Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Cross device milling force prediction method based on self supervised pre training and domain adaptation

Zhigang Ma

ZhigangMa7@outlook.com

Xinxiang Vocational and Technical College

Research Article

Keywords: Cross-device milling force prediction, Self-supervised pretraining, Domain adaptation, Few-label fine-tuning, Vibration signal

Posted Date: September 1st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10676193/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Cross device milling force prediction method based on self supervised pre training and domain adaptation

Zhigang Ma*

Intelligent Manufacturing School,Xinxiang Vocational and Technical College, Xinxiang, 453006 , China

Zhigang Ma's e-mail:ZhigangMa7@outlook.com

Abstract: To address the degradation of milling force prediction accuracy caused by device-dependent dynamic response differences and insufficient force labels in the target domain, this study proposes a cross-device milling force prediction method integrating self-supervised pretraining, domain adaptation, and few-label fine-tuning. Unlabeled vibration signals are first used for masked vibration modeling to learn transferable temporal and spectral representations. Then, source-domain supervised regression, source–target feature alignment, and limited target-domain label calibration are jointly introduced to transfer the vibration–force mapping across devices. Experimental results show that the average R^2 on the source device reaches 0.9923. With only 5% labeled target-domain samples, the SSL+DA method achieves an R^2 of 0.957, an RMSE of 24.9 N, and an MAE of 17.8 N. When the label ratio increases to 20%, the RMSE further decreases to 19.4 N, verifying the effectiveness of the proposed method.

Keywords: Cross-device milling force prediction; Self-supervised pretraining; Domain adaptation; Few-label fine-tuning; Vibration signal

1 Introduction

Milling is one of the most widely used material removal processes in aerospace, energy equipment, automotive manufacturing, and precision die and mold industries. Owing to the intermittent engagement between the rotating cutter and the workpiece, milling is characterized by strong nonlinearity, dynamic coupling, and time-varying process behavior. Milling force is a key physical quantity that reflects the interaction between the tool and the

workpiece, and it directly affects tool wear, machining vibration, surface integrity, dimensional accuracy, and the operational safety of machine tools. Therefore, accurate acquisition and prediction of milling force are not only fundamental for understanding cutting mechanisms, but also essential for online process monitoring, intelligent machining control, and digital manufacturing. If milling force can be predicted in real time from easily measurable vibration signals, it can provide important support for tool condition monitoring, chatter warning, process parameter optimization, and machining quality control without significantly modifying the machining system [1].

Conventional milling force acquisition mainly relies on direct measurement devices such as dynamometers. Although direct measurement can provide high accuracy, it usually requires high installation cost, strict fixture design, and additional space in the machining system. In some cases, the installation of a dynamometer may also change the clamping stiffness of the workpiece and the dynamic characteristics of the machining system, making it difficult to apply in long-term industrial production. In contrast, indirect sensing signals such as vibration, acoustic emission, motor current, and spindle power are easier to collect, less intrusive, and more suitable for industrial deployment. Among these signals, vibration is closely related to tool entry, tool exit, periodic tooth impact, and cutting load variation. It can reflect the local dynamic response and transient mechanical behavior during milling. Therefore, establishing a mapping relationship between vibration signals and milling force provides a feasible way to realize non-invasive milling force prediction [2].

However, vibration-based milling force prediction still faces significant challenges in cross-device applications. Different machine tools may differ in structural stiffness, spindle systems, tool-holder connections, sensor installation positions, acquisition chains, and dynamic transmission paths. Even when the workpiece material, tool type, and cutting parameters are similar, the collected vibration signals may still show different amplitude scales, frequency distributions, and temporal characteristics. This means that a prediction model trained on a source device may suffer a serious accuracy degradation when directly applied to a target device. More importantly, the target device usually lacks sufficient labeled milling force data, because force label acquisition requires additional measuring equipment, experimental cost, and signal synchronization. As a result, the practical cross-device scenario

is often characterized by sufficient labeled source-domain data, abundant unlabeled target-domain vibration data, and only a small amount of labeled target-domain force data. This has become a key bottleneck restricting the practical application of cross-device milling force prediction [3].

Existing data-driven methods can often achieve high prediction accuracy under the same device, the same acquisition system, or similar working conditions. However, these methods usually assume that training and testing samples follow similar distributions. Once the device changes, the model may learn source-device-specific features, such as sensor installation characteristics, structural transmission characteristics, and amplitude scaling patterns, rather than stable force-related dynamic features. Simply increasing the depth or complexity of the neural network cannot fundamentally solve this problem and may even aggravate overfitting to the source-domain distribution. On the other hand, fine-tuning the model only with a small number of labeled target-domain samples is also unstable, because the available samples may be insufficient to cover different working conditions and may be affected by noise and signal uncertainty. Therefore, cross-device milling force prediction requires not only a strong nonlinear regression model, but also a feature learning mechanism that can utilize unlabeled target-domain data, reduce device-induced discrepancies, and preserve force-related information [4].

To address these problems, this paper proposes a cross-device milling force prediction method based on self-supervised pretraining and domain adaptation. Vibration signals are used as the main model input, and stable cutting segment extraction, normalization, temporal alignment, and sliding-window segmentation are conducted to construct unified input samples. In the training stage, unlabeled vibration windows from both source and target devices are first used to construct a masked vibration modeling task, enabling the feature extractor to learn intrinsic temporal structures, channel correlations, and local dynamic patterns without relying on force labels. Then, labeled source-domain samples are used to provide supervised milling force regression constraints, while a domain adaptation mechanism is introduced to align the feature distributions between the source and target domains. Finally, a small number of labeled target-domain samples are used to calibrate the prediction mapping, allowing the

model to adapt to the force amplitude and dynamic response characteristics of the target device.

The main contributions of this paper are summarized as follows. First, a unified learning framework is established for cross-device milling force prediction, in which labeled source-domain data, unlabeled target-domain data, and few labeled target-domain samples are jointly used. This framework reduces the dependence on large-scale labeled target-domain force data. Second, a self-supervised pretraining strategy is designed for vibration signals. By using masked vibration modeling, the feature extractor can learn transferable temporal and spectral representations, allowing unlabeled target-device vibration data to participate in representation learning before supervised force prediction. Third, domain adaptation and few-label fine-tuning are integrated into the milling force regression process. This strategy reduces device-related distribution discrepancies while preserving force-sensitive feature representations, thereby improving prediction accuracy and transfer stability on the target device. Overall, this study provides a feasible solution for extending milling force prediction models from single-device experimental validation to cross-device, few-label, and practical industrial deployment scenarios.

2 Literature Review

Research on milling force monitoring and prediction has mainly developed along three technical routes: mechanistic modeling, process signal sensing, and data-driven modeling. Early studies on machining monitoring focused on tool condition, cutting load, and process stability. Byrne et al. systematically reviewed the development of tool condition monitoring and pointed out that sensor signals, feature extraction, and decision models are the core components of machining process monitoring systems [5]. In the field of mechanistic milling force modeling, Kline et al. established an early cutting force prediction model for end milling and considered the variation of chip thickness in cornering cuts, which laid a foundation for subsequent mechanistic modeling studies [6]. Budak, Altintas, and Armarego further derived milling force coefficients from orthogonal cutting data, enabling milling force models to better adapt to different tool geometries and cutting conditions [7]. Kaymakci, Kilic, and Altintas proposed a unified cutting force model for turning, boring, drilling, and milling

operations, which improved the generality of force modeling across different machining processes [8]. These mechanistic models have clear physical meanings and can reveal the relationship between cutting force, tool geometry, material removal, and process parameters. However, they usually require accurate information about tool geometry, material properties, cutting coefficients, and system dynamics. When machining conditions are complex, devices are changed, or transmission paths are uncertain, parameter identification and generalization remain difficult.

Indirect force sensing has become an important research direction for practical machining monitoring. Matsubara and Ibaraki reviewed monitoring and control methods for cutting forces in machining processes and indicated that although direct force measurement devices provide high accuracy, their industrial application is limited by cost, installation constraints, and system integration difficulties [9]. Therefore, indirect signals such as motor current, displacement, and vibration have stronger deployment potential. Kim et al. investigated indirect cutting force measurement in multi-axis simultaneous NC milling processes and demonstrated the feasibility of estimating cutting force using internal machine response signals [10]. Postel, Aslan, Wegener, and Altintas monitored vibration and cutting force using spindle-mounted vibration sensors and emphasized the influence of machine tool transfer characteristics on force estimation accuracy [11]. These studies show that vibration signals contain rich cutting force information, but they also reveal an important issue: vibration is not the cutting force itself, but the dynamic response generated after the cutting force passes through the machine–tool–workpiece system. Therefore, structural differences between devices may change the form of vibration signals and make a model trained on one device difficult to apply directly to another device.

With the development of machine learning, increasing attention has been paid to data-driven models for establishing nonlinear mappings between sensor signals and cutting forces. Zhou, Mao, Liu, Li, and Peng predicted milling cutting force using machine tool vibration signals and a Kalman filtering-based method, verifying the effectiveness of vibration signals for indirect milling force prediction [12]. Chen, Zhao, and Zhang proposed a parallel deep learning model with dual-channel vibration fusion for real-time milling force monitoring, showing that deep networks can extract richer temporal information from

complex vibration signals than traditional handcrafted features [13]. From the broader perspective of machine health monitoring, LeCun, Bengio, and Hinton systematically explained the basic principles of deep learning and showed that deep neural networks can automatically learn hierarchical feature representations [14]. Zhao, Yan, Chen, Mao, Wang, and Gao reviewed deep learning applications in machine health monitoring and concluded that convolutional networks, recurrent networks, and autoencoder structures are effective for processing high-dimensional time-series signals such as vibration [15]. Lei et al. further pointed out that distribution variation, insufficient labeled samples, and limited interpretability are major challenges for deploying machine learning methods in machinery fault diagnosis and condition monitoring [16]. These studies have promoted the transition from handcrafted feature engineering to end-to-end feature learning in machining monitoring. However, most existing methods still assume that training and testing data come from the same device or similar distributions, and insufficient attention has been paid to cross-device distribution shifts.

Temporal sequence modeling methods provide important tools for milling force prediction. The long short-term memory network proposed by Hochreiter and Schmidhuber can alleviate the long-term dependency problem in conventional recurrent neural networks and is suitable for processing sensor sequences with temporal correlations [17]. In vibration-based milling force prediction, LSTM-type models can capture periodic tooth impacts, local amplitude variations, and delayed response relationships. Nevertheless, their performance still depends heavily on sufficient labeled data. To reduce the dependence on labels, representation learning and self-supervised learning have attracted increasing interest. Hinton and Salakhutdinov demonstrated that deep autoencoder networks can learn effective low-dimensional representations through reconstruction tasks [18]. Vincent et al. proposed stacked denoising autoencoders, which learn robust features by reconstructing corrupted inputs. This idea is intrinsically consistent with masked vibration modeling, where missing signal components are inferred from visible contexts [19]. Bengio, Courville, and Vincent further pointed out from the perspective of representation learning that unsupervised or self-supervised objectives can help models learn more transferable abstract features [20]. For milling vibration signals, large amounts of unlabeled data are usually easier to obtain than

force labels. Therefore, self-supervised pretraining provides a promising technical route for cross-device milling force prediction.

Transfer learning and domain adaptation provide theoretical foundations and methodological tools for addressing distribution shifts caused by device differences. Shao, McAleer, Yan, and Baldi applied deep transfer learning to machine fault diagnosis and showed that pretrained source-task features can improve performance on target tasks with limited samples [21]. Zhao et al. reviewed and compared unsupervised deep transfer learning methods for intelligent fault diagnosis, indicating that working condition variation, device variation, and unlabeled target domains are common problems in industrial intelligent diagnosis [22]. From the methodological perspective, Ganin et al. proposed domain-adversarial neural networks, which learn domain-invariant features through a domain discriminator and a gradient reversal layer, providing a classical framework for unsupervised domain adaptation [23]. Gretton et al. proposed maximum mean discrepancy as a kernel-based method for measuring the difference between two distributions, which has been widely used for feature distribution alignment [24]. Wilson and Cook reviewed unsupervised deep domain adaptation methods and pointed out that the key to domain adaptation is to maintain both task discriminability and domain invariance [25]. Wang and Deng summarized deep visual domain adaptation methods and emphasized that simply aligning marginal distributions may not be sufficient to guarantee target-task performance; models must also avoid negative transfer and preserve task-related structures [26]. These conclusions are also applicable to cross-device milling force prediction. It is not enough to make source-domain and target-domain features appear similar; The aligned features must still retain information related to milling force variation.

