Citation information: DOI: 10.1109/TIM.2025.3627341, IEEE Transactions on Instrumentation and Measurement

Joint Attention-Guided Multitask Feature Sharing Network for High-Speed Train Fault Diagnosis

Yihao Xue, *Student Member, IEEE*, Rui Yang, *Senior Member, IEEE*, Xiaohan Chen, *Student Member, IEEE*, Baoye Song, Zidong Wang, *Fellow, IEEE*

***Abstract*—Intelligent fault diagnosis of traction systems is vital for the reliability and safety of high-speed trains. Conventional methods extract features solely from fault signals to determine fault categories, neglecting the impact of operating conditions on traction systems. To address this limitation, multitask learning methods have been explored to simultaneously distinguish fault categories and operating conditions. However, due to the high cost of collecting high-speed train fault data, the available data are often extremely limited. Considering the parameter-intensive nature of multitask learning models and the scarcity of fault data, these models are prone to potential overfitting risks during the training process. In this work, we propose a novel joint attention-guided multitask feature sharing network (JA-MFSN) tailored for high-speed train traction system fault diagnosis. Our JA-MFSN integrates a novel joint attention module (JAM) that captures both task-shared and task-specific features with reduced parameter overhead, effectively mitigating overfitting risks. The network architecture balances model complexity and performance, enabling robust multitask learning under data-scarce conditions. Experimental results conducted on the hardware-in-the-loop (HIL) high-speed train traction control system simulation platform clearly demonstrate the superiority of the JA-MFSN approach over several existing methods.**

***Index Terms*—High-speed train, fault diagnosis, lightweight multitask learning, overfitting, joint attention, feature sharing.**

NOMENCLATURE

Symbol	**Description**
X	Set of input feature vectors fed into JAM.
S	The i-th feature vector in the set X.
$\acute{X}$	Sensitive feature vectors after attention weighting.
$\grave{X}$	Feature vectors after concatenation of sensitive and non-sensitive features.
Z	Evaluation weights used to decide the activation of attention mechanism.
$\breve{X}$	Feature vectors adjusted by evaluation weights.
$\tilde{X}$	Final output feature vector of the JAM module.
$F_1, F_3, F_7,$ F_8, F_9	1×1 convolution.
$F_2, F_4, F_5,$ F_6, F_{10}	One-dimensional convolution.
σ	Sigmoid function.
δ	SoftMax function.
$x^{(i)}$	The i-th input sample.
$y_A^{(i)}$	Label of the i-th sample for task A.
$y_B^{(i)}$	Label of the i-th sample for task B.
C_A, C_B	Number of classes for tasks A and B.
ϑ^A, ϑ^B	Model parameters for predicting tasks A and B.
$p(y_A^{(i)})$ $p(y_B^{(i)})$	Probability outputs for tasks A and B.
$L(\vartheta^A)$ $L(\vartheta^B)$	Loss functions for tasks A and B.
$L(\vartheta^A, \vartheta^B)$	Weighted sum of loss functions of both tasks.
α, β	Hyperparameters balancing losses of the two tasks.
η	Learning rate for updating model parameters.
κ	Number of stacked JAM modules.
ν	Total number of training samples.
$\Bbbk$	Indicator function (equals 1 if prediction matches true label, else 0).

I. INTRODUCTION

HIGH-speed train has increasingly become a popular mode of transportation worldwide, favored for its safety, punctuality, comfort, and efficiency [49]. Traction asynchronous motors drive the wheels through the electric power transmission system, directly impacting critical performance indicators such as the train's speed, acceleration, and traction force [46]. Therefore, monitoring and diagnosing faults in traction asynchronous motors, as key components of the propulsion system, are crucial for maintaining the overall safe operation of high-speed train [41].

In recent years, data-driven fault diagnosis has emerged as a popular research direction [11], [19]. Data-driven approaches do not require complex mathematical modeling of mechanical systems; instead, they learn fault patterns by analyzing data collected from mechanical equipment [14], [20]. Therefore, these methods exhibit strong adaptability and flexibility, offering advantages in handling complex mechanical fault diagnosis

This work is supported by the National Natural Science Foundation of China (62233012), the Jiangsu Provincial Scientific Research Center of Applied Mathematics (BK20233002), the Suzhou Science and Technology Programme (SYG202106) and the Jiangsu Provincial Qinglan Project (2021).

Yihao Xue is with the School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, China, 215123, and also with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, United Kingdom (email: Yihao.Xue21@student.xjtlu.edu.cn).

