demonstrates strong performance across a range of downstream tasks [6]. However, a significant challenge, the task-agnostic representation issue, arises due to the separated pretraining and fine-tuning phases. This issue signifies that the relationship between the extracted representations and downstream task requirements is not explicitly considered, potentially resulting in suboptimal performance and reduced model generalization in practical applications. One viable approach to address the task-agnostic representation issue is flexibly employing the annotated fine-tuning dataset to generate a supervisory signal during pretraining, guiding the alignment of representations with the specific objectives of downstream tasks.

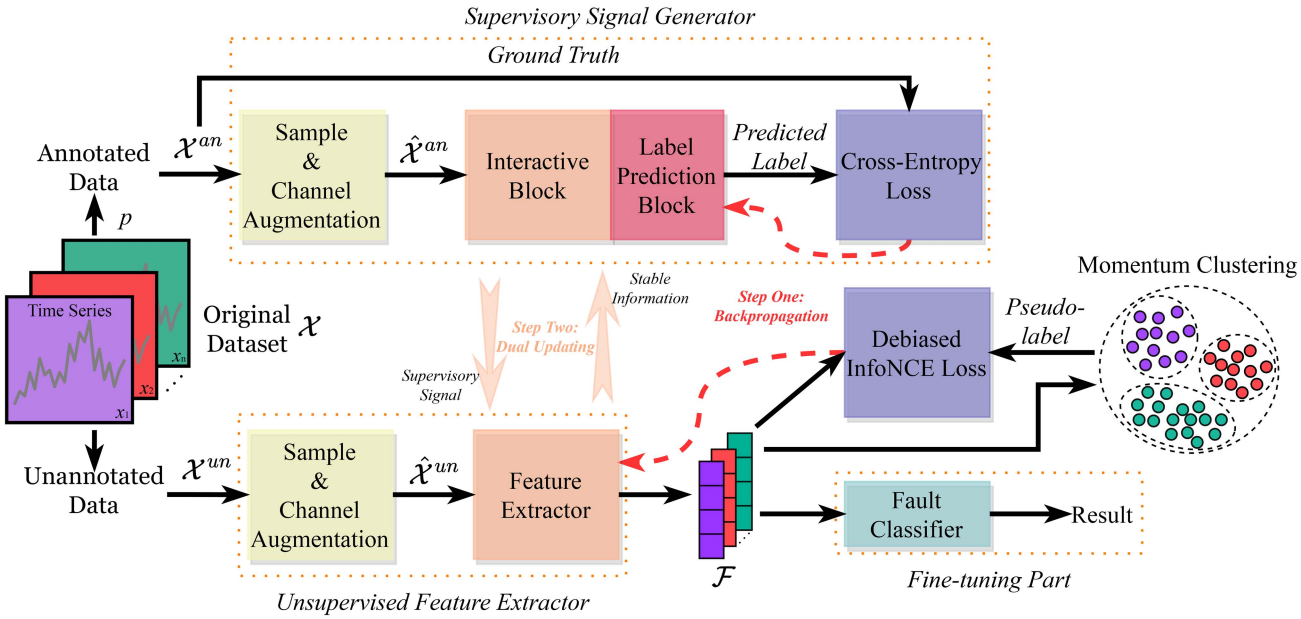

Fig. 1. Model architecture.

Sampling bias is a prominent issue arising from the lack of prior knowledge in selecting positive and negative samples. Typically, only augmented views from the same samples are considered positive, while the rest are regarded as negative. This practice may inadvertently introduce false negative examples during model training, leading to suboptimal representations. Moreover, sampling bias effectively treats each sample as an independent class, causing the representations to be uniformly distributed across the hypersphere [8]. However, this uniform distribution does not match the demands of classification or anomaly detection tasks, where distinct clustered representations in hidden states are desirable. Debiased contrastive learning [9] and contrastive learning with hard negative samples [10] introduce a hyperparameter τ^+ into the InfoNCE loss to mitigate sampling bias. Nonetheless, τ^+ is strongly influenced by the distribution of the dataset and can resemble supervised learning. A practical approach to mitigate sampling bias involves exploring the relation among features and assigning pseudolabels to address this challenge.

Cosine similarity is a widely used metric in contrastive learning [6], [11] measuring the relative angle between representations. It has shown good performance but has a significant limitation. In a polar coordinate system, a specific coordinate is defined by both angle and distance of points. Both independent elements are essential to describe the relative positions of points accurately. However, the cosine similarity metric only considers the relative angle (directional distribution) among features and neglects their distances (radial distribution), which leads to what is known as the angular-centricity issue. This means that cosine similarity is incomplete for capturing the complete information based on the polar coordinate system description. To overcome this limitation, developing a new similarity metric that appropriately accounts for both angular and radial distributions is crucial.

To address the above problems, we proposed a debiased contrastive learning with supervision guidance framework (DCLSG), illustrated in Fig. 1. Channel augmentation is proposed to fuse temporal and frequency information to enhance data mining ability. We make use of an annotated fine-tuning dataset to proliferate a supervisory signal, ensuring that the extracted representations align with the demands of downstream tasks. Compared to traditional two-phase contrastive learning, DCLSG enhances the utilization efficiency of labeled fine-tuning dataset without additional ones. Bidirectional weight updating scheme (BWUS) is proposed to transmit a supervisory signal while maintaining training stability. To combat sampling bias, we introduce momentum clustering, assigning pseudolabels to features, and assisting in determining true positive pairs. In the case of the angular-centricity issue, we develop a novel distance similarity metric known as the Gaussian distance metric (GDM) to control the radial distribution of features. GDM and cosine similarity are independent indicators that jointly describe the relative distance between features. The primary contributions of this article are as follows.

1) A channel augmentation is proposed to incorporate temporal and frequency information, endowing the model to extract features from both local and global perspectives.
2) A novel debiased contrastive learning framework for fault detection named DCLSG is proposed to alleviate task-agnostic issue. DCLSG utilizes the annotated fine-tuning dataset to align the representations with downstream tasks' objectives. BWUS transmits supervisory signal and maintains the stability of the pretraining process.
3) To mitigate sampling bias, momentum clustering is employed to assign pseudolabels to extracted representations, aiding in identifying positive pairs.
4) GDM is proposed to characterize the radial distribution of representations in the polar coordinate system, mitigating

the limitations of cosine similarity. GDM and cosine similarity are independent metrics that jointly depict the relative location among features, alleviating the angular-centricity issue.

The rest of this article is organized as follows. Section II provides the problem definition and an overview of the overall framework. Sections III–V introduce the details of our framework. The experiments are described in Section VI. Finally, Section VII concludes this article.

II. Overall Framework

A. Problem Definition

Given two datasets: Annotated subdataset $\mathcal{X}^{\text{an}} \in \mathbb{R}^{M\times L\times C}$ and unannotated subdataset $\mathcal{X}^{\text{un}} \in \mathbb{R}^{N\times L\times C}$. Samples in $\mathcal{X}^{\text{an}}$ contain the data and label, defined as $\mathcal{X}^{\text{an}} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_M\}$, where $\mathbf{X}_m = (x_m, y_m)$. In contrast, samples in $\mathcal{X}^{\text{un}}$ only contain the data, i.e., $\mathcal{X}^{\text{un}} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$, where $\mathbf{X}_n = (x_n)$. In most cases, $M \ll N$ and we define sampling factor $p = \frac{M}{N}$. Our goal is to build a robust feature extraction framework based on $\mathcal{X}^{\text{an}}$ and $\mathcal{X}^{\text{un}}$, capable of transforming the raw time series samples into representations suitable for downstream tasks.

B. Feature Extraction Framework

The architecture of the proposed framework is depicted in Fig. 1. It begins with the original dataset $\mathcal{X}$, which is initially partitioned into annotated data $\mathcal{X}^{\text{an}}$ with limited p and unannotated data $\mathcal{X}^{\text{un}}$. $\mathcal{X}^{\text{an}}$ also acts as fine-tuning dataset. Subsequently, these datasets are then transformed into augmented samples $\hat{\mathcal{X}}^{\text{an}}$ and $\hat{\mathcal{X}}^{\text{un}}$ through the proposed sample and channel augmentation techniques.