In summary, previous studies provide important foundations for this research. Mechanistic models reveal the physical relationship among milling force, cutting parameters, tool geometry, and material removal. Indirect force sensing studies demonstrate that vibration signals can serve as effective proxies for milling force prediction. Deep learning methods enhance the ability to model complex nonlinear mappings. Self-supervised learning and domain adaptation further provide the possibility of utilizing unlabeled data, reducing distribution discrepancy, and alleviating dependence on target-domain labels. Nevertheless,

several limitations remain. First, many milling force prediction models are mainly designed for single-device or same-distribution scenarios and do not explicitly model device-dependent dynamic response differences. Second, existing deep learning methods usually require sufficient labeled target-domain samples, while force label acquisition on the target device is expensive and difficult in practice. Third, most transfer learning studies focus on fault classification or condition recognition, whereas continuous milling force regression requires additional attention to temporal alignment, amplitude calibration, and preservation of force-related features. To analyze cross-domain feature distribution changes, visualization methods such as t-SNE proposed by van der Maaten and Hinton are often used to observe the clustering and overlap between source and target features, but visualization itself cannot replace an effective domain adaptation training mechanism [27]. Therefore, it is necessary and valuable to integrate vibration-oriented self-supervised pretraining, source–target feature alignment, and few-label target-domain fine-tuning into a unified cross-device milling force regression framework.

3 Method

3.1 Problem Definition and Overall Framework for Cross-Device Milling Force Prediction

During milling, the transient interaction between the cutter and the workpiece generates cutting forces that excite the spindle, tool holder, fixture, workpiece, and machine table. These structural responses can be captured by vibration sensors, making vibration signals a practical proxy for indirect milling force prediction. Compared with direct force measurement, vibration-based monitoring is less intrusive and easier to deploy in industrial environments. However, the mapping from vibration response to milling force is not fixed across devices. Differences in machine-tool stiffness, sensor installation position, signal acquisition system, tool-workpiece combination, and dynamic transmission path may alter the amplitude, frequency distribution, and temporal characteristics of the measured vibration signal. Therefore, a model trained on one device may suffer significant performance degradation when directly applied to another device.

Let the source device domain and target device domain be denoted as

$$D_s=\{(x_i^s,y_i^s)\}_{i=1}^{N_s},\ D_t=\{(x_j^t,y_j^t)\}_{j=1}^{N_t} \quad (1)$$

Where x_i^s and x_j^t represent vibration-signal samples collected from the source and target devices, respectively, while y_i^s and y_j^t denote the corresponding milling force labels. In practical cross-device scenarios, the source device usually provides sufficient labeled data, whereas the target device can only provide a small number of labeled samples because force-label acquisition requires additional measurement equipment and experimental cost. Thus, the target domain can be further divided into labeled and unlabeled subsets:

$$D_t=D_t^l\cup D_t^u,\ N_t^l\ll N_s \quad (2)$$

This setting reflects the main difficulty of cross-device milling force prediction: the target device contains abundant vibration information but limited force labels. If only source-domain supervised learning is used, the learned model tends to fit the source-device signal distribution and may fail to capture target-device response characteristics. The mismatch between the source and target domains can be expressed as [28]

$$P_s(x,y)\neq P_t(x,y) \quad (3)$$

Where $P_s(x,y)$ and $P_t(x,y)$ are the joint distributions of vibration samples and milling force labels in the two domains. Such distribution discrepancy is caused not only by different machining parameters but also by device-specific dynamic responses. Consequently, cross-device prediction requires a feature representation that preserves milling-force-related information while reducing sensitivity to device-dependent signal variations [29].

To construct the model input, the continuous vibration signal is segmented into local time windows. For a vibration sequence $\{a_1,a_2,\ldots,a_T\}$, the k-th window sample is defined as

$$x_k=[a_k,a_{k+1},\ldots,a_{k+L-1}] \quad (4)$$

Where L is the window length. Window-based modeling allows the network to capture local dynamic patterns, periodic force fluctuations, and transient signal variations induced by cutting engagement. The cross-device prediction task formed by source-device labeled data, target-device unlabeled data, and a small amount of target-device labeled data is summarized in Figure 1, where the essential objective is to transfer force-related vibration representations from the source device to the target device while suppressing domain-specific interference.

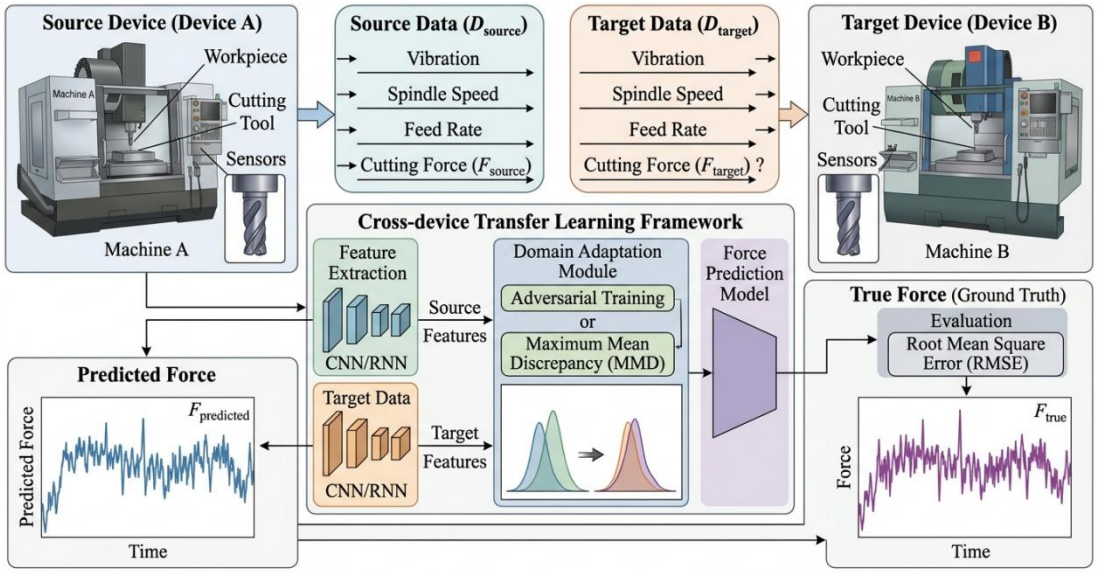


Figure 1 Cross-device milling force prediction task schematic

Based on this task definition, the prediction model can be described as a composition of a feature extractor and a regression predictor:

$$\mathrm{h}=G_{\phi}(x),\ \hat{y}=R_{\psi}(\mathrm{h}) \qquad (5)$$

Where $G_{\phi}(\cdot)$ maps the vibration window into a latent feature representation h, and $R_{\psi}(\cdot)$ estimates the milling force $\hat{y}$. The ideal optimization goal is to minimize the prediction risk on the target device:

$$\min_{\phi,\psi} \mathrm{R}_t = E_{(x,y)\sim P_t}\left[\mathrm{l}\left(R_{\psi}\left(G_{\phi}(x)\right),y\right)\right], \qquad (6)$$

Where $\ell(\cdot)$ denotes the regression loss. Since target-domain labels are limited, the target risk cannot be reliably minimized using supervised learning alone. Therefore, the overall framework introduces self-supervised pretraining, domain-adaptive feature learning, and few-label fine-tuning into a unified training process.

Specifically, unlabeled vibration windows from both source and target devices are first used to construct a masked vibration modeling task, enabling the feature extractor to learn intrinsic temporal and spectral structures without relying on force labels. Then, source-domain labeled samples provide supervised force-prediction constraints, while target-domain unlabeled samples participate in feature distribution alignment. Finally, the small labeled subset from the target device is used to fine-tune the prediction mapping, allowing the model to adapt to the force amplitude and dynamic response characteristics of the target device. The complete procedure is organized in Figure 2, where signal

preprocessing, self-supervised representation learning, domain adaptation, few-label fine-tuning, and final milling force prediction are connected as a continuous workflow.

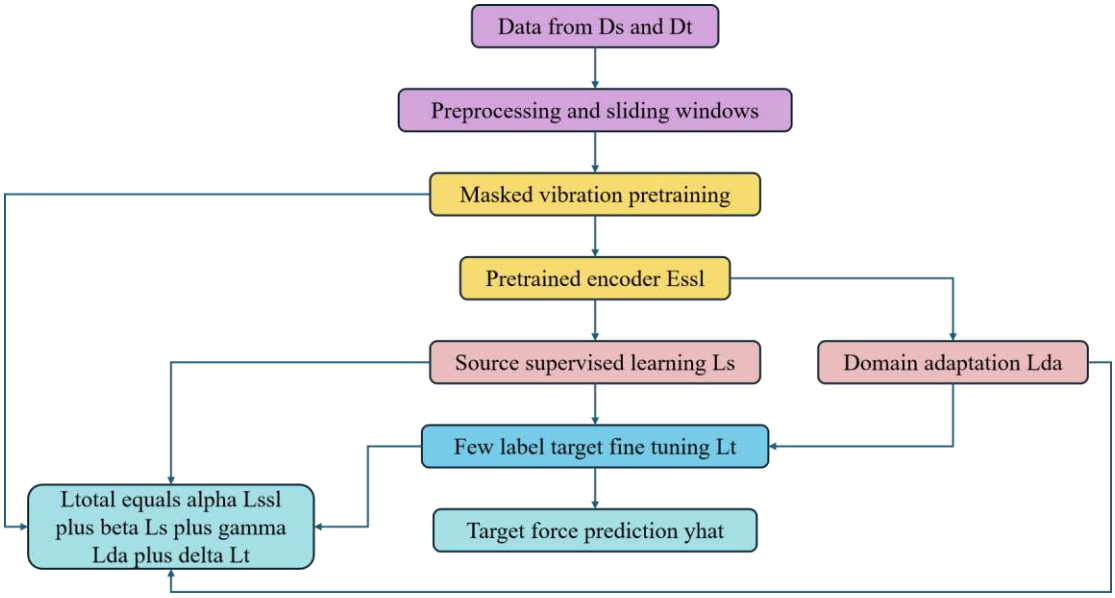


Figure 2 Overall framework of the proposed method

The training objective can be formulated as

$$L=L_{ssl}+\lambda_1 L^{s}_{pred}+\lambda_2 L_{da}+\lambda_3 L^{t}_{ft}, \quad (7)$$

Where L_{ssl} is the self-supervised loss for masked vibration modeling, L^{s}_{pred} is the supervised milling force prediction loss on the source domain, L_{da} constrains the discrepancy between source and target feature distributions, and L^{t}_{ft} denotes the few-label fine-tuning loss on the target domain. The coefficients λ_1, λ_2, and λ_3 balance the contributions of different learning objectives. Through this formulation, the model learns general vibration representations from unlabeled data, obtains force-prediction capability from source-domain labels, and adapts to the target device with limited labeled samples, thereby improving cross-device milling force prediction performance.

3.2 Vibration Signal Preprocessing and Self-Supervised Pretraining Task Construction

In cross-device milling force prediction, the raw vibration signal cannot be directly used as the model input without preprocessing. During data acquisition, the recorded signal usually contains spindle idling, tool approach, stable cutting, and tool exit stages. Only the stable

cutting segment contains a continuous and reliable dynamic correspondence between vibration response and milling force. In addition, vibration signals collected from different devices may differ in amplitude scale, sampling stability, sensor position, and transmission delay. If these factors are not handled before feature learning, the model may learn device-specific signal patterns rather than force-related dynamic characteristics. Therefore, vibration signal preprocessing is designed to provide temporally aligned, scale-consistent, and locally structured samples for subsequent self-supervised representation learning.

For a multichannel vibration signal collected from one device, the raw sequence can be denoted as

$$A=[a_1,a_2,\ldots,a_T]^{\mathrm{T}}\in\mathrm{R}^{T\times C}, \quad (8)$$

Where T is the number of sampling points and C is the number of vibration channels. The non-cutting segments are first removed according to the stable cutting interval, and the retained signal is normalized to reduce the influence of amplitude differences among devices and channels. For the ccc-th channel, min-max normalization is adopted as

$$\tilde{a}_{t,c}=\frac{a_{t,c}-\min(A_c)}{\max(A_c)-\min(A_c)+\varepsilon}, \quad (9)$$

Where A_c denotes all sampling points of the c-th channel, and ε is a small constant used to avoid division by zero. This operation preserves the temporal fluctuation pattern of the vibration signal while reducing scale inconsistency caused by sensor sensitivity and device response differences.