Rui Yang is with the School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, 215123, China (email: R.Yang@xjtlu.edu.cn).

Xiaohan Chen is with the KIOS Center of Excellence, University of Cyprus, Nicosia, 2109, Cyprus (email: chen.xiaohan@ucy.ac.cy).

Baoye Song is with the College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China (email: Songbaoye@sdust.edu.cn).

Zidong Wang is with the Department of Computer Science, Brunel University London, Uxbridge, Middlesex UB8 3PH, United Kingdom (email: Zidong.Wang@brunel.ac.uk).

Corresponding author: Rui Yang

tasks [7], [42]. Recently, data-driven approaches have also been widely deployed in various application scenarios, such as state estimation [24], [43], consensus control [2], [16], object tracking [9], [18], parameter optimization [29], [55], and fault diagnosis [5], [30], [31].

Although data-driven fault diagnosis has achieved notable success, traditional single-task learning models typically train independent models for each task (e.g., fault diagnosis or operating condition classification) using task-specific features. Each model learns to map input features to output labels, focusing on a single task at a time. However, this approach can be inefficient when tasks share underlying features or when data are limited, as the models are trained separately without leveraging the shared information [8], [10]. Moreover, conventional single-task models face challenges in simultaneously addressing both fault diagnosis and operating condition classification. In the absence or malfunctioning of sensors monitoring operating conditions, operators may lack critical information, leading to incorrect judgments during system failures and potentially resulting in unnecessary losses [33]. In contrast, multitask learning integrates multiple tasks into a unified model, enabling the joint learning of shared representations beneficial to all tasks. This not only enhances efficiency but also improves generalization by exploiting correlated data across tasks [23], [25]. Multitask models can concurrently perform fault diagnosis and operating condition classification, as these tasks share important features and signal patterns. The shared learning across tasks enhances robustness, particularly under conditions of limited fault data. Therefore, it is crucial to develop methods capable of handling different tasks simultaneously to manage potential fault scenarios in high-speed train traction systems.

Fortunately, researchers have proposed a series of feasible solutions by leveraging multitask learning techniques [34], [40]. By simultaneously learning multiple related tasks, multitask learning enables models to share features across tasks, significantly improving efficiency and generalization capabilities, while also allowing parallel processing of different tasks [50]. Specifically, multitask learning allows models to learn from the correlations between tasks by sharing feature representations that are beneficial to all tasks, thereby enhancing the model's robustness. As a result, some studies have attempted to use multitask learning to design efficient models that address both fault diagnosis and operating condition recognition tasks simultaneously. However, existing methods still face several unresolved limitations. Typically, high-speed train data collection involves high costs and risks, making it difficult to obtain sufficient data to train complex multitask models [12], [15]. Furthermore, when a model is required to handle multiple tasks, it often involves a large number of parameters, which are prone to overfitting when the training data is limited. Overfitting not only compromises the model's generalization ability but also may lead to diagnostic errors in real-world applications [39], [48]. Therefore, it is essential to improve existing multitask learning-based fault diagnosis methods by optimizing network structures and model parameters to mitigate potential overfitting issues.

To mitigate overfitting in multitask models, several studies have attempted to optimize model structure and improve computational efficiency primarily through two strategies: deploying lightweight models and incorporating attention modules [22], [27], [44]. Specifically, lightweight design not only enhances computational efficiency but also effectively prevents overfitting on small datasets [3], [47]. In multitask learning, lightweight models reduce redundant network parameters, enabling each task to focus on the most relevant features [44]. This approach retains the task-specific features for each task while leveraging shared features to enhance the model's learning ability [21]. Therefore, lightweight design is a necessary approach to address overfitting in multitask learning models. However, most existing methods achieve parameter reduction by directly removing or decomposing components of the network, often resulting in coarse-grained reduction strategies. These approaches may overlook the sensitivity of certain features, thus diminishing the model's ability to capture essential fault-related representations. Furthermore, some lightweight architectures may weaken the interactions between channels, potentially affecting the extraction of complex and subtle fault-related features.