The proposed DCLSG framework contains two primary modules: 1) supervisory signal generator (SSG); and 2) unsupervised feature extractor (UFE). The UFE is the core component of DCLSG framework and is tailored to extract the representation $\mathcal{F}$ from the unannotated dataset $\mathcal{X}^{\text{un}}$ using a self-supervised contrastive learning approach. To address the challenge caused by the lack of prior knowledge and defective distance measurement between features, we introduce two critical modules: 1) the momentum clustering method; and 2) a novel similarity metric termed the GDM. The momentum clustering method assigns pseudolabels to features during model training, enhancing the exploration of intraclass representations within the same category. In addition, the GDM quantifies the relative radial spatial distribution between features, thereby rectifying information discrepancies caused by the cosine similarity. Finally, we craft a new debiased InfoNCE loss combined with the generated pseudolabels and the GDM.

To eliminate the task-agnostic representation issue, we incorporate the supervisory signal into UFE after each training epoch. We train the interactive block under the supervised paradigm driven by $\mathcal{X}^{\text{an}}$ and treat the parameters of the interactive block as available supervisory signal. Furthermore, we introduce a new information transmission strategy called the BWUS. In contrast to conventional exponential moving average method, which only supports unidirectional information transfer, BWUS allows for bidirectional information transfer between the feature extractor and the interaction block. BWUS not only ensures training stability during the information transfer process, but also guarantees that both SSG and UFE share the same optimization objectives, thereby avoiding a mismatch between the pretraining and fine-tuning phases.

Algorithm 1: Pretraining Phase of Proposed DCLSG Framework.

Require: The annotated sub-dataset $\mathcal{X}^{an}$, the unanotated sub-dataset $\mathcal{X}^{un}$, data augmentation operation $\mathcal{T}(\cdot)$, feature extractor $f^{UFE}(\cdot)$, interactive block $f^{SSG}(\cdot)$, and label prediction block $p(\cdot)$, momentum clustering operation $mc(\cdot)$, weighting factors α and β.

Ensure: Feature extractor $f^{UFE}(\cdot)$

for epc **in** epoch **do**
  for batch $\mathcal{X}_B^{an}$, ground truth $\mathcal{Y}_B^{gt}$ **in** $\mathcal{X}^{an}$ **do**
    Get data augmentation as (1)-(5): $\hat{\mathcal{X}}_B^{an} = \mathcal{T}(\mathcal{X}_B^{an})$;
    Get predicted labels: $\mathcal{Y}_B^{prd} = p(f^{SSG}(\hat{\mathcal{X}}_B^{an}))$;
    Substitute $\mathcal{Y}_B^{prd}$ and $\mathcal{Y}_B^{gt}$ into Cross-Entropy loss to calculate $\mathcal{L}^{SSG}$;
    Update $f^{SSG}(\cdot)$ and $p(\cdot)$ to minimize $\mathcal{L}^{SSG}$;
  end for
  for batch $\mathcal{X}_B^{un}$ **in** $\mathcal{X}^{un}$ **do**
    Get data augmentation as (1)-(5): $\hat{\mathcal{X}}_B^{un} = \mathcal{T}(\mathcal{X}_B^{un})$;
    Get feature batch: $\mathcal{F}_B = f^{UFE}(\hat{\mathcal{X}}_B^{un})$;
    Get pseudo-labels via Alg. 2: $\mathcal{C}_B^{psd} = mc(\mathcal{F}_B)$;
    Substitute $\mathcal{F}_B$ and $\mathcal{C}_B^{psd}$ into (15) or (16) to calculate debiased InfoNCE loss $\mathcal{L}_1$ or $\mathcal{L}_2$;
    Update $f^{UFE}(\cdot)$ to minimize $\mathcal{L}_1$ or $\mathcal{L}_2$;
  end for
  Substitute $f^{SSG}(\cdot)$, $f^{UFE}(\cdot)$, α, and β into (17) and (18) to operate BWUS between $f^{SSG}(\cdot)$ and $f^{UFE}(\cdot)$;
end for
return Feature extractor $f^{UFE}(\cdot)$

The pseudocode for the pretraining phase of the proposed DCLSG framework is illustrated in Algorithm 1.

III. Data Augmentation

In this article, we introduce two kinds of augmentation methods: 1) sample augmentation; and 2) channel augmentation. Sample augmentation generates various views of the original samples through weak and strong augmentations, encouraging the model to extract invariant features. Channel augmentation incorporates the information from both temporal and frequency domains, endowing the model with multifaceted feature extraction capabilities.

A. Sample Augmentation

We generate weak augmented views x_w and strong augmented views x_s from the original sample x through jittering and permutation, as shown in Fig. 2(a). Strong augmentation involves

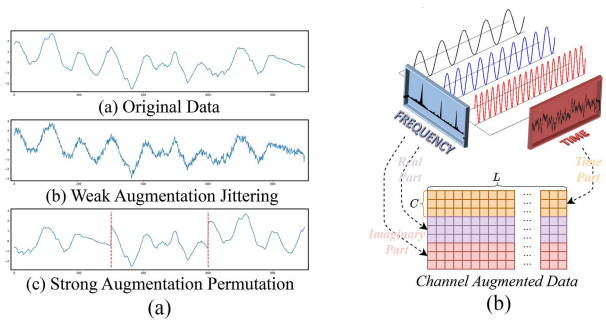

Fig. 2. Data augmentation. (a) Sample augmentation. (b) Channel augmentation.

disrupting the chronological order of time series, which can significantly impact the original characteristics. In contrast, weak augmentation introduces subtle variations to the time series without significantly altering their overall shape. Combining weak and strong sample augmentation allows the model to explore intrinsic information in the original data.

In DCLSG, we create two data subsets, $\mathcal{X}^{\text{an}}$ and $\mathcal{X}^{\text{un}}$, by removing the labels from $\mathcal{X}$ and sampling p percent of the annotated data, respectively. These subsets are then transformed into $\dot{\mathcal{X}}^{\text{an}}$, $\dot{\mathcal{X}}_w^{\text{un}}$, and $\dot{\mathcal{X}}_s^{\text{un}}$ via sample augmentation

$$\dot{\mathcal{X}}^{\text{an}} = \mathcal{C}(\mathcal{X}^{\text{an}}, \mathcal{T}_w(\mathcal{X}^{\text{an}}), \mathcal{T}_s(\mathcal{X}^{\text{an}})) \tag{1}$$

$$\dot{\mathcal{X}}_w^{\text{un}} = \mathcal{T}_w(\mathcal{X}^{\text{un}}) \tag{2}$$

$$\dot{\mathcal{X}}_s^{\text{un}} = \mathcal{T}_s(\mathcal{X}^{\text{un}}) \tag{3}$$

where $\dot{\mathcal{X}}^{\text{an}} \in \mathbb{R}^{3B*p\times L\times C}$ is the full augmented data of $\mathcal{X}^{\text{an}}$, while $\dot{\mathcal{X}}_w^{\text{un}} \in \mathbb{R}^{B\times L\times C}$ and $\dot{\mathcal{X}}_s^{\text{un}} \in \mathbb{R}^{B\times L\times C}$ are the weakly and strongly augmented views of $\mathcal{X}^{\text{un}}$, respectively. B stands for the batch size, and L and C denote the length and channel number of a sample, respectively. $\mathcal{C}(\cdot)$ denotes a concatenation operation, and $\mathcal{T}_w(\cdot)$ and $\mathcal{T}_s(\cdot)$ represent weak and strong sample augmentation operations, respectively.

B. Channel Augmentation

In signal analysis, temporal and frequency domains offer distinct perspectives for describing signals while preserving the same semantics. The temporal domain primarily captures the local and dynamic characteristics, highlighting rapid changes between adjacent points. In contrast, the frequency domain focuses on the orthogonal pattern decomposition of macrotime series, enabling the extraction of global and stable modes.