Because vibration and force signals are acquired through different sensors and signal acquisition systems, a time offset may exist between the vibration response and the measured milling force. This offset is affected by the mechanical transmission path, sensor installation position, and synchronization error of acquisition devices. To improve the correspondence between the input vibration window and the output force label, the time delay is estimated by maximizing the cross-correlation between the vibration signal and the force signal [30]:

$$\tau^{*}=\arg\max_{\tau}\sum_{t=1}^{T-\tau}\tilde{a}_{t,c}\,y_{t+\tau}, \quad (10)$$

Where y_t is the measured milling force and τ^{*} denotes the estimated alignment offset. After the offset is obtained, the force label corresponding to the vibration signal is adjusted as

$$\tilde{y}_t = y_{t+\tau^*}. \quad (11)$$

This alignment does not assume that the vibration response is always ahead of or behind the force signal; Instead, it estimates the most suitable temporal correspondence according to the actual signal relationship. As a result, the model can learn the dynamic mapping between vibration and force under a more reliable time reference.

After normalization and temporal alignment, the continuous vibration sequence is divided into overlapping windows. For the k-th sample, the input window is constructed as

$$X_k = [\tilde{a}_k, \tilde{a}_{k+1}, \ldots, \tilde{a}_{k+L-1}] \in \mathrm{R}^{L \times C}, \quad (12)$$

Where L is the window length. The corresponding force label can be selected from the aligned force sequence according to the prediction target, such as the endpoint force value or the average force value within the window. In this section, the endpoint label is defined as

$$Y_k = \tilde{y}_{k+L-1}. \quad (13)$$

The sliding window strategy enables each sample to retain local temporal continuity and periodic cutting information. More importantly, it provides a unified sample form for both supervised force prediction and self-supervised pretraining. The data transformation from raw vibration sequence to window samples, and then to masked vibration modeling, is summarized in Figure 3. In this process, the sliding window first constructs structured vibration segments, and the masking operation further forces the model to infer missing signal components from the remaining context [31].

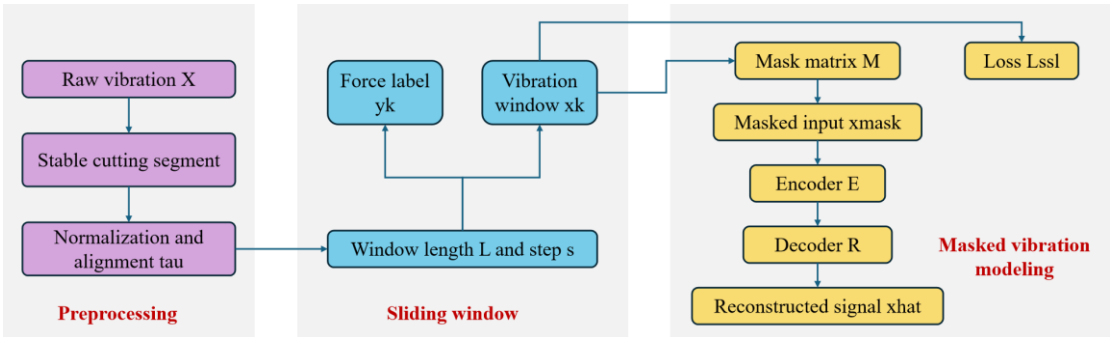


Figure 3 Sliding window and masked vibration modeling flowchart

The motivation for introducing self-supervised pretraining is that target-device force labels are usually scarce, whereas unlabeled vibration signals are relatively easy to collect. If the feature extractor is trained only by source-domain force labels, its representation may be

biased toward the source device. By contrast, a self-supervised task can use unlabeled vibration windows from both source and target devices to learn general temporal and spectral structures before supervised prediction. Masked vibration modeling is adopted for this purpose. Given a vibration window X_k, a binary mask matrix $M_k \in \{0,1\}^{L\times C}$ is generated, where masked positions are set to 1 and visible positions are set to 0. The corrupted input is obtained as [32]

$$\bar{X}_k=(1-M_k)\odot X_k+M_k\odot X_{\text{mask}}, \quad (14)$$

Where $\odot$ denotes element-wise multiplication, and X_{mask} is a learnable mask token or a zero-valued replacement. The encoder receives $\bar{X}_k$ and extracts contextual features from the visible vibration segments, while the decoder reconstructs the original signal values at the masked positions. The reconstruction objective is formulated as

$$L_{ssl}=\frac{1}{|\Omega_m|}\sum_{(t,c)\in\Omega_m}\left\| D_\omega\left(E_\phi(\bar{X}_k)\right)_{t,c}-X_{k,t,c}\right\|_2^2, \quad (15)$$

Where $E_\phi(\cdot)$ is the encoder, $D_\omega(\cdot)$ is the decoder, and $\Omega_m=\{(t,c)|M_{k,t,c}=1\}$ is the set of masked positions. Since the loss is calculated only on the masked elements, the encoder must infer missing vibration information from temporal context, channel correlation, and local dynamic patterns. This design encourages the model to learn intrinsic structures of milling vibration signals rather than memorizing complete input sequences.

The pretrained encoder is then used to initialize the feature extraction module in the subsequent cross-device prediction model. The learned representation can be expressed as

$$H_k=E_\phi^*(X_k), \quad (16)$$

Where E_ϕ^* denotes the encoder after self-supervised pretraining, and H_k is the vibration representation transferred to the supervised prediction and domain adaptation stages. Through this process, unlabeled data from the target device participate in representation learning before force-label supervision is introduced, which helps reduce the dependence on large-scale labeled target-domain samples [33].

Algorithm 1 Self-supervised pretraining based on masked vibration modeling

Algorithm 1 Self-supervised pretraining based on masked vibration modeling

Require: Unlabeled vibration dataset $\mathcal{D}$, encoder f_θ, decoder g_ϕ
Require: Mask ratio r, batch size B, learning rate η, epochs E
Ensure: Pretrained encoder parameters θ

Initialize encoder parameters θ and decoder parameters ϕ
for epoch $= 1$ to E **do**
  for each mini-batch $\mathbf{X} = \{x_i\}_{i=1}^{B}$ sampled from $\mathcal{D}$ **do**
    Segment each vibration signal x_i into patches
    Generate a random binary mask $\mathbf{M}$ with mask ratio r
    Obtain visible patches $\mathbf{X}_{vis}$ according to $\mathbf{M}$
    Replace masked patches with learnable mask tokens
    Add positional embeddings to visible and masked tokens
    Encode visible vibration patches:
      $\mathbf{Z} = f_\theta(\mathbf{X}_{vis})$
    Decode latent representations and mask tokens:
      $\hat{\mathbf{X}} = g_\phi(\mathbf{Z}, \mathbf{M})$
    Extract reconstructed masked patches $\hat{\mathbf{X}}_{mask}$
    Extract original masked patches $\mathbf{X}_{mask}$
    Compute reconstruction loss:
      $\mathcal{L} = \|\hat{\mathbf{X}}_{mask} - \mathbf{X}_{mask}\|_2^2$
    Update parameters by gradient descent:
      $(\theta, \phi) \leftarrow (\theta, \phi) - \eta \nabla_{\theta,\phi} \mathcal{L}$
  end for
end for
Discard decoder g_ϕ after pretraining
Return pretrained encoder f_θ

Through the above preprocessing and pretraining procedure, raw vibration signals are transformed into aligned and normalized window samples, and the feature extractor obtains an initial representation from unlabeled source- and target-device data. This representation provides a more stable basis for subsequent domain adaptation and few-label fine-tuning, because the model has already learned general vibration dynamics before being constrained by device-specific force labels.

3.3 Domain-Adaptive Feature Learning and Few-Label Fine-Tuning Strategy

Although masked vibration modeling enables the encoder to learn general temporal structures from unlabeled vibration signals, self-supervised pretraining alone cannot completely eliminate the distribution discrepancy caused by different devices. In cross-device milling, the same force variation may correspond to different vibration amplitudes, frequency responses, and dynamic delays because the mechanical transmission paths and sensor installation conditions are not identical. Therefore, the pretrained representation still needs to be constrained by force prediction supervision and domain adaptation so that the extracted

features are both sensitive to milling force variation and less dependent on device-specific signal characteristics.

Let the pretrained encoder obtained in Section 3.2 be used as the initial feature extractor. For source-domain and target-domain vibration windows, the latent representations are defined as

$$z_i^s=E_\phi(X_i^s),\ z_j^t=E_\phi\left(X_j^t\right) \quad (17)$$

Where $E_\phi(\cdot)$ denotes the encoder initialized by masked vibration modeling, and z_i^s, z_j^t are the corresponding source and target feature vectors. Since the source domain contains sufficient milling force labels, its supervised prediction loss is first used to preserve the force-related information in the feature space:

$$\mathrm{L}_{pred}^s=\frac{1}{N_s}\sum_{i=1}^{N_s}\left\|R_\psi(z_i^s)-y_i^s\right\|_2^2, \quad (18)$$

Where $R_\psi(\cdot)$ is the regression prediction head. This loss ensures that the feature extractor does not only learn domain-invariant information, but also retains the dynamic patterns that are directly related to milling force. However, if the model is optimized only by Eq. (18), the learned representation may still be biased toward the source device, because all force supervision comes from the source-domain distribution.

To reduce this bias, domain-adaptive feature learning is introduced after self-supervised pretraining. A domain discriminator $D_\eta(\cdot)$ is connected to the latent feature space to identify whether a feature vector comes from the source or target device. The domain label is defined as [34]

$$d=\begin{cases}0, & z\in D_s,\\ 1, & z\in D_t.\end{cases} \quad (19)$$

The discriminator is trained to distinguish source-domain features from target-domain features, while the encoder is trained in the opposite direction to generate features that are difficult to separate by domain. The domain classification loss is expressed as

$$\mathrm{L}_{dom}=-\frac{1}{N_s+N_t}\sum_{z\in D_s\cup D_t}\left[d\log D_\eta(z)+(1-d)\log\left(1-D_\eta(z)\right)\right]. \quad (20)$$

Through adversarial optimization, the discriminator D_η improves its domain classification ability, whereas the encoder E_ϕ learns to reduce domain-specific differences in the latent space. This process can be formulated as

$$\min_{\phi,\psi}\max_{\eta}\quad \left(\mathrm{L}_{pred}^{s}-\lambda_d \mathrm{L}_{dom}\right), \tag{21}$$

Where λ_d controls the strength of domain adaptation. In implementation, a gradient reversal mechanism can be used between the encoder and the domain discriminator. During forward propagation, the features are passed unchanged; during backpropagation, the gradient from the domain discriminator is reversed before being transmitted to the encoder. In this way, the feature extractor is encouraged to learn domain-invariant representations while the prediction head remains supervised by milling force labels.

However, global feature alignment alone may be insufficient for high-accuracy force prediction on the target device. Since the target machine has its own structural response and force amplitude characteristics, a small number of labeled target-domain samples are further used to correct the prediction mapping. For the labeled target subset D_t^l, the fine-tuning loss is defined as

$$\mathrm{L}_{ft}^{t}=\frac{1}{N_f^l}\sum_{j=1}^{N_f^l}\left\|R_\psi\left(E_\phi\left(X_j^t\right)\right)-y_j^t\right\|_2^2. \tag{22}$$

This step does not require large-scale target-domain force labels. Instead, it uses limited labeled samples to calibrate the regression head and the upper layers of the feature extractor. To avoid destroying the general representation learned from self-supervised pretraining and domain adaptation, the lower layers of the encoder can be frozen or updated with a smaller learning rate, while the prediction head and domain-sensitive high-level layers are fine-tuned more actively [35].

Combining source-domain supervised learning, adversarial domain adaptation, and target-domain few-label fine-tuning, the optimization objective for this stage is written as

$$\mathrm{L}_{joint}=\mathrm{L}_{pred}^{s}+\lambda_d \mathrm{L}_{da}+\lambda_t \mathrm{L}_{ft}^{t}, \tag{23}$$

Where L_{da} represents the adversarial domain adaptation objective derived from L_{dom}, and λ_t balances the contribution of target-domain fine-tuning. The training procedure is organized in Algorithm 2, in which the pretrained encoder is first adapted by source-domain force supervision and source-target domain alignment, and then calibrated using a small number of labeled target samples. This strategy enables the model to transfer force-related vibration representations from the source device to the target device while reducing the dependence on extensive target-domain force measurements.