Given their effectiveness in highlighting salient features and improving computational efficiency, attention modules have been widely adopted for optimizing model parameters [27], [32]. These modules are designed to capture task-relevant features using compact architectures, enabling the model to learn long-range dependencies and thereby alleviating overfitting [33], [38]. While attention mechanisms can preserve critical information and simultaneously reduce parameter counts, they may also result in the unintended loss of important features due to suboptimal attention configurations [4], [51]. This limitation stems from the fact that traditional attention mechanisms often apply fixed or static weights to features, without adequately adapting to the dynamic variations present in the input. Consequently, features that are essential but not prominent under current input conditions may be inadvertently downweighted or neglected, leading to degradation in representation quality. Moreover, conventional attention mechanisms generally lack the capability to autonomously determine whether attention should be activated based on the input characteristics. Such rigid attention strategies may lead to improper feature weighting and further contribute to potential information loss.

Based on the literature review, the deployment of lightweight models and the incorporation of attention mechanisms have demonstrated effectiveness in mitigating overfitting in high-speed train fault diagnosis tasks [22], [33]. Nevertheless, there remains significant potential for further improvement. One promising direction is the development of a novel multitask model with constrained parameter complexity, capable of reducing model size while maintaining robust and efficient feature extraction. Furthermore, the design of an advanced attention module could mitigate feature loss and enhance the model's ability to capture task-sensitive features under limited parameter conditions. The resulting improved multitask framework would offer the following advantages: 1) it enables the extraction of comprehensive fault-relevant features while alleviating overfitting induced by redundant parameters; and 2) it allows the model to dynamically adjust attention activations in response to variations in the input feature distribution, thereby avoiding the inappropriate suppression of critical features that often occurs with conventional attention mechanisms.

Based on the discussions above, this study aims to enhance the structure of the multitask model and optimize the attention module by proposing a novel lightweight multitask learning method for high-speed train fault diagnosis. Specifically, a novel lightweight joint attention-guided multitask feature sharing network (JA-MFSN) method is proposed, which can effectively integrate related tasks and extract general fault features. Additionally, a novel joint attention module (JAM) is proposed to capture task-relevant and fault-sensitive features across different tasks with limited parameters, while also mitigating the feature loss. Overall, the proposed JA-MFSN method can efficiently capture beneficial information from different yet related tasks and identify sensitive fault features under parameter-constrained conditions, thereby avoiding potential overfitting issue. The main contributions of this research are as follows:

1) A novel JAM is proposed to efficiently extract fault-sensitive features from different tasks with limited parameters while effectively mitigating the feature loss problem;
2) A novel JA-MFSN method is proposed for high-speed train fault diagnosis, aimed at effectively integrating fault signals from related tasks and extracting in-depth fault features with limited parameters, thereby preventing potential overfitting issue;
3) The proposed JA-MFSN method has been validated on the hardware-in-the-loop (HIL) high-speed train traction control system simulation platform, demonstrating the superiority of the proposed method over several other distinguished methods.

The remainder of this paper can be divided into three sections. Section II provides a detailed description of the proposed JA-MFSN method, including the overall structure of the JA-MFSN and JAM. In Section III, we meticulously analyze the experimental setup of this study along with the analysis of the corresponding results. Finally, Section IV concludes this study and presents future research directions.

II. Methodology

A. Joint Attention Module

In deep neural networks, attention mechanisms can adjust feature weights to enhance relevant features and reduce the influence of irrelevant ones, thus achieving efficient feature extraction. However, redundant attention mechanisms can mistakenly reduce the weights of crucial features, leading to feature loss and incomplete feature representation, which can degrade model performance. To alleviate these limitations, we propose a novel JAM that determines the necessity of activating the attention mechanism using weight evaluation coefficients. If these coefficients suggest a potential negative impact, the attention channel is blocked, reverting it to a regular convolutional module. This allows the stacking of multiple JAMs, enabling the model to automatically decide which attention modules to activate or suppress, thus preventing feature loss due to incorrect attention. Overall, the process of the JAM can be divided into five stages, and the framework of the JAM is illustrated in Fig. 1.

Stage 1. Attention Coefficient Computation: Let the input feature vectors to the JAM be denoted as $X = [x_1, x_2, \ldots, x_n]$. Initially, they undergo a 1×1 convolution for preliminary feature extraction, followed by a global average pooling (GAP) layer to perform averaging operations on the input fault feature vectors. The GAP layer can compress a one-dimensional feature vector into a single scalar value, thereby reducing multiple input vectors into a single vector Y. Each scalar value in the obtained vector Y retains the overall information of the input feature vector. Subsequently, Y is processed through a one-dimensional convolution and a 1×1 convolution for feature extraction and channel compression. Finally, the Sigmoid function is used to normalize each element in the output vector of the 1×1 convolution to a range of 0 to 1, thereby obtaining the attention coefficients S. Each element in S represents the importance of the corresponding feature vector. In summary, the computation process of *Stage 1* can be represented as follows:

$$S = \sigma\left(F_3\left(F_2\left(\mathrm{GAP}\left(F_1\left(X\right)\right)\right)\right)\right) \tag{1}$$

where F_1 and F_3 denote the 1×1 convolutions, F_2 denotes the one-dimensional convolution, and σ denotes the Sigmoid function. Additionally, the l-th element in S, denoted as s_l, represents the importance coefficient of the feature vector x_l.