To seamlessly integrate information from the temporal and frequency domains, we propose a channel augmentation technique that empowers the model to extract features from local and global perspectives. In frequency information with complex form, the real part represents the cosine component, while the imaginary part denotes the sine component of the time series at a specific frequency component. We separate these real and imaginary components as new channels and combine them with the original time series data to create channel-augmented data, as depicted in Fig. 2(b). This operation expands the dataset dimension from $\mathbb{R}^{B\times L\times C}$ to $\mathbb{R}^{B\times L\times 3C}$ through the following process:

$$\mathcal{R} + \mathcal{I}j = FFT(\dot{\mathcal{X}}) \tag{4}$$

$$\hat{\mathcal{X}} = \mathcal{C}(\dot{\mathcal{X}}, \mathcal{R}, \mathcal{I}) \tag{5}$$

where $\mathcal{R}$ and $\mathcal{I}$ represent the real and imaginary parts of the frequency, respectively, and $FFT(\cdot)$ denotes the fast Fourier transform operator. After the sample and channel augmentation operation, we obtain $\hat{\mathcal{X}}^{\text{an}} \in \mathbb{R}^{3B*p\times L\times 3C}$, $\hat{\mathcal{X}}_w^{\text{un}} \in \mathbb{R}^{B\times L\times 3C}$, and $\hat{\mathcal{X}}_s^{\text{un}} \in \mathbb{R}^{B\times L\times 3C}$. $\hat{\mathcal{X}}^{\text{un}}$ is represented as $[\hat{\mathcal{X}}_w^{\text{un}}; \hat{\mathcal{X}}_s^{\text{un}}]$.

IV. Debiased InfoNCE Loss

A. Traditional InfoNCE Loss

The InfoNCE loss is a fundamental component that enhances the similarity of positive pairs, allowing them to capture common and invariant information [3]. At the same time, it encourages the features from randomly sampled negative pairs to become more distinct. The classical InfoNCE loss function is defined in (6)

$$\mathcal{L}_{\text{InfoNCE}} = \mathbb{E}\left[-\log \frac{e^{\text{sim}(f,f^+)/\tau}}{e^{\text{sim}(f,f^+)/\tau} + \sum_{n\in N(i)} e^{\text{sim}(f,f_n^-)/\tau}}\right] \tag{6}$$

where f is the extracted features, f^+ and f^- are positive and negative features of f, respectively. $sim(\cdot,\cdot)$ denotes the similarity metric, τ is the temperature coefficient, $N(i)$ is the set of negative features of f, and n is the index of $N(i)$.

In the absence of prior knowledge, establishing positive and negative pairs can be a challenging task. The current approach defines two views augmented from the same sample as a positive pair, and treats view from different samples as negative. However, this strategy overlooks the meaningful relation among samples and treats each sample as an independent class. Consequently, the features of samples from the same class end up being uniformly distributed within the hidden state. Unfortunately, a uniform distribution of features is suboptimal for classification or fault detection tasks.

In contrastive learning, cosine value is a widely used metric to gauge the similarity between features, defined as follows:

$$\cos(f_1, f_2) = \frac{f_1^T f_2}{\|f_1\|_2 \|f_2\|_2} \tag{7}$$

where f_1 and f_2 are features, and $\|\cdot\|_2$ represents the ℓ_2 norm.

As depicted in Fig. 3, there exists a clear mathematical relationship between the angle θ_1 and the distance d_1' of the projection point F_1' and F'

$$\cos(f_1', f') = \cos\theta_1 = 1 - \frac{\|d_1'\|_2^2}{2}. \tag{8}$$

However, the cosine similarity neglects the distance of the features f, f_1, and f_2, projecting them into a unit polar coordinate system to generate f', f_1', and f_2', which is a lossy projection method.

In a polar coordinate system, angle and distance are essential for pinpointing a location. Similarly, to describe the relative

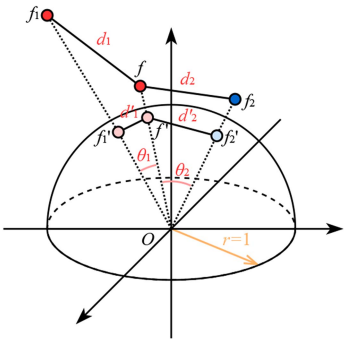

Fig. 3. Feature distribution in hidden state. f_1 and f_2 are the positive and negative feature for feature f, O is the origin of hidden state. f', f_1', and f_2' are the projection points of f, f_1, and f_2 in unit hypersphere [8]. θ and d are the angle and distance between features.

positions among multiple points, considering the relative angles and distances between features is of paramount importance. Angles represent the directional distribution among features, and distances describe the radial distribution. In summary, it is necessary to construct a new metric to measure radial distribution.

B. Momentum Clustering

Traditional InfoNCE loss considers each sample an independent class and neglects the shared characteristics among samples with the same ground truth. The projections of extracted features on the unit hypersphere are uniform distribution, which do not match the demand of time series classification.

Building upon the principles of supervised contrastive learning [4], which integrates ground truth labels into the InfoNCE loss for learning shared characteristics among samples of the same class, we extend traditional InfoNCE loss. In cases where prior knowledge is lacking, we assign pseudolabels to each sample by feature clustering operation. This approach allows us to leverage pseudolabels to facilitate learning common characteristics through an improved InfoNCE.

Momentum clustering plays a crucial role in generating pseudolabels by considering the distances between features and clustering centroids. This approach leverages the clustering centroids from the previous epoch as the initial values. It involves calculating the distances among feature and clustering centroids, assigning labels to features based on distances, and subsequently applying momentum-based updating to establish new centroids as the initial values for the next epoch. This momentum-based updating of cluster centroids, as depicted in (9), is instrumental in maintaining the stability of the training process. The concrete algorithm of momentum clustering is shown in Algorithm 2.

To measure the spatial distance between feature and centroid, the ℓ_2 norm is utilized, as expressed in (10).

$$ctr\_next = ctr\_prev * \eta + ctr\_curr * (1 - \eta), \tag{9}$$

$$distance(f_i, c_j) = \|f_i - c_j\|_2 \tag{10}$$

Algorithm 2: Pseudolabel Generation through Momentum Clustering.

Require: Feature set $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$, number of clusters K, updating momentum factor η, clustering centriods ctr_curr

Ensure: Cluster assignment lb_pred and updated clustering centroids ctr_next

```
if ctr_curr == {φ} then
    Randomly initialize K cluster centroids: ctr_curr;
end if
Define temporary variables: ctr_prev = {φ},
 lb_new = {0}, lb_old = {φ};
while lb_new == lb_old do
    Save old labels: lb_old = lb_new;
    for i in range(1, m) do
        Calculate the index of nearest centroid for feature
        f_i via (10):
        k_i^* = arg min_k distance(f_i, ctr_curr_k);
        Assign the pseudo-label to f_i: lb_new_i = k_i^*;
    end for
    Momentum update clustering centroids via (9):
    ctr_prev = ctr_curr;
    for k in range(0, K) do
        ctr_curr_k = 1/|F_k| Σ_{f_i ∈ F_k} f_i
        where F_k is the set of features in class k, |F_k|
        denotes the number of features in |F_k|;
        ctr_curr_k =
        ctr_prev_k * η + ctr_curr_k * (1 − η);
    end for
end while
Save results: lb_pred = lb_new,
ctr_next = ctr_curr;
return Cluster assignment lb_pred and clustering
centroids ctr_next
```

where ctr_next, ctr_prev, and ctr_curr donate the clustering centroids in the next, previous, and current training batch, respectively. Meanwhile, f_i and c_j represent the ith feature and jth centroid. η is the updating momentum factor.

C. Gaussian Distance Metric (GDM)

Cosine value is a broadly employed metric for quantifying the similarity between features. However, the cosine similarity only considers the directional relationship of features while ignoring their radial distribution, shown in Fig. 3. To describe the radial distribution of features, we propose GDM to measure relative location in radial. The GDM quantifies the ratio between the distances of features and shares a similar convexity property with the cosine function, making it well-suited for gradient optimization.