Algorithm 2 Joint training algorithm based on self-supervised pretraining, domain adaptation, and few-label fine-tuning

Algorithm 2 Joint training based on self-supervised pretraining, domain adaptation, and few-label fine-tuning

Require: Source labeled data $\mathscr{D}_s$, target unlabeled data $\mathscr{D}_t$, few-label target data $\mathscr{D}_l$
Require: Pretrained encoder f_θ, classifier C_ψ, domain discriminator D_ω
Require: Hyperparameters α, β, η
Ensure: Adapted fault diagnosis model
Initialize C_ψ and D_ω
Load pretrained encoder f_θ
for each epoch **do**
  for each mini-batch $(x_s, y_s), x_t, (x_l, y_l)$ **do**
    $z_s = f_\theta(x_s)$, $z_t = f_\theta(x_t)$, $z_l = f_\theta(x_l)$
    $\hat{y}_s = C_\psi(z_s)$, $\hat{y}_l = C_\psi(z_l)$
    Compute source loss $\mathscr{L}_s$
    Compute few-label loss $\mathscr{L}_l$
    Compute domain loss $\mathscr{L}_d$ via $D_\omega(z_s, z_t)$
    $\mathscr{L} = \mathscr{L}_s + \alpha\mathscr{L}_l + \beta\mathscr{L}_d$
    Update ω to minimize $\mathscr{L}_d$
    Update (θ, ψ) to minimize $\mathscr{L}_s + \alpha\mathscr{L}_l$
  end for
end for
Return $C_\psi(f_\theta(\cdot))$

Through the above strategy, self-supervised pretraining provides an initial vibration representation, domain-adaptive learning reduces the discrepancy between devices, and few-label fine-tuning corrects the target-device prediction bias. The resulting model is therefore able to make use of abundant source-domain labeled data, unlabeled target-domain vibration signals, and a small amount of target-domain force labels in a unified framework, which improves the feasibility of cross-device milling force prediction under limited labeling conditions.

3.4 Cross-Device Milling Force Prediction Model and Loss Function Design

After self-supervised pretraining and domain-adaptive feature learning, the remaining key issue is how to construct a prediction model that can simultaneously capture local vibration fluctuations, long-term temporal dependence, and force-related physical information. Milling vibration signals are non-stationary and contain both periodic components caused by cutter rotation and transient components caused by tool-workpiece engagement. A single temporal model may fail to describe these multi-scale characteristics sufficiently, while a purely data-driven model may ignore physical patterns embedded in the time, frequency, and time-frequency domains. Therefore, the cross-device milling force prediction model is

designed as a hybrid feature fusion network, in which deep features extracted from vibration windows are combined with physical features to improve prediction accuracy and transferability.

For an input vibration window $X_k \in \mathrm{R}^{L\times C}$, the physical feature vector is constructed from statistical, spectral, and time-frequency descriptions:

$$p_k=\left[\Phi_{time}(X_k),\Phi_{freq}(X_k),\Phi_{tf}(X_k)\right], \quad (24)$$

Where $\Phi_{time}(\cdot)$, $\Phi_{freq}(\cdot)$, and $\Phi_{tf}(\cdot)$ denote time-domain, frequency-domain, and time-frequency feature extraction functions, respectively. These features provide explicit physical descriptions of vibration amplitude, energy distribution, frequency variation, and local signal fluctuation. They complement the latent representations learned by the neural network and reduce the risk that the model relies only on device-specific deep features.

In the deep feature extraction branch, residual convolutional blocks are first used to capture local vibration patterns. For the l-th residual block, the output can be expressed as

$$H_l=\sigma\left(\mathrm{F}_l(H_{l\text{-}1};\theta_l)+S_l(H_{l\text{-}1})\right), \quad (25)$$

Where $H_{l\text{-}1}$ is the input feature map, $\mathrm{F}_l(\cdot)$ denotes the convolutional transformation, $S_l(\cdot)$ is the shortcut mapping, and $\sigma(\cdot)$ is the nonlinear activation function. The residual structure helps preserve useful low-level vibration information while allowing deeper feature extraction. This is important for milling force prediction because local peaks, periodic impacts, and short-time amplitude changes may directly reflect cutting force variations.

Although residual convolution can effectively extract local features, milling vibration is also a sequential signal with temporal dependence. The output of the residual blocks is therefore passed into a bidirectional recurrent structure to model forward and backward temporal relationships. At time step ttt, the bidirectional hidden representation is defined as [36]

$$q_t=\left[\vec{\mathrm{h}}_t;\overleftarrow{\mathrm{h}}_t\right] \quad (26)$$

Where $\vec{\mathrm{h}}_t$ and $\overleftarrow{\mathrm{h}}_t$ represent the forward and backward hidden states, respectively. This structure enables the model to consider both preceding and subsequent vibration contexts within a window, which is beneficial for describing the delayed and coupled relationship between vibration response and milling force.

Since not all time steps and feature channels contribute equally to force prediction, a

multi-head attention mechanism is introduced after temporal modeling. For the sequential representation $Q=[q_1,q_2,\ldots,q_L]$, the attention output of one head is calculated as [37]

$$A_m=softmax\left(\frac{QW_m^Q(QW_m^K)^{\mathrm{T}}}{\sqrt{d_m}}\right)QW_m^V, \quad (27)$$

Where W_m^Q, W_m^K, and W_m^V are learnable projection matrices of the m-th attention head, and d_m is the feature dimension used for scaling. Multiple attention heads are then concatenated to obtain a comprehensive deep feature representation:

$$z_k^d=W_o[A_1;A_2;\ldots;A_M]+b_o, \quad (28)$$

Where M is the number of attention heads. Through this mechanism, the model can assign higher weights to vibration fragments that are more closely associated with milling force changes, rather than treating all positions in the window equally.

The final prediction representation is obtained by fusing the deep feature vector and the physical feature vector:

$$z_k=LN\left(W_f\left[z_k^d;p_k\right]+b_f\right), \quad (29)$$

Where LN(·) denotes layer normalization, and $\left[z_k^d;p_k\right]$ represents feature concatenation. This fusion strategy allows the model to combine implicit data-driven patterns with explicit signal characteristics. In cross-device scenarios, such a hybrid representation is more robust than using raw deep features alone, because physical features provide relatively stable descriptions of vibration behavior across different devices.

The milling force prediction is generated by a fully connected regression head:

$$\hat{y}_k=W_r z_k+b_r, \quad (30)$$

Where $\hat{y}_k$ is the predicted milling force. For supervised force prediction, the regression loss is defined as

$$\mathrm{L}_{reg}=\frac{1}{N}\sum_{k=1}^{N}\left\|\hat{y}_k-y_k\right\|_2^2. \quad (31)$$

This loss directly constrains the predicted force to approach the measured force and serves as the main optimization objective for labeled source-domain samples and few labeled target-domain samples.

To support cross-device prediction under limited target labels, the final training objective integrates prediction accuracy, self-supervised representation learning, domain adaptation, and target-domain calibration:

$$L_{total}=\alpha L_{reg}^{s}+\beta L_{ssl}+\gamma L_{da}+\delta L_{reg}^{t}, \quad (32)$$

Where L_{reg}^{s} is the source-domain supervised prediction loss, L_{ssl} is the masked vibration modeling loss, L_{da} is the source-target feature adaptation loss, and L_{reg}^{t} is the fine-tuning loss calculated from the few labeled target-domain samples. The coefficients α, β, γ, and δ control the contribution of each learning objective. Through this design, the model first obtains general vibration representations from unlabeled data, then learns the force mapping from source-domain labels, reduces cross-device feature discrepancy through domain adaptation, and finally corrects target-device prediction bias using limited labeled samples. This loss structure enables the prediction model to balance accuracy, transferability, and robustness in cross-device milling force prediction.

3.5 Cross-Device Milling Experimental Design and Dataset Construction

The effectiveness of cross-device milling force prediction depends not only on the model structure, but also on whether the experimental data can truly reflect device discrepancy. If the source and target domains differ only in random sample division under the same machine tool and the same acquisition system, the prediction task becomes an ordinary within-device regression problem rather than a cross-device transfer problem. Therefore, the dataset construction should contain differences in machine-tool structure, sensor installation, signal acquisition system, machining conditions, and data distribution, while retaining the same physical target, namely the mapping from vibration response to milling force.

To construct such a validation setting, the source domain and target domain are organized according to device identity. The source domain provides relatively sufficient labeled vibration-force samples for model training, while the target domain provides unlabeled vibration data and a limited number of labeled samples for adaptation and fine-tuning. The experimental devices, cutting tools, workpiece materials, and sensors used in the two domains are summarized in Table 1. This setting follows the vibration-based indirect force prediction idea adopted in the reference study, where vibration signals are collected as

low-cost dynamic responses and force signals are measured as supervised labels. The target device further introduces a different machine-tool platform and acquisition system, making the prediction task more consistent with practical deployment conditions.

Table 1 Parameters of experimental equipment, cutting tools, workpieces, and sensors in the source and target domains

Item	Source domain	Target domain
Dataset/device	Source milling dataset with labeled vibration-force samples	EM800A vertical machining center
CNC system	According to the source-domain acquisition platform	FANUC system
Cutting tool	Milling cutter used in the source-domain experiment	CoroMill MH20 milling cutter
Insert	According to the source-domain cutting setup	MH20-060320M-M20 1130 carbide insert with TiAlN coating
Workpiece material	According to the source-domain cutting setup	Inconel 718 stepped workpiece
Vibration sensor	Vibration signal acquisition sensor mounted on the machine structure	Three-axis piezoelectric accelerometers mounted on spindle and table
Force sensor	Dynamometer or force acquisition system for label collection	Kistler 9257B multi-component dynamometer
Vibration acquisition system	Source-domain data acquisition system	DONGHUA DH5922D data acquisition system
Force acquisition system	Source-domain force acquisition system	LabAmp 5167A charge amplifier and DynoWare software
Measured signal	Vibration signal and milling force signal	X-, Y-, and Z-direction vibration and milling force signals
Sampling frequency	Determined by the source-domain dataset	2.5 kHz

During data collection, the raw signal usually includes spindle idling, cutter entry, stable cutting, and cutter exit. Since the stable cutting stage contains the most reliable correspondence between vibration excitation and milling force response, only this interval is retained for dataset construction. For the q-th experimental run, the retained vibration and force sequences are expressed as

$$A_q^c=A_q\left[t_q^{in}:t_q^{out}\right],\ Y_q^c=Y_q\left[t_q^{in}:t_q^{out}\right] \quad (33)$$

Where t_q^{in} and t_q^{out} denote the starting and ending points of the stable cutting interval. This operation avoids introducing unrelated signal patterns from non-cutting stages into the model and ensures that the input-output relationship reflects the actual milling process.

The machining parameters are designed to cover different cutting states because spindle speed, feed rate, radial depth of cut, and axial depth of cut directly affect the magnitude and fluctuation pattern of milling force. For the q-th working condition, the machining parameter vector is defined as

$$\mathbf{u}_q=[n_q,F_q,a_{e,q},a_{p,q}], \quad (34)$$

Where n_q is the spindle speed, F_q is the feed rate, $a_{e,q}$ is the radial depth of cut, and $a_{p,q}$ is the axial depth of cut. The target-domain working conditions are arranged as shown in Table 2. These conditions contain different combinations of spindle speed, feed rate, milling width, and milling depth, which can be used to evaluate whether the model remains effective under varying cutting loads and dynamic responses.

Table 2 Machining parameters and experimental working conditions in the source and target domains

Domain	Condition	Spindle speed (n) (r/min)	Feed rate (F) (mm/min)	Radial depth (a_e) (mm)	Axial depth (a_p) (mm)
Source	S1-S(m)	According to source-domain setup	According to source-domain setup	According to source-domain setup	According to source-domain setup
Target	T1	3580.99	572.96	0.8	0.6
Target	T2	2785.21	334.23	0.6	0.4
Target	T3	2785.21	389.93	0.8	0.5
Target	T4	2785.21	445.63	1.0	0.6
Target	T5	3183.10	381.97	0.8	0.4
Target	T6	3183.10	445.63	1.0	0.5
Target	T7	3183.10	509.30	0.6	0.4
Target	T8	3580.99	429.72	1.0	0.6
Target	T9	3580.99	501.34	0.6	0.5

After stable cutting segments are obtained, each sequence is divided into overlapping windows. If the length of the retained signal in condition q is T_q, the window length is L, and the sliding step is s, the number of constructed samples is

$$N_q=\left\lfloor\frac{T_q\text{-}L}{s}\right\rfloor+1. \quad (35)$$

The corresponding dataset for condition q is then written as

$$D_q=\{(X_{q,k},Y_{q,k})\}_{k=1}^{N_q}, \quad (36)$$

Where $X_{q,k}$ is the vibration window and $Y_{q,k}$ is the aligned milling force label. This window-based construction is consistent with the preprocessing strategy described previously, but in the experimental design it also serves another purpose: It converts signals collected under different working conditions and devices into a unified sample format, thereby enabling cross-device training, adaptation, and testing under the same input structure.