Stage 2. Sensitive Feature Weighting: The obtained attention coefficients S can be utilized to weight the sensitive feature vectors, thereby enhancing the weights of momentous features and reducing the impact of irrelevant features. Specifically, the inputs X of JAM first undergo a 1×1 convolution, followed by processing through a one-dimensional convolution to extract deep features. Next, the extracted deep features are weighted using the attention coefficients S to enhance the significant features and suppress the influence of irrelevant features. Finally, a residual structure is incorporated to retain shallow fault features and prevent potential feature loss. The above process can be summarized as follows:

$$\acute{X} = F_4\left(F_1\left(X\right)\right) \odot S + F_1\left(X\right) \tag{2}$$

where F_4 denotes the one-dimensional convolution, and $\odot$ denotes the element-wise multiplication.

Stage 3. Non-Attention Feature Extraction: Since attention mechanisms may not always be effective, JAM incorporates a non-attention network flow to extract general fault features. This network flow consists of a one-dimensional convolution followed by a 1×1 convolution. The extracted non-attention fault features are concatenated with the sensitive fault features, effectively combining both sets of features. The concatenating operation ensures the temporary preservation of both attention and non-attention features. The above process can be summarized as follows:

$$\grave{X} = \mathrm{Concat}\left[F_7\left(F_6\left(F_1\left(X\right)\right)\right), F_5\left(\acute{X}\right)\right] \tag{3}$$

Citation information: DOI: 10.1109/TIM.2025.3627341, IEEE Transactions on Instrumentation and Measurement