When evaluating the GDM for f_j with respect to f_i, we introduce the residual distance ratio $\lambda_{i,j}$, as expressed in (11), to facilitate a more meaningful comparison of the radial positions of f_i and f_j. Subsequently, we apply a Gaussian function to map the range of $\lambda_{i,j}$ from -1 to $+\infty$ into the range of $GDM(f_i, f_j)$

from 0 to 1. This mapping helps to align the order and range of $GDM(f_i, f_j)$ with $\cos(f_i, f_j)$, simplifying the process of balancing their respective weights.

$$\lambda_{i,j} = \frac{\|f_j\|_2 - \|f_i\|_2}{\|f_i\|_2}, \tag{11}$$

$$GDM(f_i, f_j) = GDM(\lambda_{i,j}) = \exp\left(-\frac{\lambda_{i,j}^2}{2\sigma^2}\right). \tag{12}$$

Furthermore, based on the derivation process in (13), it can be observed that the ratio of the first derivatives of $GDM(f_i, f_j)$ and $\cos(f_i, f_j)$ within the neighborhood of the optimization endpoint remains close to $1/\sigma^2$ in which σ is the factor to adjust the smoothness of GDM. This illustrates that the two functions exhibit similar convexity in the vicinity of the optimization focus, further streamlining the utilization of a single optimizer for collaborative optimization.

$$\lim_{f_j \to f_i} \frac{GDM'(f_i, f_j)}{\cos'(f_i, f_j)} = \lim_{\substack{\lambda_{i,j}\to 0 \\ \theta_{i,j}\to 0}} \frac{GDM'(\lambda_{i,j})}{\cos'\theta_{i,j}}$$

$$= \lim_{\substack{\lambda_{i,j}\to 0 \\ \theta_{i,j}\to 0}} \frac{-\exp(-\frac{\lambda_{i,j}^2}{2\sigma^2})\frac{\lambda_{i,j}}{\sigma^2}}{-\sin\theta_{i,j}} = \lim_{\epsilon_{i,j}\to 0} \frac{-\exp(-\frac{\epsilon_{i,j}^2}{2\sigma^2})\frac{\epsilon_{i,j}}{\sigma^2}}{-\sin\epsilon_{i,j}} = \frac{1}{\sigma^2} \tag{13}$$

where $f_j \to f_i$ can be separated as two independent processes $\lambda_{i,j} \to 0$ and $\theta_{i,j} \to 0$ from the perspective of polar coordinate system. $\epsilon_{i,j}$ is used to replace the $\lambda_{i,j}$ and $\theta_{i,j}$ to represent their simultaneous approximation to zero.

Once we have obtained a suitable metric for quantifying the radial distribution, we introduce two strategies of adapted debiased InfoNCE to fuse cosine and GDM similarity as (14)–(16)

$$sim(f_i, f_j) = \gamma * \cos(f_i, f_j) + (1-\gamma) * GDM(f_i, f_j), \tag{14}$$

$$\mathcal{L}_1 = \frac{1}{|I|}\sum_{i\in I}\frac{-1}{|P(i)|}\sum_{p\in P(i)}\log\frac{e^{\text{sim}(f_i, f_p^+)/\tau}}{\sum_{n\in N(i)} e^{\text{sim}(f_i, f_n^-)/\tau}}, \tag{15}$$

$$\mathcal{L}_2 = \frac{1}{|I|}\sum_{i\in I}\frac{-1}{|P(i)|}\sum_{p\in P(i)}\left[\gamma * \log\frac{e^{\cos(f_i, f_p^+)}}{\sum_{n\in N(i)} e^{\cos(f_i, f_n^-)}} + (1-\gamma) * \log\frac{e^{GDM(f_i, f_p^+)}}{\sum_{n\in N(i)} e^{GDM(f_i, f_n^-)}}\right] \tag{16}$$

where I is the set of features, $P(i)$ is the set of positive features for f_i. $|\cdot|$ denotes the number of elements. γ is the weight factor between cosine and GDM similarity.

$\mathcal{L}_1$ and $\mathcal{L}_2$ employ distinct optimization strategies for cosine and GDM similarity. Specifically, $\mathcal{L}_1$, as expressed in (14)–(15), combines cosine and GDM into a novel similarity metric $sim(f_i, f_j)$ and directly optimizes this integrated metric. In contrast, $\mathcal{L}_2$, as outlined in (16), adopts a separate optimization strategy for cosine and GDM, subsequently fusing their contributions based on the weight factor γ. The choice of $\mathcal{L}_1$ and $\mathcal{L}_2$ should be based on the relative difficulty of optimizing the two similarities, cosine and GDM. If the optimization paths of cosine and GDM are similar from the beginning to the end of the process, then $\mathcal{L}_1$ can better balance the two by combining them into a single metric. Conversely, if the optimization paths of cosine and GDM differ significantly, $\mathcal{L}_2$ is needed to optimize cosine and GDM independently and then merge their results. The relative optimization difficulty of cosine and GDM depends on various dataset characteristics.

V. Bidirectional Weight Updating Scheme

In a traditional contrastive learning framework, a small amount of labeled data is utilized during the fine-tuning phase to map extracted features to specific downstream tasks. Given the condition above, we flexibly utilize the annotated fine-tuning dataset to create supervisory signal in the pretraining phase without additional labeled data. This signal serves as guidance for the feature extractor, ensuring it is trained following the requirements of the downstream task.

We propose a novel framework, illustrated in Fig. 1, to generate and transfer effective supervisory signal. This framework consists of two main components: 1) SSG; and 2) UFE, both of which are trained parallel and independently. SSG comprises an interactive block and a label prediction block, which are driven by augmented annotated $\mathcal{X}^{\text{an}}$, i.e., fine-tuning dataset. SSG engages in supervised learning directly connected to downstream tasks to create the required supervisory signal. UFE conducts adapted debiased cluster-wise contrastive learning to extract robust representations. The interactive block and feature extractor share the same architecture, facilitating seamless information exchange. After being individually trained, BWUS processes bidirectional information transfer between these components.

After the individual training of the SSG and UFE, BWUS facilitates bidirectional information transfer between these components. BWUS treats the weights of the interactive block and feature extractor as supervisory signal and stable information, respectively. Subsequently, both the interactive block and feature extractor undergo weight updates simultaneously after each training epoch, as illustrated in (17)–(18). During BWUS, supervisory signal is transmitted to feature extractor to align the extracted features with the objectives of downstream tasks. Simultaneously, stable information is transmitted in reverse to the interactive block to prevent overfitting and mitigate parameter dispersion between the SSG and UFE.

$$\hat{\mathcal{P}}_i^{\text{gen}} = \alpha * \mathcal{P}_i^{\text{gen}} + (1-\alpha) * \mathcal{P}_i^{\text{ext}}, i = 1, 2, 3, \ldots, n \tag{17}$$

$$\hat{\mathcal{P}}_i^{\text{ext}} = \beta * \mathcal{P}_i^{\text{ext}} + (1-\beta) * \mathcal{P}_i^{\text{gen}}, i = 1, 2, 3, \ldots, n \tag{18}$$

where $\mathcal{P}_i^{\text{gen}}$ and $\mathcal{P}_i^{\text{ext}}$ are the ith corresponding parameters in the interactive block and feature extractor, respectively. $\hat{\mathcal{P}}_i^{\text{gen}}$ and $\hat{\mathcal{P}}_i^{\text{ext}}$ are the updated parameters. α and β are the weighting factors, n denotes the number of parameters.

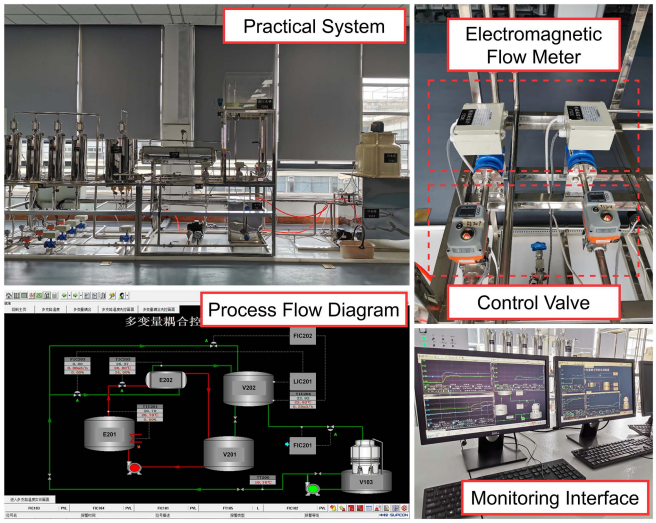

Fig. 4. Real hardware experiment system.

VI. Experiments and Results

A. Dataset Description

Two widely used time series datasets are employed to demonstrate the performance of the DCLSG framework. The details of the datasets are as follows.