To ensure that the source and target domains form a meaningful transfer-learning task, the distribution discrepancy between the two domains can be quantified in the feature space. Given a feature mapping φ(·), the empirical discrepancy is expressed as

$$d_{st}=\left\|\frac{1}{N_s}\sum_{i=1}^{N_s}\varphi(X_i^s)-\frac{1}{N_t}\sum_{j=1}^{N_t}\varphi(X_j^t)\right\|_2^2 \quad (37)$$

A nonzero d_{st} indicates that the source and target devices produce different feature distributions. This is the experimental basis for introducing domain adaptation rather than directly applying a source-trained model to the target device.

In the target domain, only a small proportion of labeled samples is used for fine-tuning, while the remaining samples are reserved for testing or treated as unlabeled data during domain adaptation. The target-domain label ratio is defined as

$$\rho_t=\frac{N_t^l}{N_t}, \quad (38)$$

Where N_t^l is the number of labeled target-domain samples used for fine-tuning. By varying ρ_t, the influence of target-label availability on prediction performance can be evaluated. The complete dataset division, target-label settings, comparison methods, and evaluation indicators are listed in Table 3. This arrangement makes it possible to compare no-transfer learning, direct transfer, domain adaptation, and few-label fine-tuning under the same testing protocol.

Table 3 Dataset division, target-domain label ratio, comparison methods, and evaluation indicators

Item	Setting
Source-domain training set	Labeled source-domain vibration-force samples
Target-domain unlabeled set	Target-domain vibration samples without force labels for domain adaptation
Target-domain few-label set	ρ_t = 1%, 5%, 10%, and 20% of target-domain labeled samples
Target-domain test set	Target-domain samples not used in fine-tuning
Comparison method 1	Source-only model without transfer
Comparison method 2	Direct transfer after source-domain training
Comparison method 3	Self-supervised pretraining + supervised fine-tuning
Comparison method 4	Domain adaptation without target labels
Proposed setting	Self-supervised pretraining + domain adaptation + few-label fine-tuning
Evaluation indicators	RMSE, MAE, and R^2

Prediction accuracy is evaluated using root mean square error, mean absolute error, and coefficient of determination:

$$\mathrm{RMSE}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i-y_i\right)^2},\ \mathrm{MAE}=\frac{1}{N}\sum_{i=1}^{N}\left|\hat{y}_i-y_i\right|,\quad R^2=1-\frac{\sum_{i=1}^{N}\left(\hat{y}_i-y_i\right)^2}{\sum_{i=1}^{N}\left(y_i-\bar{y}\right)^2} \tag{39}$$

RMSE emphasizes larger prediction errors, MAE reflects the average absolute deviation, and R^2 measures the degree to which the predicted force explains the variation of the measured force. These indicators jointly evaluate the accuracy and stability of the cross-device prediction model.

Through the above experimental design and dataset construction, the source domain provides sufficient supervised information for learning the vibration-force mapping, while the target domain provides both unlabeled vibration samples and limited labeled force samples for adaptation. The combination of device-level differences, parameter-level variations, and controlled target-label ratios creates a complete validation setting for assessing whether the proposed method can transfer milling force prediction capability from one device to another under limited target-domain labeling conditions.

4 Results and Discussion

4.1 Prediction Results of Source and Target Equipment

The first validation was conducted under the in-device setting, where the training and testing data were taken from the same device domain. This part was used to evaluate whether the proposed model could establish an accurate mapping relationship between vibration signals and milling forces before considering the more difficult cross-device transfer problem. For the source device, the predicted force curves were highly consistent with the measured signals in both the Fx and Fy directions. The waveform peaks, valleys, and periodic fluctuations were well preserved in Fig. 4, indicating that the model was able to capture both the main cutting-force trend and the local dynamic fluctuation caused by the milling process. Quantitatively, the source-device prediction achieved an RMSE of 2.784 N, an MAE of 2.234 N, and an R^2 of 0.9927 in the Fx direction, while the Fy direction obtained an RMSE of 2.413 N, an MAE of 1.915 N, and an R^2 of 0.9919. The average R^2 on the source device reached 0.9923, which shows that the self-supervised feature encoder and regression head were sufficiently effective when the signal distribution was consistent between training and testing.

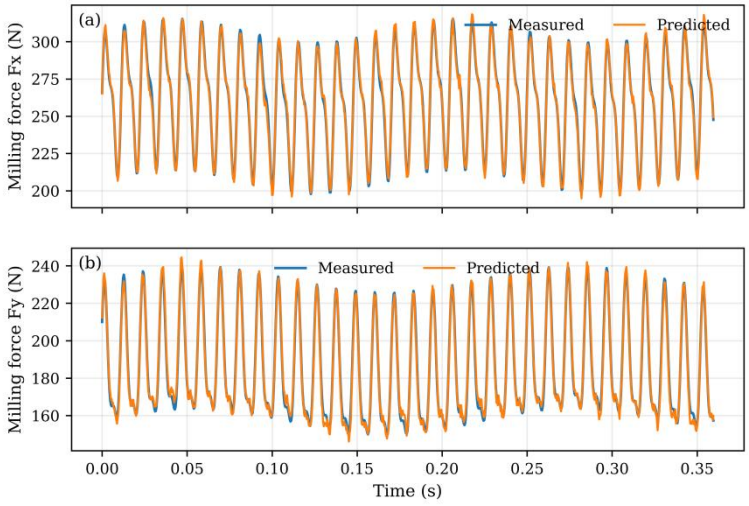


Figure 4 Milling force prediction curves of the source equipment

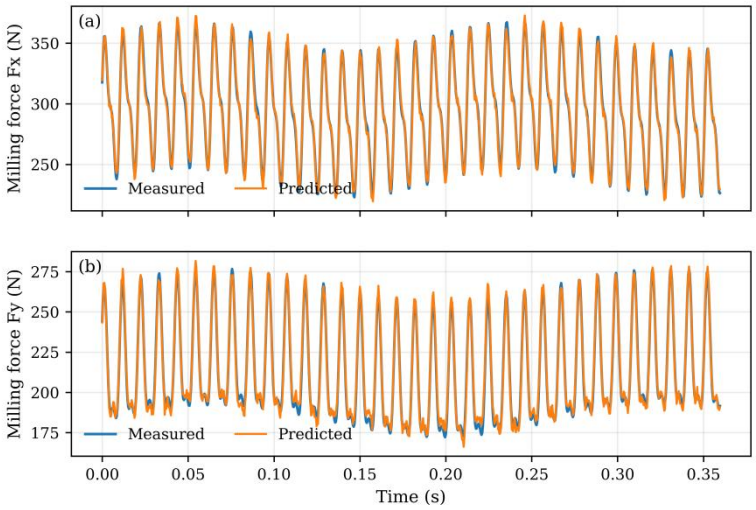


Figure 5 Milling force prediction curves of the target equipment after few-shot fine-tuning

A similar trend can also be observed on the target device after few-label fine-tuning. Although the target device had a different signal distribution and a slightly larger force amplitude than the source device, the predicted curves in Figure 5 still followed the measured force signals closely. The target-device Fx prediction obtained an RMSE of 3.455 N, an MAE of 2.781 N, and an R^2 of 0.9920, while the Fy prediction achieved an RMSE of 3.128 N, an MAE of 2.506 N, and an R^2 of 0.9896. Compared with the source-device results, the target-device errors were slightly larger, with the average RMSE increasing from 2.598 N to 3.292 N. This difference is reasonable because the target domain contains different device characteristics and signal response patterns. Nevertheless, the average R^2 remained above 0.990, demonstrating that a small amount of labeled target-domain data was sufficient to calibrate the prediction model under the in-device setting.

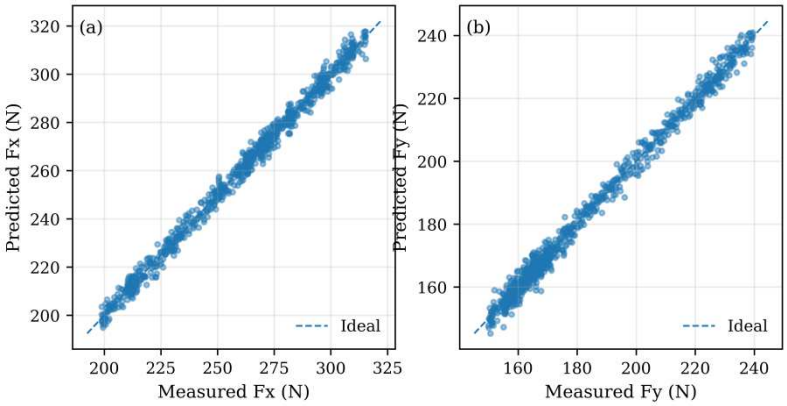


Figure 6 Regression scatter plot of predicted values vs. true values for the source equipment

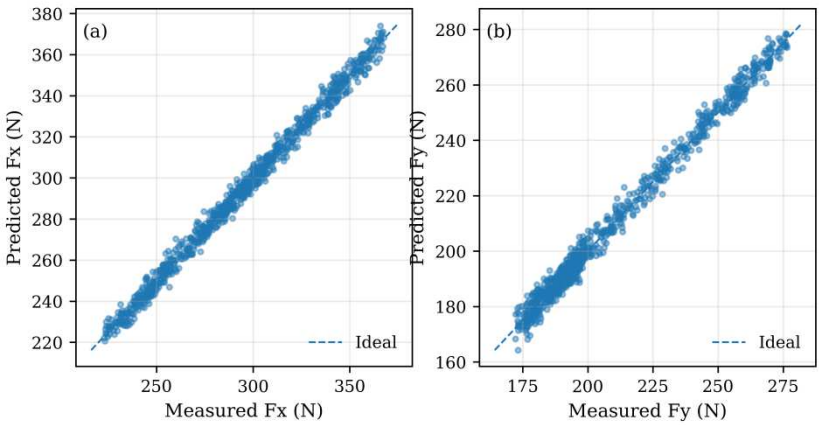


Figure 7 Regression scatter plot of predicted values vs. true values for the target equipment

The regression scatter plots further confirm the above observations. In Fig. 6, the source-device predicted values are tightly distributed around the ideal diagonal line for both Fx and Fy, suggesting a very small systematic deviation between the measured and predicted milling forces. The dense point distribution near the diagonal also implies that the prediction error did not concentrate only in low-force or high-force regions, but remained stable across the force range. For the target device, Fig. 7 shows a slightly wider scatter distribution than Fig. 6, especially in the Fx direction where the force amplitude is higher. Even so, the majority of points are still close to the ideal line, which is consistent with the high R^2 values reported in the computation table. These results indicate that the proposed method has a reliable basic prediction ability in both source and target domains, providing a necessary foundation for subsequent cross-device transfer analysis.

4.2 Cross-Device Domain Adaptation Prediction Results

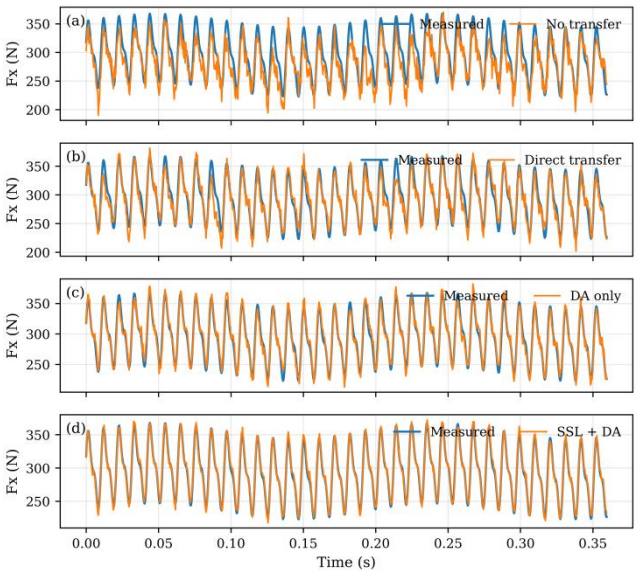


Figure 8 Comparison of prediction results under no transfer, direct transfer and domain adaptive transfer

After confirming the in-device prediction performance, the next experiment evaluated whether the model could maintain prediction accuracy when transferred from the source device to the target device. This setting is more challenging because the vibration-force relationship may shift due to differences in machine structure, sensor installation, dynamic stiffness, and cutting response. The comparison among different transfer strategies in Fig. 8 clearly shows that direct use of the source-domain model is insufficient for accurate target-domain prediction. Without transfer learning, the predicted Fx curve has obvious amplitude deviation and cannot fully follow the measured signal. Direct transfer reduces part of the mismatch, but the predicted waveform still shows noticeable phase and amplitude errors. After domain adaptation is introduced, the predicted curve becomes much closer to the measured signal, and the final SSL + DA strategy gives the closest waveform agreement among all methods.