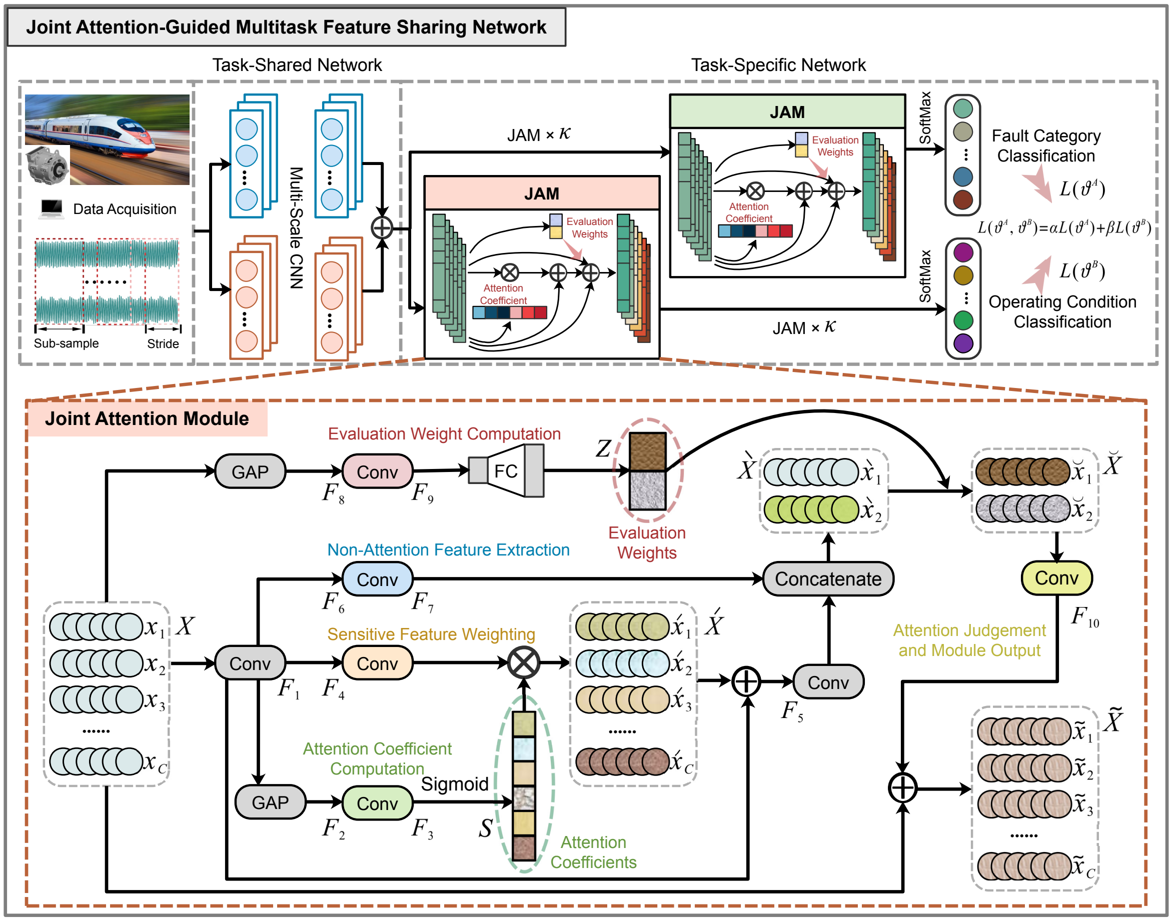

Fig. 1. Structure diagram of the proposed JA-MFSN method.

where F_5 and F_6 denote the one-dimensional convolutions, F_7 indicates the 1×1 convolution, and Concat denotes the concatenating operation.

Stage 4. Evaluation Weight Computation: In this stage, evaluation weights are computed to assess the necessity of the attention mechanism in JAM. These weights are utilized to determine whether the current module requires an attention mechanism, and the proportion of attention and non-attention network flows. Initially, the fault features X pass through a GAP layer to increase the receptive field and extract global information. Subsequently, the extracted features undergo two layers of 1×1 convolutions for non-linear transformations. Finally, evaluation weights are obtained through a fully-connected layer and SoftMax function. These weights are dynamically computed by the model based on input X, rather than being fixed values set manually. As input X changes, the evaluation weights are adjusted accordingly. Moreover, the SoftMax function ensures that the sum of evaluation weight values is constrained to 1, thereby normalizing the evaluation weights generated for the concatenated feature $\grave{X}$. The computation process of evaluation weights is illustrated as follows:

$$Z = \delta\left(\mathrm{FC}\left(F_9\left(F_8\left(\mathrm{GAP}\left(X\right)\right)\right)\right)\right) \tag{4}$$

where F_8 and F_9 denote the 1×1 convolutions, FC denotes the fully-connected layer, and δ denotes the SoftMax function. The resulting evaluation weights Znot only control the contribution ratio between attention-based and non-attention-based pathways but also allow the JAM module to autonomously decide whether to activate the attention mechanism based on the actual input features. This dynamic design effectively mitigates the unintended suppression of critical features commonly encountered in traditional attention modules.

Stage 5. Attention Judgement and Module Output: At this stage, we utilize the obtained attention evaluation weights Z to determine the contributions of attention and non-attention features. By performing element-wise multiplication, we combine the evaluation weights Z with the concatenated feature $\grave{X}$ to determine the weighted contributions of attention and non-attention feature vectors, as illustrated below:

$$\breve{X} = \grave{X} \odot Z \tag{5}$$

Upon determining the proportions of attention and non-attention features, the resulting features $\breve{X}$ undergo a one-dimensional convolution to further extract in-depth fault features. Subsequently, the extracted features are combined with the input X of the JAM to preserve the original features and prevent potential feature loss. Finally, the output of the JAM can be obtained as follows:

$$\tilde{X} = F_{10}\left(\breve{X}\right) + X \tag{6}$$

Overall, the proposed JAM effectively highlights sensitive fault features within the input feature vectors using attention mechanisms, while dynamically evaluating the proportions of attention and non-attention network flows, thereby avoiding redundant attention mechanisms.

B. Complexity Analysis

This subsection provides a comprehensive analysis of the computational complexity of the proposed JAM, in comparison with several well-known and effective attention mechanisms, namely non-local attention (NLA) [28], multi-head attention (MHA) [36], criss-cross attention (CCA) [13], the Swin Transformer module (STM) [53], and the Axial attention module (AAM) [17]. The complexity of these attention mechanisms is primarily determined by the number of model parameters and the computational cost associated with feature weighting operations. Table I presents a summary of the learnable parameters and floating-point operations (FLOPs) for each module, where C denotes the number of feature channels, L represents the sequence length, and W refers to the attention window size.

TABLE I
COMPLEXITY ANALYSIS OF DIFFERENT MODULES.