1) *ISDB dataset:* The international stiction data base (ISDB), as described by [12], is widely recognized as a benchmark for validating new methods related to the control loop performance assessment. These datasets comprise control loops obtained from diverse process industries, including chemical plants, pulp and paper mills, buildings, mining, and power plants. In our article, we approach the fault detection task as a classification problem, where the goal is to determine whether a given control loop exhibits stiction.
2) *CWRU dataset:* This dataset, obtained from the Case Western Reserve University (CWRU) bearing data center, comprises vibration signals collected from various sensors, including drive-end accelerometer data, fan-end accelerometer data, and base accelerometer data with a 12 kHz sampling rate. CWRU dataset was conducted on different fault diameters bearing (specifically 7, 14, and 21 miles), and each fault diameter was associated with three fault types, including inner race defect, outer race defect, and ball defect. In this article, we test the CWRU dataset with ten classes.
3) *Practical dataset:* This dataset contains the control loop signals collected from hardware experimental system and actual industrial environment. In Fig. 4, the hardware experimental system includes a liquid level control loop (LIC201) and two flow control loops (FIC201 and FIC202), simulating a coupled heat exchange process in chemical engineering. In the process flow diagram, cold water from vessel V103 is transferred to heat exchanger E202 and vessel V202. Heater E201, E202, and vessel V201 form a heating loop. E202 raise the temperature of cold water by 10° before transferring the hot water to V202. In vessel V202, hot and cold water are mixed in vessel V202 to achieve temperature control. Finally, water in V202 reflows to V103. LIC201 controls the water level of V202, FIC201 controls the flow of reflow water from V202 to V103, and FIC202 controls the cold water flow from V103 to V202. The three other control loops, PIC23002, FIC3107, and F6304, are collected from practical industrial environments, in which PIC23002 is a pressure control loop, and it is affected by unknown external disturbances, FIC3107 is a flow control loop and its state is normal, and F6304 is flow control loop.

B. Experiment Settings

For the CWRU dataset, our experiment involved constructing a dataset consisting of one normal baseline condition and nine randomly selected fault conditions. Under each condition, We randomly sampled 450 time series samples. Each sample had a length of 512 and a dimension of 2. To simulate real-world scenarios where obtaining extensive data may be challenging, we limited the number of samples in our train subdatasets. Specifically, we randomly selected 50 samples from each class for the train subdataset $\mathcal{X}$, resulting in 500 samples. The remaining 400 samples from each class were assigned to the test subdataset, resulting in 4000 samples.

For the ISDB dataset, we selected a total of 85 control loops with available data. We randomly sampled 60 time series samples from each control loop, each of which had a length of 800 and a dimension of 2. As a result, our final dataset consisted of a total of 5100 samples. To facilitate meaningful comparisons with other stiction detection methods, we selected 26 control loops corresponding to 1560 samples as the test subdataset. The remaining 59 control loops (3540 samples) were assigned as the training subdataset $\mathcal{X}$.

For the practical dataset, we randomly sampled 21 subseries from each control loop signal. Each time series has a length of 512 and a dimension of 2. The practical dataset includes various working conditions such as normal, external distribution, and stiction. However, our experiments focus solely on detecting the stiction state of control loops, i.e., there are only no-stiction (Nonstic) and stiction (Stic) labels. We utilize the practical dataset to compare and validate the generalization and robustness of models in real industrial deployment. Specifically, the models are trained in ISDB dataset due to its diverse data source and then used to predict the states of samples from the practical dataset. Finally, we determine the state of each control loops through the voting results of the 21 subseries.

In our experimental setup, we employed random sampling to create two subsets from the training subdataset $\mathcal{X}$. The annotated subset $\mathcal{X}^{\text{an}}$ consists of a randomly selected p portion from $\mathcal{X}$ with labels to serve as the foundation of generating the supervisory signal. The second subset, denoted as $\mathcal{X}^{\text{un}}$, encompasses the

entire $\mathcal{X}$ but with label masked. $\mathcal{X}^{un}$ is utilized to extract effective features for downstream tasks.

Two-layers temporal convolutional network (TCN) [13] with kernel size 4 and hidden dimension 128 is employed as the specific structure of interactive block and feature extractor. label prediction block and fault classifier are two-layer multilayer perceptrons with hidden dimension 128. The number of momentum clustering centroids is equal to the class number of the dataset. The hyperparameters settings are as follows: $\eta = 0.8$, $\sigma = 1.0$, $\alpha = 0.2$, $\beta = 0.4$, and $\gamma = 0.2$.

C. Evaluation Metrics

We adopt average accuracy (Acc) as our primary metric to evaluate the model performance; weighted recall ($w{-}Recall$) and weighted F1 score ($w{-}F1$) are also employed as auxiliary assessments.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}, \tag{19}$$

$$Prec_i = \frac{TP_i}{TP_i + FP_i}, \quad Recall_i = \frac{TP_i}{TP_i + FN_i}, \tag{20}$$

$$w{-}Recall = \sum_{i=1}^{C} w_i \cdot Recall_i = \sum_{i=1}^{C} \frac{n_i}{N} \cdot Recall_i, \tag{21}$$

$$w{-}F1 = \sum_{i=1}^{C} w_i \cdot F1_i = \sum_{i=1}^{C} \frac{n_i}{N} \cdot \frac{2 \times Prec_i \times Recall_i}{Prec_i + Recall_i} \tag{22}$$

where TP and TN represent the numbers of true positive and true negative, FP and FN indicate the numbers of false positive and false negative, respectively. Subscript i denotes the attribute of class c_i, C and N are the number of classes and samples, respectively. w_i and n_i signify the proportion factor and number of samples with ground truth c_i.

Acc measures the percentage of correctly classified instances out of the total instances, providing a straightforward evaluation metric. However, its suitability diminishes in imbalanced datasets, where one class significantly outweighs the others. For a more comprehensive assessment of model performance, we introduce $w{-}Recall$ and $w{-}F1$ metrics tailored to address imbalanced datasets.

$w{-}Recall$ evaluates the capacity of the model to identify all positive instances while accounting for the varying number of instances in each class. On the other hand, $w{-}F1$ is the harmonic mean of precision and recall and also adapts the imbalanced dataset. These metrics offer a nuanced perspective on model performance, particularly in scenarios with imbalanced class distributions, providing a more insightful evaluation.

D. Detection and Comparison Results

This study first presents the detection results of the ISDB dataset, evaluating the model's performance revolves around determining loop stiction using the Acc metric. The state of the entire control loop is determined through a voting mechanism based on the classification results of each sample within the same loop. The numbers enclosed in brackets denote the count of untested loops out of 26 loops. The comparative results are presented in Table I alongside thirteen other fault detection algorithms. The evaluation encompassed 26 control loops, consistent with previous methods. In Table I, DCLSG attains Acc of 0.8461 (22/26 loops) in the ISDB dataset, showcasing the best performance. Compared with traditional methods based on statistical methods, such as curve fitting, peak slope, and relay-based methods, DCLSG demonstrates good suitability for a wide range of industrial loop data. Furthermore, comparison with other methods, such as the BSD-CNN method, D-value ANN method, and MTCNN, all of which employ deep learning techniques, reveals that our method achieves a higher Acc. Our approach matches the previous best model MTFCC's performance, yielding a total detection Acc of 0.8461 (22/26 loops), representing the highest among the considered methods. Meanwhile, MTFCC is an image processing algorithm that uses two channels to create one relative position diagram and extract features from it. However, with more than two channels, the number of relative position diagrams increases exponentially, limiting the deployment of the MTFCC algorithm. In contrast, DCLSG directly utilizes time series data to extract relevant features for fault detection tasks, making it more suitable for industrial deployment.

TABLE I
COMPARISON RESULTS FOR THE ISDB AND CWRU

ISDB dataset		CWRU dataset	
Method	Acc	Method	Acc
Statistics method [12]	0.6400 (1)	MSN [14]	0.948
Curve fitting method [15]	0.4800 (1)	LeNet-5 [16]	0.943
Peak slope method [17]	0.5600 (1)	DBN [18]	0.643
Zone Segmentation method [17]	0.6000 (2)	SVM [19]	0.892
D-value ANN method [20]	0.7917 (0)	IMSN [21]	0.954
Relay-based method [22]	0.6538 (0)	sdiAE [23]	0.961
Waveform shape analysis [24]	0.4231 (0)	RAE [25]	0.924
PSD/ACF method [26]	0.6923 (0)	IPDL [27]	0.954
BSD-CNN method [28]	0.7692 (0)	LSTM [16]	0.956
MTCNN [5]	0.8076 (0)	PGCNN [29]	0.982
RAE [25]	0.6538 (0)	UDWGAN [30]	0.989
IPDL [27]	0.8076 (0)	MobileNet [31]	0.990
MTFCC [1]	**0.8461 (0)**	**DCLSG (ours)**	**0.996**
DCLSG (ours)	**0.8461 (0)**	-	-

For the CWRU dataset, we conducted a comparative analysis against the existing twelve fault detection methods. The results are showcased in the right part of Table I. IMSN, MSN, and LeNet-5 were initially designed for classification tasks and can thus be directly applied to fault detection tasks. From the results, it can be seen that our method, DCLSG, achieves the highest detection Acc of 0.996. This outperformance, especially under the condition of utilizing the same annotated data, is substantial compared to other fault detection methods. Compared to the last state-of-the-art methods (sdiAE, PGCNN, UDWGAN, and MobileNet), DCLSG also achieves better performance.