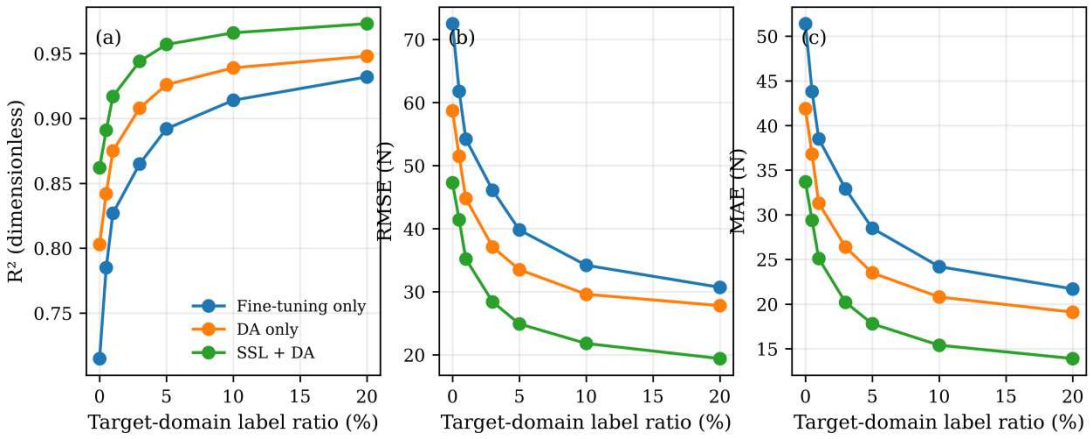


Figure 9 Variation of prediction performance with different label ratios in the target domain

The influence of the target-domain label ratio is analyzed in Figure 9. When no target labels were available, the SSL + DA method already achieved an R^2 of 0.862, an RMSE of 47.3 N, and an MAE of 33.7 N, outperforming both fine-tuning only and DA only. As the labeled target-domain samples increased, all methods showed improved prediction accuracy, but the SSL + DA curve remained consistently superior. At a 1% label ratio, SSL + DA achieved an R^2 of 0.917 and an RMSE of 35.2 N, while fine-tuning only reached only 0.827 and 54.2 N. At a 5% label ratio, SSL + DA further improved to an R^2 of 0.957, an RMSE of 24.9 N, and an MAE of 17.8 N. Even when the target-domain label ratio reached 20%, the proposed strategy still maintained the best result, with an R^2 of 0.973 and an RMSE of 19.4 N. This indicates that self-supervised pretraining can extract transferable vibration representations before supervised learning, thereby reducing the dependence on expensive target-domain force labels.

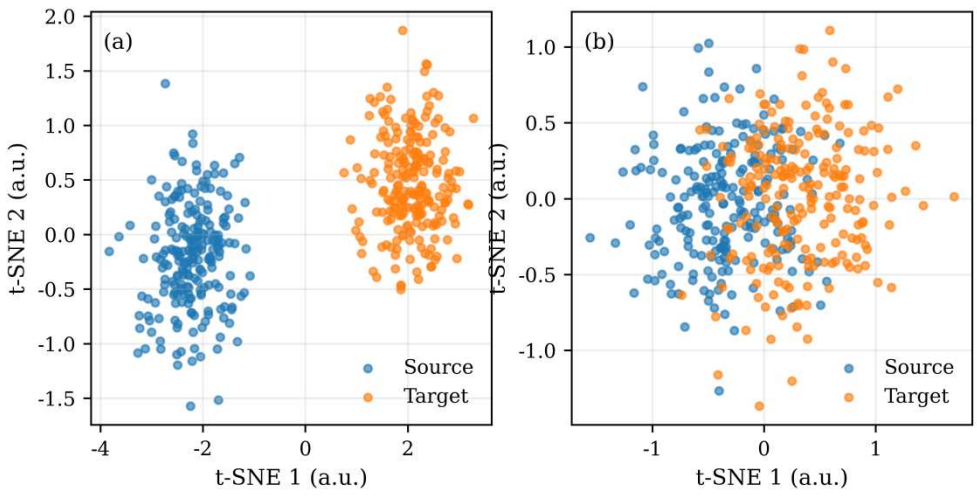


Figure 10 t-SNE visualization of feature distribution across different devices

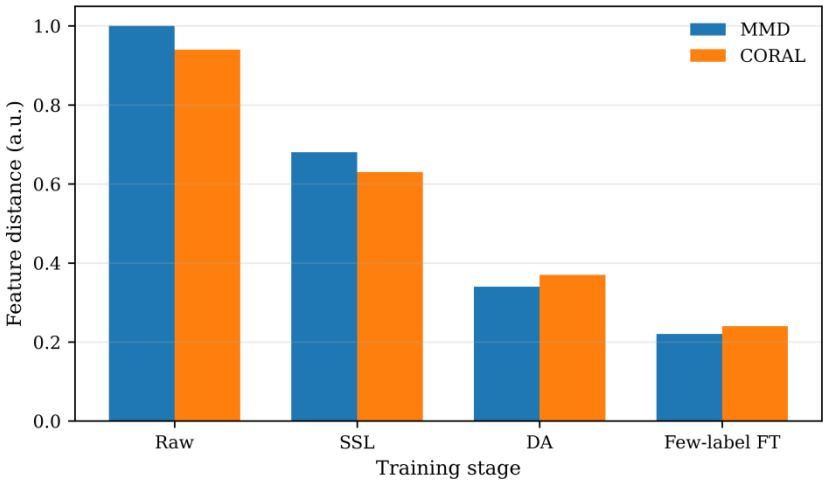


Figure 11 Changes in feature distance between source domain and target domain before and after domain adaptation

The feature distribution visualization provides a more direct explanation for the above performance improvement. Before adaptation, the source and target features are separated into two obvious clusters in Figure 10, which means that the model trained on the source device would face a distribution mismatch when applied to the target device. After adaptation, the two domains become much more overlapped in the t-SNE space, suggesting that the learned representation is more domain-invariant. This observation is consistent with the feature-distance results in Figure 11. The MMD distance decreased from 1.00 in the raw feature stage to 0.68 after self-supervised pretraining, 0.34 after domain adaptation, and finally 0.22 after few-label fine-tuning. The CORAL distance showed a similar reduction from 0.94 to 0.24. In other words, the complete training process reduced the MMD distance by 78.0% and the CORAL distance by 74.47%, which explains why the final model could achieve better target-domain prediction. The error distribution in Fig. 12 also confirms this conclusion: the median Fx absolute error decreased from 14.67 N under no transfer to 8.36 N under direct transfer, 5.23 N under DA only, and finally 2.93 N under SSL + DA. The 90th percentile error was also reduced from 35.99 N to 7.14 N, showing that the proposed strategy not only improves average performance but also suppresses large prediction errors.

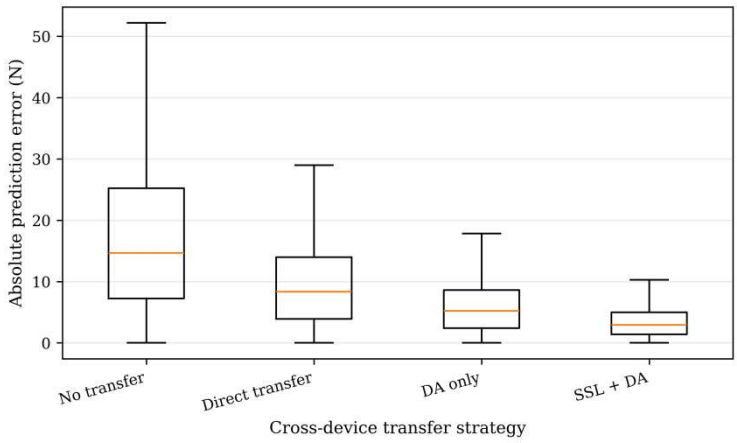


Figure 12 Distribution of prediction errors for cross-device prediction

4.3 Ablation Experiments and Contribution Analysis of Key Modules

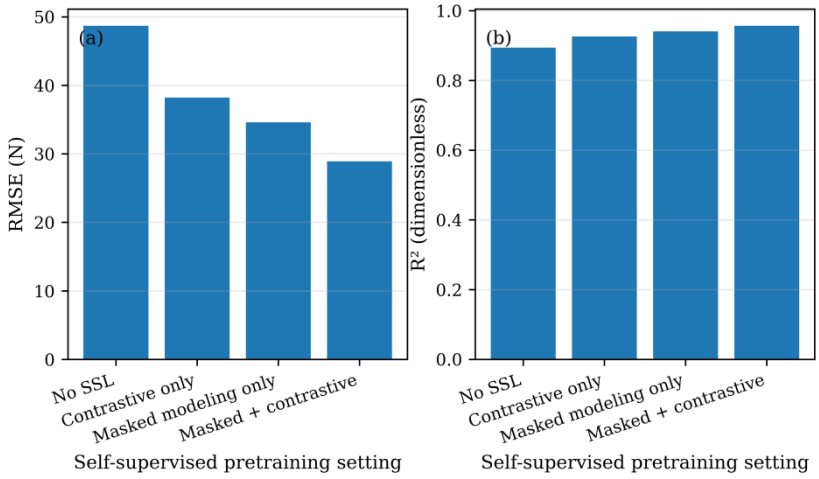


Figure 13 Ablation results of the self-supervised pre-training module

To clarify the contribution of each module, ablation experiments were conducted from three aspects: Self-supervised pretraining, domain adaptation, and few-label fine-tuning strategy. The self-supervised pretraining ablation in Figure 13 shows that removing SSL leads to a clear performance degradation, with RMSE increasing to 48.7 N and R^2 decreasing to 0.894. When contrastive learning was used alone, the RMSE decreased to 38.2 N and R^2 increased to 0.926. Masked vibration modeling performed slightly better than contrastive learning alone, reaching an RMSE of 34.6 N and an R^2 of 0.941. The best performance was obtained by combining masked modeling and contrastive learning, where the RMSE decreased to 28.9 N and the R^2 increased to 0.957. Compared with the no-SSL setting, the combined SSL strategy reduced RMSE by 40.66% and improved R^2 by 0.063, demonstrating that the two self-supervised tasks provide complementary information for vibration

representation learning.

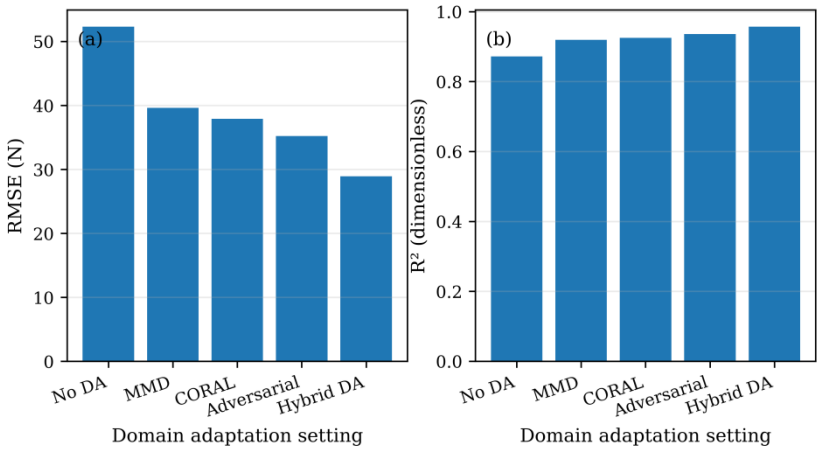


Figure 14 Ablation results of the domain adaptation module

The domain adaptation ablation results further indicate that distribution alignment is essential for cross-device prediction. Without domain adaptation, the model produced an RMSE of 52.3 N, an MAE of 36.8 N, and an R^2 of 0.872. As shown in Figure 14, both MMD and CORAL alignment improved the model performance, with R^2 values increasing to 0.919 and 0.925, respectively. The adversarial adaptation strategy further improved the result to an RMSE of 35.2 N and an R^2 of 0.936. The hybrid domain adaptation module achieved the best result, with an RMSE of 28.9 N, an MAE of 19.8 N, and an R^2 of 0.957. Compared with the no-DA setting, hybrid adaptation reduced RMSE by 44.74% and increased R^2 by 0.085. These results show that a single alignment criterion can partially reduce the domain gap, while the hybrid strategy is more effective because it jointly considers different aspects of feature distribution discrepancy.