Module	Params (Big-O)	FLOPs (Big-O)
NLA	$O\left(C^2\right)$	$O\left(C^2L + CL^2\right)$
MHA	$O\left(C^2\right)$	$O\left(C^2L + CL^2\right)$
CCA	$O\left(C^2\right)$	$O\left(C^2L + CL\sqrt{L}\right)$
STM	$O\left(C^2\right)$	$O\left(C^2L + CLW\right)$
AAM	$O\left(C^2\right)$	$O\left(C^2L + CL\sqrt{L}\right)$
JAM (Ours)	$O\left(C^2\right)$	$O\left(C^2L\right)$

Specifically, the term C^2 characterizes interactions among channels within a module, while CL represents interactions between feature channels and sequence length. A complexity of CL^2 indicates a quadratic dependency on sequence length, whereas $CL\sqrt{L}$ typically arises from the adoption of dimensionality reduction or simplification strategies to mitigate computational overhead along the length dimension. The term CLW corresponds to local attention mechanisms, where computation is confined to features within a local window. As summarized in Table I, the proposed JAM module maintains a comparable number of parameters relative to existing attention mechanisms, while substantially reducing FLOPs, thereby demonstrating superior computational efficiency. This improvement is primarily attributed to JAM's innovative architecture, which dynamically determines the necessity of activating attention pathways based on learned weights. By selectively enabling or suppressing attention paths, JAM eliminates redundant computations that commonly encumber conventional attention mechanisms, thus achieving an optimal balance between feature extraction capability and model efficiency.

C. Joint Attention-Guided Multitask Feature Sharing Network

To achieve simultaneous classification of fault categories and operating conditions for high-speed train asynchronous traction motors, a novel JA-MFSN method is proposed in this study, mainly consisting of two parts with the detailed structure illustrated in Fig. 1: the task-shared network and the task-specific network. In the task-shared network, signals collected from the traction system are initially processed through multi-scale convolutional neural network (CNN) layers for in-depth fault feature extraction, capturing both local and global fault features. In the task-specific network, multiple JAM modules are stacked to extract task-related and fault-sensitive features while effectively preventing feature loss. The proposed JA-MFSN method can efficiently handle different classification tasks with limited model parameters, thus mitigating the overfitting problem commonly caused by parameter redundancy in conventional multitask models. The specific framework of the JA-MFSN method is illustrated in Fig. 1, with detailed information provided in Table II.

TABLE II
DETAILS OF THE JA-MFSN MODEL STRUCTURE.

Layer	Type	Filter	Kernel/ Stride	Activation Function	Output	Connection
			Task-Shared Network			
1	Input	/	/	/	$(N,1)$	/
2	1D-Conv	32	$64/2$	ReLU	$(N/2,32)$	1
3	1D-Conv	32	$32/2$	ReLU	$(N/4,32)$	2
4	1D-Conv	32	$16/2$	ReLU	$(N/8,32)$	3
5	1D-Conv	16	$8/1$	ReLU	$(N/8,16)$	4
6	1D-Conv	16	$9/2$	ReLU	$(N/2,16)$	1
7	1D-Conv	16	$5/2$	ReLU	$(N/4,16)$	6
8	1D-Conv	16	$3/2$	ReLU	$(N/8,16)$	7
9	Add	/	/	/	$(N/8,32)$	5, 8
			Task-Specific Network			
10	JAM$\times\kappa$	/	/	/	$(N/8,16)$	9
11	JAM$\times\kappa$	/	/	/	$(N/8,16)$	9
12	GAP	/	/	/	16	10
13	GAP	/	/	/	16	11
14	Output A	/	/	/	m	12
15	Output B	/	/	/	n	13

In the framework shown in Fig. 1, the collected fault signals are first processed by the task-shared network. This network incorporates a multi-scale CNN, which employs convolutional kernels of varying sizes to concurrently capture both local fine-grained details and broader global contextual information. This architectural choice enables the extraction of comprehensive feature representations, thereby improving the network's generalization capability across diverse fault categories and operating conditions. To ensure gradient stability, batch normalization (BN) is applied after each convolutional layer to normalize the extracted fault features. In the task-specific network, multiple JAMs are introduced to extract task-specific features and prevent potential feature loss, thereby facilitating high-performance classification for different tasks. The weight evaluation coefficients in JAM can effectively determine the activation level of attention mechanisms within the module, thereby avoiding unnecessary attention mechanisms that could negatively impact diagnostic performance.