It is worth noting that MobileNet, despite being a lightweight network, has 4.2 million parameters, whereas the core part of DCLSG for deployment (feature extractor and fault classifier)

TABLE II
DEPLOYMENT RESULTS ON THE PRACTICAL DATASET

Method	PIC23002	FIC3107	F6304	FIC201	FIC202	LIC201
Ground Truth	External Distribution	Normal	Stiction	Normal	Normal	Normal
SVM [32]	Nonstic	Nonstic	Nonstic	Stic	Nonstic	Nonstic
Rand Forest [33]	Nonstic	Nonstic	Nonstic	Stic	Nonstic	Nonstic
XgBoost [34]	Nonstic	Nonstic	Stic	Stic	Nonstic	Nonstic
LeNet-5 [35]	Stic	Stic	Stic	Nonstic	Stic	Nonstic
MTCNN [5]	Nonstic	Stic	Stic	Nonstic	Nonstic	Nonstic
MTFCC [1]	Nonstic	Nonstic	Stic	Nonstic	Nonstic	Stic
DCLSG (ours)	Nonstic	Nonstic	Stic	Nonstic	Nonstic	Nonstic

has only 0.4 million parameters. The larger number of parameters in MobileNet occupies more computing resources. It is essential to emphasize that our ultimate goal is not solely to achieve 100% detection Acc but rather to demonstrate the effectiveness of our method, particularly when labeled samples are scarce. Our experimental results underscore the advantages of our approach under such conditions.

To evaluate the applicability of DCLSG in actual industrial deployment, we directly applied the model trained on the ISDB dataset to detect stiction state of control loops in the practical dataset without fine-tuning. The detection results are presented in Table II, where the underlined entries indicate false detection results. Compared with statistic-based methods (SVM, Random Forest, and XgBoost) and deep learning-based methods (LeNet-5, MTCNN, and MTFCC), DCLSG correctly detects the stiction state of all loops. The statistic-based, including SVM, Rand Forest, and XgBoost, have a solid theoretical foundation, resulting in wide applications in industry. Meanwhile, the deep learning-based methods, including LeNet-5, MTCNN, and MTFCC, are the newest valve detection algorithms and have performed well in valve stiction detection tasks. The comparative results verify the feasibility and robustness of the proposed DCLSG framework in real industrial deployment.

DCLSG has demonstrated outstanding performance across two industrial scenarios, valve stiction and bearing failure, as well as in practical industrial deployment. These diverse applications highlight the framework's robustness and versatility in addressing various types of industrial faults.

E. Ablation Study

The proposed DCLSG framework consists of four main modules: channel augmentation (CA), supervisory signal (SS), pseudolabel (PL), and GDM. In this subsection, we aim to demonstrate the effectiveness of each module through ablation experiments conducted on both the CWRU and ISDB datasets. Table III presents experiments conducted under different modules based on the Basic contrastive learning method (Basic), discussing their specific contributions to the framework. The selection of loss functions $\mathcal{L}_1$ and $\mathcal{L}_2$ is dataset-specific, and the rationale behind this choice is based on the actual effect.The metrics in bold represent the highest values, while the underlined metrics denote the second-highest values in Table III. To comprehensively describe the performance of DCLSG on the ISDB dataset, we deviate from the experiments in Section VI-D. Instead, we collect the results for each sample without aggregating the results for loops through voting.

1) Contribution of CA: This study initially validates the effectiveness of the proposed CA using the CWRU and ISDB datasets. The experiments were conducted with the Basic contrastive learning model as a baseline, and the results can be found in Table III under the Basic and Basic+CA columns. The findings suggest that Basic+CA significantly enhances the performance of the Basic model, with an average Acc, $w{-}Recall$, and $w{-}F1$ improvement of 0.049, 0.028, 0.038 in CWRU and 0.041, 0.031, 0.045 in ISDB. Furthermore, this improvement remains stable regardless of the value of p. Based on these detection results, CA proves to be an effective technique, seamlessly integrating temporal and frequency information and empowering the model to extract features from both local and global perspectives.

2) Contribution of SS: This study further validates the effectiveness of the SS. Experiments were conducted using the Basic contrastive learning model with CA (Basic+CA) as a baseline, and the results are presented in Table III under the Basic+CA and Basic+CA+SS columns. On the CWRU dataset, SS notably enhances the time series classification capabilities, boosting max Acc, $w{-}Recall$, and $w{-}F1$ from 0.495, 0.492, 0.429 to 0.997, 0.997, 0.997. The improvements in the ISDB dataset are also notable. In summary, the SS effectively addresses the task-agnostic representation issue and strengthens the connection between the pretraining and fine-tuning phases without increasing the reliance on annotated data compared to Basic contrastive learning methods.

3) Contribution of PL: This study validates the effectiveness of PL. Experiments were conducted using Basic+CA+SS as a baseline, and the results are presented in Table III under the Basic+CA+SS and Basic+CA+SS+PL columns. For the CWRU dataset, PL notably enhances the Acc, $w{-}Recall$, and $w{-}F1$ values by approximately 0.150 when $p \leq 0.6$. However, the metrics are slightly lower than Basic+CA+SS by about 0.020 when $p > 0.6$. PL significantly improves performance when there is a lack of SS. When p is larger, the SS dominates the performance improvement, and PL, as an additional contrastive, increases the instability of the model, potentially generating false PL that may influence the model. The efficacy of DCLSG on the ISDB dataset mirrors its behavior on the CWRU dataset. In summary, although PL may increase the instability of the model due to its inherent inaccuracies, it can significantly enhance the performance of the model under the influence of low SS.

4) Contribution of GDM: This study conclusively validates the effectiveness of GDM. Independent effect experiments (Basic+CA+SS & Basic+CA+SS+GDM) and joint effect experiments (Basic+CA+SS+PL & Basic+CA+SS+PL+GDM) in Table III were conducted to investigate the influence of GDM from different perspectives. In independent effect experiments, GDM primarily plays a role under low supervision signals ($p \leq 0.6$) in both datasets. In the joint effect experiment, the synergy of PL and GDM takes full advantage of both.