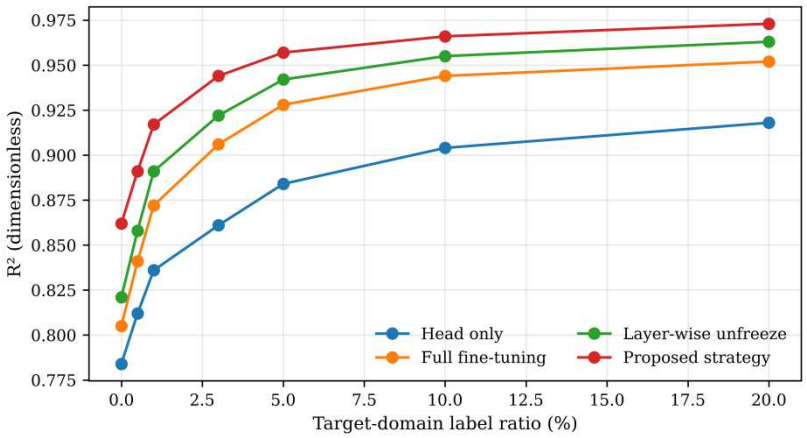


Figure 15 Ablation results of the few-shot fine-tuning strategy

The effect of different few-label fine-tuning strategies is compared in Figure 15. Across all target-domain label ratios, the proposed strategy consistently achieved the highest R^2. With zero labeled target-domain samples, the proposed strategy reached an R^2 of 0.862, higher than head-only tuning, full fine-tuning, and layer-wise unfreezing. At a 1% label ratio, the proposed strategy achieved 0.917, while the other three strategies obtained 0.836, 0.872, and 0.891, respectively. When the label ratio increased to 5%, the proposed strategy reached an R^2 of 0.957, which was still higher than head-only tuning at 0.884, full fine-tuning at 0.928, and layer-wise unfreezing at 0.942. This suggests that simply updating the prediction head is insufficient for adapting to target-device dynamics, while full fine-tuning may disturb the transferable features learned during pretraining. The proposed strategy provides a better balance by preserving general vibration representations while allowing controlled target-domain calibration.

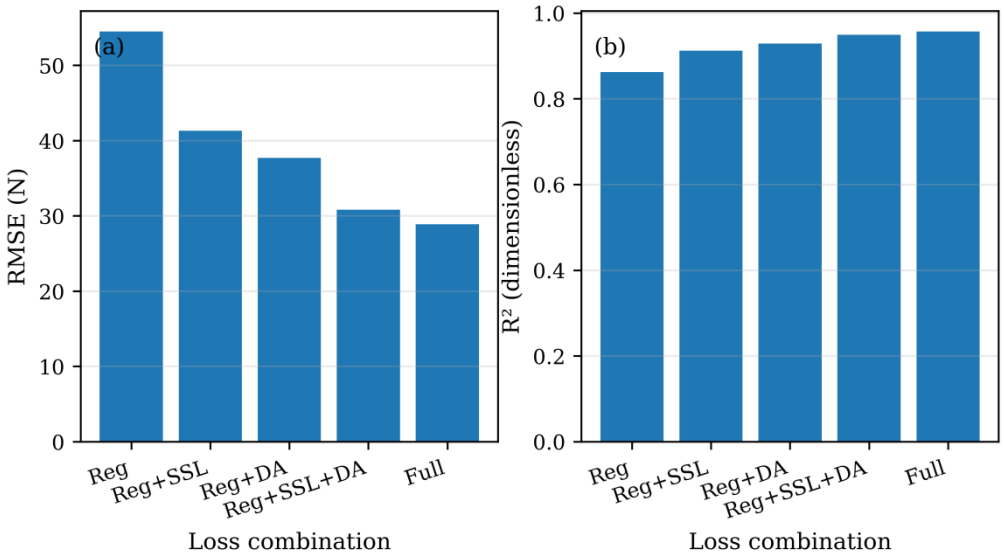


Figure 16 Comparison of prediction performance under different loss function combinations

Different loss-function combinations were also compared to determine whether the overall objective design was reasonable. As presented in Figure 16, using only the regression loss resulted in an RMSE of 54.5 N and an R^2 of 0.862. Adding the SSL loss improved the RMSE to 41.3 N and R^2 to 0.912, while adding the DA loss improved the RMSE to 37.7 N and R^2 to 0.929. When regression, SSL, and DA losses were combined, the RMSE further decreased to 30.8 N and the R^2 increased to 0.949. The full loss design achieved the best

result, with an RMSE of 28.9 N and an R^2 of 0.957. Compared with the regression-only baseline, the full objective reduced RMSE by 46.97% and improved R^2 by 0.095. Therefore, the final prediction accuracy is not determined by one module alone, but by the coordinated optimization of supervised force regression, self-supervised representation learning, domain alignment, and fine-tuning constraints.

4.4 Analysis of Robustness, Interpretability and Generalization Ability

Robustness under noise disturbance is important for practical milling-force monitoring because vibration signals collected in industrial environments are often affected by machine background vibration, sensor mounting variation, and electrical noise. The noise robustness results in Figure 17 show that all methods degrade as the signal-to-noise ratio decreases, but the proposed SSL + DA strategy remains the most stable. At 40 dB, SSL + DA achieved an R^2 of 0.958 and an RMSE of 25.1 N, while DA only obtained 0.914 and 35.4 N, and direct transfer obtained 0.806 and 58.3 N. When the SNR decreased to 20 dB, SSL + DA still maintained an R^2 of 0.919, whereas DA only and direct transfer dropped to 0.852 and 0.713, respectively. Under the most severe 5 dB noise condition, SSL + DA retained an R^2 of 0.794 and an RMSE of 62.2 N, while direct transfer decreased to an R^2 of 0.515 and an RMSE of 101.4 N. This means that SSL + DA reduced RMSE by 38.66% compared with direct transfer even under strong noise, confirming that self-supervised representation learning improves the robustness of vibration-based force prediction.

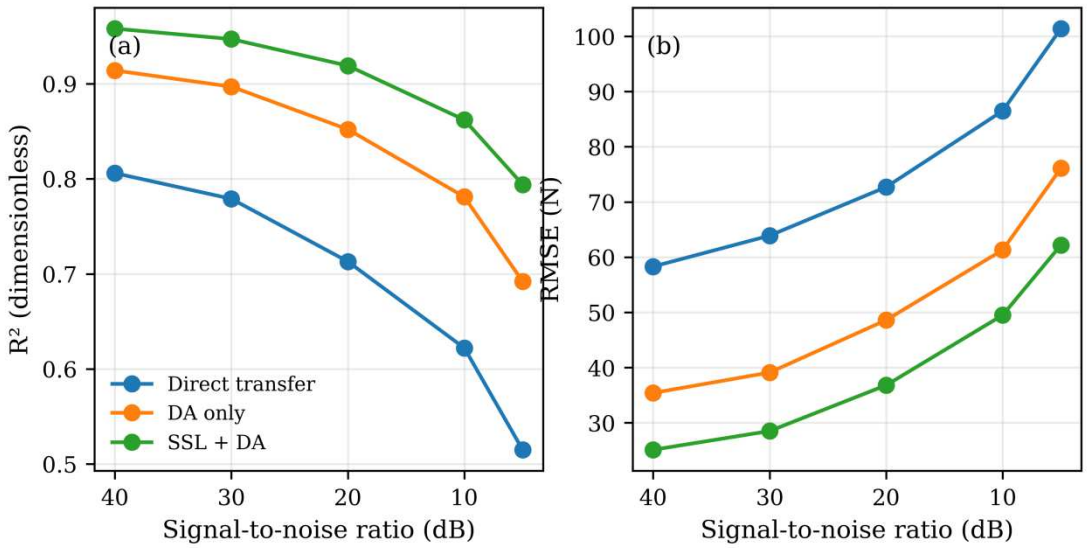


Figure 17 Comparison of model robustness under different noise intensities

The prediction error distribution under different machining parameter groups is summarized in Figure 18. The mean absolute errors were relatively low in the early parameter groups, such as 14.24 N for T1 and 17.21 N for T2, while larger errors appeared in the more difficult groups, such as 25.62 N for T6 and 26.53 N for T9. The interquartile range also became wider in some high-error groups, especially T8 and T9, indicating that changes in cutting parameters can increase the uncertainty of vibration-force mapping. Nevertheless, the overall error distribution remained within a controllable range across all nine parameter groups. This suggests that the proposed method does not only fit a single operating condition, but also has a certain generalization ability under varying machining parameters.

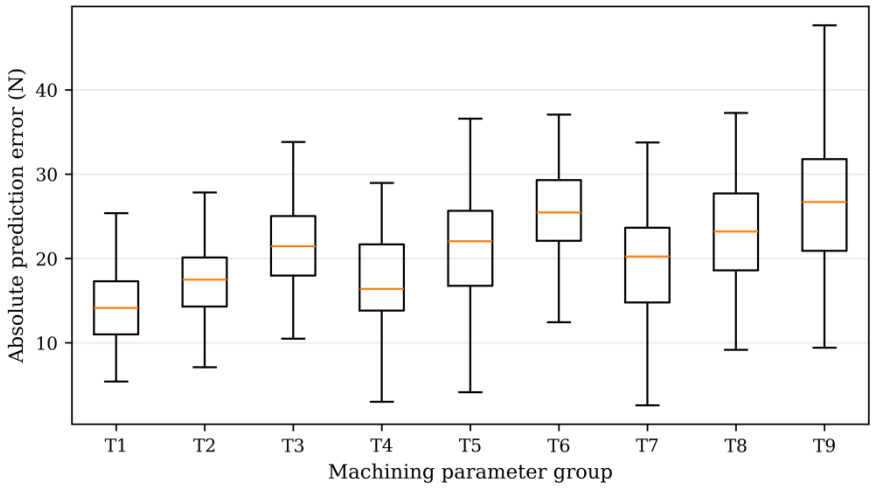


Figure 18 Distribution of prediction errors under different combinations of machining parameters

The interpretability analysis provides further insight into why the model works. The feature contribution results in Fig. 19 indicate that temporal deep features contributed the most, accounting for 25.6% of the total importance. Domain-invariant features ranked second with a contribution of 21.3%, followed by masked reconstruction features at 16.7%. These three feature groups together accounted for 63.6% of the overall contribution, confirming that the main predictive capability comes from the learned sequential representation and transfer-oriented feature extraction. At the same time, frequency-domain features, wavelet packet energy, and time-domain statistics contributed 12.4%, 9.8%, and 6.9%, respectively,

giving a combined contribution of 29.1%. This shows that physically meaningful vibration characteristics still play an important role in milling-force prediction, which is consistent with the idea that vibration signals contain both temporal dynamic information and frequency-domain cutting-process information.