D. Fault Diagnosis based on the Proposed Method

In traditional neural networks, where both mechanical component faults and operating conditions are input together, the model may treat these two tasks separately, failing to effectively share features between them. While conventional methods allow the network to learn features related to both tasks, they do not fully exploit the shared structure inherent in the tasks. For instance, fault diagnosis and operating condition classification may rely on similar patterns within the input features, but the network may not efficiently share these features across the tasks. This inefficiency can result in unnecessary complexity and an increased risk of overfitting, particularly when data are scarce. To address the limitations of existing methods, this study proposes a lightweight JA-MFSN method incorporating the JAM. The proposed method can effectively integrate fault signals from related tasks, enabling simultaneous classification of fault categories and operating conditions. The JAM can dynamically adjust focus between shared and task-specific features, allowing the model to efficiently share features across tasks while preventing feature loss. By enhancing the overall model structure and optimizing the attention module, the JA-MFSN method can capture the most relevant features for each task, mitigating redundancy and overfitting risks associated with traditional neural networks. The subsequent paragraphs will provide a detailed application of the JA-MFSN method to high-speed train fault diagnosis tasks.

Given the training set $\left\{x^{(i)}\right\}_{i=1}^{\nu}$, each sample corresponds to a label pair $\left\{y_A^{(i)}, y_B^{(i)}\right\}_{i=1}^{\nu}$ for two outputs A and B. Here, $y_A \in \{1, 2, \ldots, C_A\}$ and $y_B \in \{1, 2, \ldots, C_B\}$, where C_A and C_B represent the number of categories for output A and output B, respectively. The variable x represents an input fault signal sample, where each x is a one-dimensional time-series vector (i.e., a univariate fault signal). Typically, x is acquired by sensors within the high-speed train traction system and is preprocessed via normalization and segmentation. The notation $x^{(i)}$ refers to the i-th fault sample in the dataset, with the superscript (i) indicating the sample index. The symbol ν denotes the total number of samples in the training dataset. Thus, the probability of each fault type ϱ for output A can be computed as follows:

$$p\left(y_A^{(i)} = \varrho | x^{(i)}; \vartheta^A\right) = \frac{e^{(\vartheta_\varrho^A)^T x^{(i)}}}{\sum_{l=1}^{C_A} e^{(\vartheta_l^A)^T x^{(i)}}} \tag{7}$$

Similarly, the probability of each operating condition ς for output B can be computed as follows:

$$p\left(y_B^{(i)} = \varsigma | x^{(i)}; \vartheta^B\right) = \frac{e^{(\vartheta_\varsigma^B)^T x^{(i)}}}{\sum_{l=1}^{C_B} e^{(\vartheta_l^B)^T x^{(i)}}} \tag{8}$$

where $y_A^{(i)}$ and $y_B^{(i)}$ represent the ground truth labels for Task A (fault type classification) and Task B (operating condition classification), respectively, corresponding to the i-th sample. C_A denotes the number of fault categories, and C_B denotes the number of operating conditions. The symbols ϑ^A and ϑ^B denote the parameter sets (weights) specific to the models for predicting outputs A and B, respectively. The cross-entropy loss function is then employed to compute the losses for fault category and operating condition classification. The loss $L(\vartheta^A)$ for fault category classification is given by:

$$L\left(\vartheta^A\right) = -\frac{1}{\nu}\left[\sum_{i=1}^{\nu}\sum_{\varrho=1}^{C_A} \Bbbk\left\{y_A^{(i)} = \varrho\right\} \log p\left(y_A^{(i)} = \varrho | x^{(i)}; \vartheta^A\right)\right] \tag{9}$$

Similarly, the loss $L(\vartheta^B)$ for operating conditions can be calculated as follows:

$$L\left(\vartheta^B\right) = -\frac{1}{\nu}\left[\sum_{i=1}^{\nu}\sum_{\varsigma=1}^{C_B} \Bbbk\left\{y_B^{(i)} = \varsigma\right\} \log p\left(y_B^{(i)} = \varsigma | x^{(i)}; \vartheta^B\right)\right] \tag{10}$$

where ν denotes the number of training samples, and $\Bbbk$ is an indicator function that outputs 1 when the predicted result matches the true label and 0 otherwise. Next, the total loss $L\left(\vartheta^A, \vartheta^B\right)$ of the model is the weighted sum of the losses from the two tasks:

$$L\left(\vartheta^A, \vartheta^B\right) = \alpha L\left(\vartheta^A\right) + \beta L\left(\vartheta^B\right) \tag{11}$$

where α and β are hyperparameters used to balance the losses of the two tasks.