TABLE III
DETECTION RESULTS OF DIFFERENT MODULE EFFECTIVENESS

Dataset	p	Basic			Basic+CA			Basic+CA+SS		
		Acc	*w-Recall*	*w-F1*	*Acc*	*w-Recall*	*w-F1*	*Acc*	*w-Recall*	*w-F1*
CWRU $\mathcal{L}_2$	0.1	0.214±0.032	0.259±0.050	0.181±0.039	0.240±0.035	0.283±0.048	0.201±0.035	0.330±0.043	0.391±0.058	0.301±0.045
	0.2	0.239±0.026	0.290±0.053	0.202±0.030	0.282±0.040	0.315±0.051	0.238±0.043	0.460±0.046	0.524±0.048	0.431±0.055
	0.3	0.277±0.037	0.309±0.046	0.239±0.035	0.339±0.043	0.359±0.056	0.294±0.040	0.640±0.063	0.687±0.050	0.623±0.070
	0.4	0.320±0.044	0.354±0.043	0.279±0.044	0.386±0.049	0.396±0.053	0.335±0.056	0.768±0.076	0.799±0.062	0.751±0.087
	0.5	0.335±0.040	0.331±0.057	0.305±0.036	0.393±0.053	0.348±0.033	0.326±0.052	0.806±0.083	0.834±0.070	0.787±0.095
	0.6	0.367±0.078	0.395±0.082	0.318±0.076	0.435±0.047	0.421±0.057	0.374±0.051	0.935±0.050	0.944±0.039	0.932±0.055
	0.7	0.394±0.068	0.404±0.064	0.337±0.065	0.446±0.055	0.447±0.071	0.387±0.056	**0.975±0.026**	**0.977±0.022**	**0.974±0.029**
	0.8	0.421±0.059	0.431±0.059	0.364±0.058	0.459±0.049	0.452±0.058	0.391±0.052	0.988±0.017	0.989±0.013	0.988±0.020
	0.9	0.449±0.059	0.458±0.053	0.398±0.057	0.495±0.052	0.492±0.059	0.429±0.052	0.994±0.015	0.995±0.012	0.993±0.018
	1.0	0.450±0.057	0.472±0.075	0.391±0.058	0.484±0.055	0.476±0.067	0.415±0.063	**0.997±0.004**	**0.997±0.004**	**0.997±0.004**
ISDB $\mathcal{L}_1$	0.1	0.593±0.029	0.623±0.035	0.565±0.038	0.656±0.051	0.670±0.050	0.648±0.057	0.726±0.052	0.733±0.053	0.724±0.053
	0.2	0.659±0.042	0.688±0.041	0.644±0.050	0.724±0.044	0.736±0.045	0.720±0.046	0.804±0.058	0.808±0.056	0.803±0.060
	0.3	0.697±0.044	0.722±0.040	0.688±0.049	0.755±0.047	0.768±0.042	0.752±0.050	0.822±0.044	0.827±0.042	0.821±0.045
	0.4	0.735±0.039	0.754±0.037	0.729±0.042	0.784±0.049	0.795±0.046	0.782±0.051	0.845±0.050	0.851±0.049	0.845±0.050
	0.5	0.727±0.049	0.747±0.042	0.721±0.054	0.799±0.040	0.810±0.037	0.797±0.041	0.862±0.045	0.872±0.039	0.861±0.046
	0.6	0.773±0.060	0.788±0.058	0.769±0.063	0.809±0.049	0.822±0.041	0.806±0.051	0.875±0.048	0.881±0.045	0.875±0.049
	0.7	0.795±0.043	0.813±0.035	0.791±0.046	0.843±0.036	0.850±0.034	0.842±0.036	0.885±0.042	0.892±0.038	0.885±0.043
	0.8	0.785±0.045	0.809±0.035	0.780±0.050	0.837±0.050	0.847±0.047	0.836±0.051	0.904±0.040	0.910±0.037	0.903±0.040
	0.9	0.816±0.059	0.834±0.049	0.812±0.063	0.853±0.050	0.863±0.046	0.852±0.051	0.903±0.064	0.911±0.054	0.902±0.067
	1.0	0.831±0.049	0.848±0.041	0.828±0.051	0.871±0.051	0.880±0.046	0.870±0.052	0.904±0.044	0.911±0.042	0.904±0.045

Dataset	p	Basic+CA+SS+GDM			Basic+CA+SS+PL			Basic+CA+SS+PL+GDM		
		Acc	*w-Recall*	*w-F1*	*Acc*	*w-Recall*	*w-F1*	*Acc*	*w-Recall*	*w-F1*
CWRU $\mathcal{L}_2$	0.1	0.362±0.042	0.443±0.042	0.341±0.055	0.496±0.125	0.529±0.145	0.441±0.131	**0.550±0.089**	**0.585±0.097**	**0.493±0.098**
	0.2	0.497±0.060	0.560±0.031	0.472±0.059	0.656±0.113	0.700±0.109	0.601±0.128	**0.704±0.091**	**0.728±0.109**	**0.657±0.107**
	0.3	0.686±0.038	0.722±0.040	0.672±0.047	0.826±0.084	0.852±0.084	0.800±0.102	**0.840±0.076**	**0.853±0.085**	**0.816±0.092**
	0.4	0.803±0.077	0.834±0.058	0.789±0.085	**0.893±0.072**	**0.907±0.075**	**0.875±0.087**	0.888±0.074	0.904±0.075	0.871±0.090
	0.5	0.862±0.078	0.865±0.084	0.846±0.095	0.897±0.072	0.908±0.085	0.878±0.091	**0.924±0.077**	**0.930±0.082**	**0.910±0.095**
	0.6	0.940±0.044	0.944±0.042	0.931±0.055	**0.961±0.043**	**0.967±0.041**	**0.957±0.052**	0.952±0.051	0.958±0.052	0.945±0.063
	0.7	0.971±0.032	0.972±0.035	0.968±0.041	0.954±0.046	0.954±0.059	0.946±0.059	0.970±0.045	0.970±0.051	0.964±0.057
	0.8	0.989±0.013	0.990±0.010	0.989±0.013	0.968±0.043	0.974±0.034	0.962±0.054	**0.992±0.021**	**0.992±0.024**	**0.991±0.026**
	0.9	**0.995±0.005**	**0.995±0.005**	**0.995±0.005**	0.977±0.042	0.978±0.042	0.972±0.054	0.994±0.019	0.995±0.019	0.993±0.023
	1.0	0.992±0.018	0.993±0.012	0.991±0.022	0.989±0.021	0.992±0.014	0.988±0.025	0.996±0.019	0.995±0.024	0.995±0.024
ISDB $\mathcal{L}_1$	0.1	0.756±0.044	0.769±0.043	0.753±0.045	0.755±0.055	0.780±0.054	0.750±0.057	**0.766±0.044**	**0.794±0.031**	**0.761±0.048**
	0.2	**0.821±0.027**	**0.827±0.027**	**0.820±0.027**	0.814±0.051	0.831±0.042	0.812±0.053	0.806±0.042	0.815±0.040	0.805±0.043
	0.3	0.827±0.035	0.835±0.035	0.826±0.036	0.836±0.058	0.852±0.053	0.833±0.059	**0.876±0.030**	**0.881±0.029**	**0.875±0.031**
	0.4	0.852±0.035	0.859±0.032	0.852±0.035	0.865±0.034	0.877±0.027	0.863±0.035	**0.891±0.034**	**0.898±0.030**	**0.891±0.035**
	0.5	0.867±0.039	0.873±0.036	0.867±0.039	0.862±0.051	0.875±0.042	0.861±0.053	**0.901±0.044**	**0.906±0.039**	**0.900±0.045**
	0.6	0.870±0.045	0.877±0.041	0.869±0.045	0.878±0.042	0.888±0.037	0.877±0.043	**0.890±0.032**	**0.897±0.031**	**0.890±0.032**
	0.7	0.889±0.036	0.897±0.032	0.889±0.036	0.877±0.043	0.888±0.037	0.876±0.044	**0.906±0.026**	**0.911±0.024**	**0.905±0.027**
	0.8	0.901±0.030	0.906±0.028	0.901±0.030	0.878±0.058	0.891±0.049	0.876±0.061	**0.908±0.032**	**0.914±0.028**	**0.908±0.032**
	0.9	0.915±0.037	0.921±0.033	0.914±0.037	0.904±0.044	0.911±0.039	0.904±0.045	**0.924±0.044**	**0.934±0.033**	**0.923±0.045**
	1.0	0.916±0.046	0.922±0.042	0.915±0.047	0.899±0.046	0.906±0.042	0.899±0.047	**0.920±0.054**	**0.932±0.043**	**0.919±0.055**

VII. CONCLUSION

This article introduced a novel debiased contrastive learning with supervision guidance framework in the context of industrial fault detection tasks. The main focus is addressing three biases inherent in two-phase contrastive learning: Task-agnostic represenation issue, sampling bias, and angular-centricity issue. The supervisory signal was introduced to tackle task-agnostic representation issue by utilizing the BWUS for information interaction. Pseudolabels were assigned to representations through momentum clustering to mitigate sampling bias, and GDM was proposed to measure the radial distribution between representations comprehensively. Moreover, a channel augmentation technique was proposed, fusing temporal and frequency information and empowering DCLSG to mine representations from both local and global perspectives. Experimental results on the ISDB, CWRU, and practical datasets validated the effectiveness of these four innovations in enhancing the performance of the proposed framework.

Although GDM has demonstrated success, it is not a norm. It results in asymmetry where $GDM(f_i, f_j) \neq GDM(f_j, f_i)$ though having same optimization target ($f_i = f_j$). The disadvantage of nonnorm may lead to an unstable optimization process and affect the convergence of DCLSG. Future research will focus on finding a suitable function that unifies $GDM(f_i, f_j)$ and $GDM(f_j, f_i)$.

REFERENCES

[1] K. Zhang, Y. Liu, Y. Gu, J. Wang, and X. Ruan, "Valve stiction detection using multitimescale feature consistent constraint for time-series data," *IEEE/ASME Trans. Mechatron.*, vol. 28, no. 3, pp. 1488–1499, Jun. 2023.