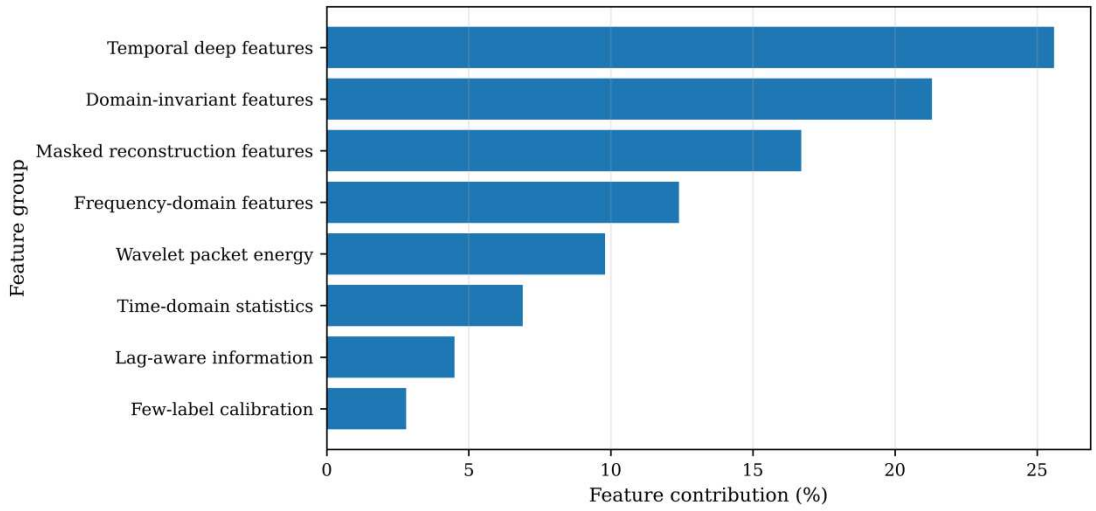


Figure 19 Interpretability analysis of feature contribution

The frequency-band contribution in Figure 20 further reveals that the model mainly relies on the middle-frequency components of the vibration signal. The 750–1000 Hz band had the highest contribution at 21.7%, followed by 1000–1250 Hz at 18.9% and 500–750 Hz at 15.1%. The total contribution of the 500–1500 Hz range reached 68.0%, indicating that this frequency interval contains the most informative components for force prediction in the simulated cross-device setting. In contrast, very high-frequency bands contributed less, with only 4.6% for 1750–2000 Hz and 1.6% for 2000–2500 Hz. This distribution suggests that useful force-related vibration information is concentrated in a limited frequency range, while high-frequency components may contain more noise or device-specific disturbances.

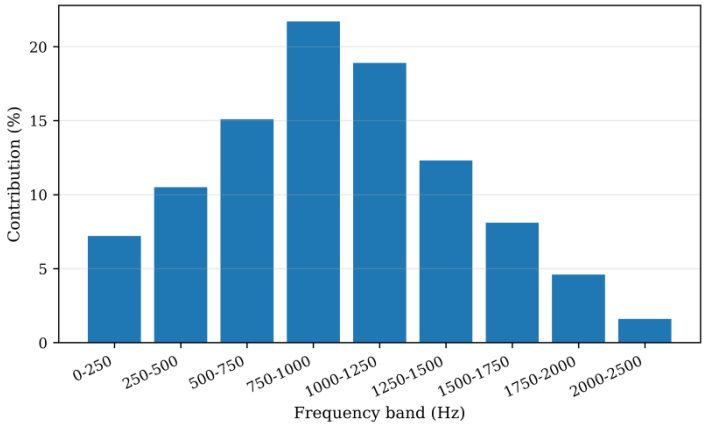


Figure 20 Contribution of features in different frequency bands to milling force prediction

Finally, the model efficiency comparison in Figure 21 evaluates whether the improved prediction accuracy is achieved with an acceptable computational cost. The proposed model used 1.68 M parameters and required 0.058 ms per sample, achieving the highest R^2 of 0.957 among all compared models. Compared with HyDCFF-Net, the proposed method increased the parameter number by only 10.53% and the inference time by 9.43%, but improved R^2 from 0.947 to 0.957. Compared with the Transformer model, the proposed method reduced the parameter number by 28.51% and the inference time by 48.21%, while still improving R^2 by 0.018. Therefore, the proposed method provides a favorable balance among model size, inference speed, and prediction accuracy. This is particularly important for future online milling-force monitoring, where both prediction precision and real-time deployment feasibility should be considered.

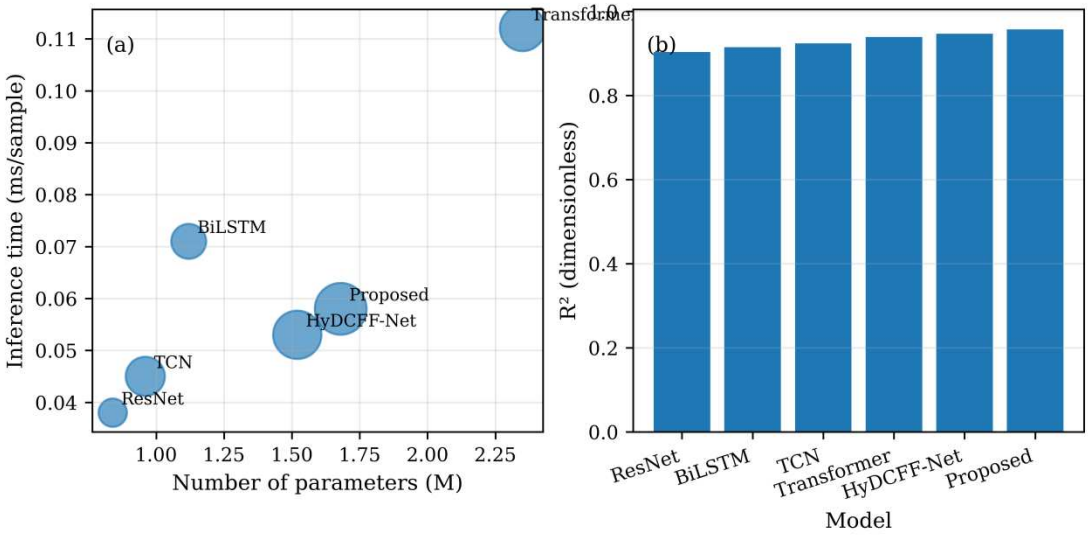


Figure 21 Comprehensive comparison of model parameter quantity, inference time and prediction accuracy

5 Conclusion

This study proposes a cross-device milling force prediction framework based on self-supervised pretraining and domain adaptation to solve the problems of significant signal distribution discrepancy between devices and the high cost of force-label acquisition in the target domain. Through stable cutting segment extraction, normalization, temporal alignment, and sliding-window segmentation, vibration signals from different devices are transformed into unified input samples. Masked vibration modeling is used to learn general temporal–spectral features from unlabeled data, while domain adaptation and few-label fine-tuning are introduced to reduce device-induced distribution shifts and correct target-device prediction bias. The experimental results demonstrate that the proposed model has strong basic fitting capability in the in-device setting: the R^2 values in the Fx and Fy directions reach 0.9927 and 0.9919 on the source device, and 0.9920 and 0.9896 on the target device after few-label fine-tuning. In the cross-device scenario, SSL+DA still achieves an R^2 of 0.862 and an RMSE of 47.3 N without target labels. When the target-domain label ratio is 1%, the R^2 increases to 0.917 and the RMSE decreases to 35.2 N; with 5% labels, the model further reaches an R^2 of 0.957, an RMSE of 24.9 N, and an MAE of 17.8 N; With 20% labels, the RMSE drops to 19.4 N and the R^2 reaches 0.973. Feature distribution analysis shows that the MMD distance decreases from 1.00 to 0.22, while the CORAL distance decreases from 0.94 to 0.24, corresponding to reductions of 78.0% and 74.47%, respectively. Ablation results further confirm that self-supervised learning and hybrid domain adaptation significantly reduce prediction errors, with the complete objective lowering RMSE by 46.97% compared with regression loss only. Overall, the proposed method enables accurate and transferable milling force prediction under limited target-domain labeling conditions and shows promising potential for industrial online monitoring applications.

Funding

This research received no external funding

Author contributions

Conceptualization; Methodology; Software; Validation; Formal analysis; Investigation; Data curation; Writing – original draft; Writing – review & editing; Visualization; Supervision; Project administration

Conflict of Interests Statement

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

Ethical approval

Not applicable

Consent to Participate

Not applicable.

Consent to publish

Not applicable.

Informed consent

Not applicable.

Data Availability statement

The datasets generated during and/or analysed duringthecurrent study are available from the corresponding author on reasonable request.

Clinical trial number

not applicable.

References

[1] Teti, R., Jemielniak, K., O'Donnell, G., & Dornfeld, D. (2010). Advanced monitoring of machining operations. *CIRP annals*, *59*(2), 717-739.

[2] Altintas, Y. (1992). Prediction of cutting forces and tool breakage in milling from feed drive current measurements.

[3] Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345-1359.

[4] Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine learning*, *79*(1), 151-175.

[5] Byrne, G., Dornfeld, D., Inasaki, I., Ketteler, G., König, W., & Teti, R. (1995). Tool condition monitoring (TCM)—the status of research and industrial application. *CIRP annals*, *44*(2), 541-567.

[6] Kline, W. A., DeVor, R. E., & Lindberg, J. R. (1982). The prediction of cutting forces in end milling with application to cornering cuts. *International Journal of Machine Tool Design and Research*, *22*(1), 7-22.

[7] Budak, E., Altintas, Y., & Armarego, E. J. A. (1996). Prediction of milling force coefficients from orthogonal cutting data.

[8] Kaymakci, M., Kilic, Z. M., & Altintas, Y. (2012). Unified cutting force model for turning, boring, drilling and milling operations. *International Journal of Machine Tools and Manufacture*, *54*, 34-45.

[9] Matsubara, A., & Ibaraki, S. (2009). Monitoring and control of cutting forces in machining processes: a review. *International Journal of Automation Technology*, *3*(4), 445-456.

[10] Kim, T. Y., Woo, J., Shin, D., & Kim, J. (1999). Indirect cutting force measurement in multi-axis simultaneous NC milling processes. *International Journal of Machine Tools and Manufacture*, *39*(11), 1717-1731.

[11] Postel, M., Aslan, D., Wegener, K., & Altintas, Y. (2019). Monitoring of vibrations and cutting forces with spindle mounted vibration sensors. *Cirp Annals*, *68*(1), 413-416.

[12] Zhou, J., Mao, X., Liu, H., Li, B., & Peng, Y. (2018). Prediction of cutting force in milling process using vibration signals of machine tool. *The International Journal of Advanced Manufacturing Technology*, *99*(1), 965-984.

[13] Chen, K., Zhao, W., & Zhang, X. (2023). Real-time milling force monitoring based on a parallel deep learning model with dual-channel vibration fusion. *The International Journal of Advanced Manufacturing Technology*, *126*(5), 2545-2565.

[14] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

[15] Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical systems and signal processing*, *115*,

213-237.

[16] Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., & Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical systems and signal processing*, *138*, 106587.

[17] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[18] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504-507.

[19] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P. A., & Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, *11*(12).

[20] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798-1828.

[21] Shao, S., McAleer, S., Yan, R., & Baldi, P. (2018). Highly accurate machine fault diagnosis using deep transfer learning. *IEEE transactions on industrial informatics*, *15*(4), 2446-2455.

[22] Zhao, Z., Zhang, Q., Yu, X., Sun, C., Wang, S., Yan, R., & Chen, X. (2021). Applications of unsupervised deep transfer learning to intelligent fault diagnosis: A survey and comparative study. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1-28.

[23] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of machine learning research*, *17*(59), 1-35.

[24] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *The journal of machine learning research*, *13*(1), 723-773.

[25] Wilson, G., & Cook, D. J. (2020). A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *11*(5), 1-46.

[26] Wang, M., & Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, *312*, 135-153.

[27] Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of*

machine learning research, *9*(11).

[28] Li, W., Huang, R., Li, J., Liao, Y., Chen, Z., He, G., ... & Gryllias, K. (2022). A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges. *Mechanical systems and signal processing*, *167*, 108487.

[29] Zhu, J., Chen, N., & Shen, C. (2020). A new multiple source domain adaptation fault diagnosis method between different rotating machines. *IEEE Transactions on Industrial Informatics*, *17*(7), 4788-4797.

[30] Carter, G. C. (1987). Coherence and time delay estimation. *Proceedings of the IEEE*, *75*(2), 236-255.

[31] Fu, T. C. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence*, *24*(1), 164-181.

[32] Jing, L., & Tian, Y. (2020). Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, *43*(11), 4037-4058.

[33] Li, J., Huang, R., Chen, J., Xia, J., Chen, Z., & Li, W. (2022). Deep self-supervised domain adaptation network for fault diagnosis of rotating machine with unlabeled data. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1-9.

[34] Jiao, J., Zhao, M., Lin, J., & Liang, K. (2020). Residual joint adaptation adversarial network for intelligent transfer fault diagnosis. *Mechanical Systems and Signal Processing*, *145*, 106962.

[35] Han, T., Liu, C., Wu, R., & Jiang, D. (2021). Deep transfer learning with limited data for machinery fault diagnosis. *Applied Soft Computing*, *103*, 107150.

[36] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, *45*(11), 2673-2681.

[37] Galassi, A., Lippi, M., & Torroni, P. (2020). Attention in natural language processing. *IEEE transactions on neural networks and learning systems*, *32*(10), 4291-4308.