[2] C. Yang, J. Liu, Q. Xu, and K. Zhou, "A generalized graph contrastive learning framework for few-shot machine fault diagnosis," *IEEE Trans. Ind. Inform.*, vol. 20, no. 2, pp. 2692–2701, Feb. 2024.

[3] R. Chen, Z. Cai, and J. Yuan, "UIESC: An underwater image enhancement framework via self-attention and contrastive learning," *IEEE Trans. Ind. Inform.*, vol. 19, no. 12, pp. 11701–11711, Dec. 2023.

[4] P. Peng, J. Lu, T. Xie, S. Tao, H. Wang, and H. Zhang, "Open-set fault diagnosis via supervised contrastive learning with negative out-of-distribution data augmentation," *IEEE Trans. Ind. Inform.*, vol. 19, no. 3, pp. 2463–2473, Mar. 2023.

[5] K. Zhang, Y. Liu, Y. Gu, X. Ruan, and J. Wang, "Multiple-timescale feature learning strategy for valve stiction detection based on convolutional neural network," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 3, pp. 1478–1488, Jun. 2022.

[6] X. Zhang, Z. Zhao, T. Tsiligkaridis, and M. Zitnik, "Self-supervised contrastive pre-training for time series via time-frequency consistency," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 3988–4003, 2022.

[7] L. Guo, Y. Lei, S. Xing, T. Yan, and N. Li, "Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data," *IEEE Trans. Ind. Electron.*, vol. 66, no. 9, pp. 7316–7325, Sep. 2019.
[8] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 9929–9939.
[9] C.-Y. Chuang, J. Robinson, Y.-C. Lin, A. Torralba, and S. Jegelka, "Debiased contrastive learning," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 8765–8775, 2020.
[10] J. D. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, "Contrastive learning with hard negative samples," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–29.
[11] R. Cai, L. Peng, Z. Lu, K. Zhang, and Y. Liu, "DCS: Debiased contrastive learning with weak supervision for time series classification," in *Proc. IEEE ICASSP 2024-2024 Int. Conf. Acoust., Speech Signal Process.*, 2024, pp. 5625–5629.
[12] M. Jelali and B. Huang, *Detection and Diagnosis of Stiction in Control Loops: State of the Art and Advanced Methods*. London: Springer-Verlag. 2010.
[13] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.
[14] Z. Hu, Y. Hu, B. Wu, J. Liu, D. Han, and T. Kurfess, "Hand pose estimation with multi-scale network," *Appl. Intell.*, vol. 48, pp. 2501–2515, 2018.
[15] Q. P. He, J. Wang, M. Pottmann, and S. J. Qin, "A curve fitting method for detecting valve stiction in oscillating control loops," *Ind. Eng. Chem. Res.*, vol. 46, no. 13, pp. 4549–4560, 2007.
[16] Y. Wang, Z. Xiao, and G. Cao, "A convolutional neural network method based on adam optimizer with power-exponential learning rate for bearing fault diagnosis," *J. Vibroengineering*, vol. 24, pp. 666–678, Mar. 2022.
[17] J. W. Dambros, M. Farenzena, and J. O. Trierweiler, "Data-based method to diagnose valve stiction with variable reference signal," *Ind. Eng. Chem. Res.*, vol. 55, no. 39, pp. 10316–10327, 2016.
[18] H. Shao, H. Jiang, X. Zhang, and M. Niu, "Rolling bearing fault diagnosis using an optimization deep belief network," *Meas. Sci. Technol.*, vol. 26, no. 11, 2015, Art. no. 115002.
[19] L. Wen, L. Gao, X. Li, M. Xie, and G. Li, "A new data-driven intelligent fault diagnosis by using convolutional neural network," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage.*, 2017, pp. 813–817.
[20] A. A. A. Mohd Amiruddin, H. Zabiri, S. S. Jeremiah, W. K. Teh, and B. Kamaruddin, "Valve stiction detection through improved pattern recognition using neural networks," *Control Eng. Pract.*, vol. 90, pp. 63–84, 2019.
[21] Z.-X. Hu, Y. Wang, M.-F. Ge, and J. Liu, "Data-driven fault diagnosis method based on compressed sensing and improved multiscale network," *IEEE Trans. Ind. Electron.*, vol. 67, no. 4, pp. 3216–3225, Apr. 2020.
[22] M. Rossi and C. Scali, "A comparison of techniques for automatic detection of stiction: Simulation and application to industrial data," *J. Process Control*, vol. 15, no. 5, pp. 505–514, 2005.
[23] W. Mao, W. Feng, Y. Liu, D. Zhang, and X. Liang, "A new deep auto-encoder method with fusing discriminant information for bearing fault diagnosis," *Mech. Syst. Signal Process.*, vol. 150, 2021, Art. no. 107233.
[24] A. Singhal and T. I. Salsbury, "A simple method for detecting valve stiction in oscillating control loops," *J. Process Control*, vol. 15, no. 4, pp. 371–382, 2005.
[25] M. Khodayar, O. Kaynak, and M. E. Khodayar, "Rough deep neural architecture for short-term wind speed forecasting," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 2770–2779, Dec. 2017.
[26] S. Karra and M. N. Karim, "Comprehensive methodology for detection and diagnosis of oscillatory control loops," *Control Eng. Pract.*, vol. 17, no. 8, pp. 939–956, 2009.
[27] M. Khodayar, J. Wang, and M. Manthouri, "Interval deep generative neural network for wind speed forecasting," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3974–3989, Jul. 2019.
[28] B. Kamaruddin, H. Zabiri, A. Mohd Amiruddin, W. Teh, M. Ramasamy, and S. Jeremiah, "A simple model-free butterfly shape-based detection (BSD) method integrated with deep learning cnn for valve stiction detection and quantification," *J. Process Control*, vol. 87, pp. 1–16, 2020.
[29] D. Ruan, J. Wang, J. Yan, and C. Gühmann, "CNN parameter design based on fault signal analysis and its application in bearing fault diagnosis," *Adv. Eng. Inform.*, vol. 55, 2023, Art. no. 101877.
[30] Z. Meng et al., "A novel generation network using feature fusion and guided adversarial learning for fault diagnosis of rotating machinery," *Expert Syst. With Appl.*, vol. 234, 2023, Art. no. 121058.
[31] W. Yu and P. Lv, "An end-to-end intelligent fault diagnosis application for rolling bearing based on mobilenet," *IEEE Access*, vol. 9, pp. 41925–41933, 2021.
[32] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, pp. 273–297, 1995.
[33] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001.
[34] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.
[35] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Jul. 2018.

Rongyao Cai (Graduate Student Member, IEEE) received the B.Eng. degree in chemical engineering from the Zhejiang University of Technology, Hangzhou, China, in 2022. He is currently working toward the M.Eng. degree in control engineering with the College of Control Science and Engineering, Zhejiang University, Hangzhou.

His research interests include time series analysis, deep learning, and data-driven industrial modeling.

Wang Gao received the M.S. degree in control engineering from The Third Research Institute, China Aerospace Science and Industry Corporation, Beijing, China, in 2017.

His research interests include computer vision, scene matching, and visual navigation.

Linpeng Peng is currently working toward the Ph.D. degree in control engineering with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China.

His goal is to develop intelligent algorithms and systems for robots to finish complex tasks and help people. His research interests include computer vision and robotics.

Zhengming Lu received the B.S. degree in physics from Nanjing University, Nanjing, China. He is currently working toward the M.E. degree in control engineering with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China.

His research interests include time series analysis, deep reinforcement learning, and adversarial attack on deep learning system.

Kexin Zhang (Graduate Student Member, IEEE) received the B.Eng. and M.Eng. degrees in control engineering from the China University of Geosciences, Wuhan, China, in 2016 and 2019, respectively, and the Ph.D. degree in control engineering and science from Zhejiang University, Hangzhou, China, in 2023.

His research interests include intelligent time series analysis, deep learning, data-driven industrial fault diagnosis, and artificial intelligence security.

Yong Liu (Member, IEEE) received the B.S. degree in computer science and engineering and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2001 and 2007, respectively.

He is currently a Professor with the Institute of Cyber-Systems and Control, College of Control Science and Engineering, Zhejiang University. He has authored or coauthored more than 100 research papers in machine learning, computer vision, information fusion, and robotics. His current research interests include machine learning, robotics vision, information processing, and granular computing.