By minimizing this total loss function, the model can be simultaneously trained to optimize the performance of both classification tasks. Finally, the model parameters ϑ can be iteratively updated using the gradients of the total loss function:

$$\vartheta \leftarrow \vartheta - \eta \frac{\partial L\left(\vartheta^A, \vartheta^B\right)}{\partial \vartheta} \tag{12}$$

where η denotes the learning rate.

Repeating the above steps until reaching the specified number of iterations or the loss function converges to a sufficiently small value. The procedure for high-speed train fault diagnosis is clarified in Algorithm 1, presenting the pseudocode of the proposed JA-MFSN method. Overall, applying the JA-MFSN method to high-speed train fault diagnosis involves six main steps:

Algorithm 1 The proposed JA-MFSN method

```
Input data: $D = \left\{x^{(i)}, (y_A^{(i)}, y_B^{(i)})\right\}_{i=1}^{\nu}$
Output: $\left\{(\widehat{y}_A^{(i)}, \widehat{y}_B^{(i)})\right\}_{i=1}^{\nu}$

Initialize the JA-MFSN
for each epoch do
    for each batch do
        1. Extract general fault features from input samples through multi-scale CNN, followed by normalization using BN;
        2. Feed the extracted feature vectors $X$ into JAM, and attention coefficients $S$ are computed using (1);
        3. Obtain the sensitive fault features $\acute{X}$ according to the attention coefficients $S$ through (2);
        4. Combine attention and non-attention features using (3);
        5. Compute attention evaluation weights using input by (4);
        6. Compute the proportion of attention and non-attention features to avoid feature loss caused by redundant attention mechanisms through (5);
        7. Preserve the original features and obtain the final output of JAM through (6);
        8. Calculate and minimize the total loss function $L\left(\vartheta^A, \vartheta^B\right)$ through (7) to (11);
        9. Compute the partial derivatives of $L\left(\vartheta^A, \vartheta^B\right)$ and update the model parameters iteratively through (12).
    end for
    Note the present diagnostic results $\left\{(\widehat{y}_A^{(i)}, \widehat{y}_B^{(i)})\right\}_{i=1}^{\nu}$
end for
return The supreme diagnostic results $\left\{(\widehat{y}_A^{(i)}, \widehat{y}_B^{(i)})\right\}_{i=1}^{\nu}$
```

Step 1. Data Collection and Preprocessing: Fault signals from the high-speed train traction system, such as vibration, voltage, and current signals, are collected using sensors. The collected signals are then normalized and segmented into samples of specific lengths.

Step 2. JA-MFSN Initialization: Before training the model, initialize the weights and biases in the model with small initial values.

Step 3. JA-MFSN Training: Randomly shuffle the input fault samples and divide them into training and test sets. During the iterative training process, select a specified subset from the training set for model training and update the model parameters through backpropagation.

Step 4. JA-MFSN Iterative Optimization: Iteratively compute the total loss function $L\left(\vartheta^A, \vartheta^B\right)$ and simultaneously update the corresponding model parameters ϑ^A and ϑ^B until the model converges on both tasks.

Step 5. JA-MFSN Testing: Utilize the Softmax function of outputs A and B to calculate their probability distributions for the test samples of task A and task B, respectively. The class with the highest probability in each output is the predicted result for the corresponding task.

Step 6. JA-MFSN Application: The trained model can effectively integrate fault data from related tasks and accurately classify different but related tasks. The limited model parameters can also help mitigate potential overfitting, thereby enhancing the accuracy and efficiency of the multitask model in high-speed train fault diagnosis.

III. Experimental Results and Analysis

In this study, the proposed JA-MFSN fault diagnosis method is deployed and evaluated on the HIL simulation platform of the high-speed train traction control system, as illustrated in Fig. 2. The data utilized in the HIL platform is directly acquired through sensor signal sampling from actual high-speed train traction control systems. The simulation settings are configured to reflect real operating conditions, enabling the platform to closely replicate the data characteristics and fault patterns observed in real-world environments. This simulation approach is widely adopted in engineering validation and is recognized for its high reliability and credibility. Prior studies [26], [45] have demonstrated that the HIL simulation method is extensively employed in algorithm verification for practical industrial applications, such as high-speed trains, power electronics, and aerospace systems, consistently exhibiting strong alignment with real-world deployment outcomes. Additionally, to further demonstrate the effectiveness of JA-MFSN, this method is tested and compared with several other well-known multitask learning methods on the same simulation platform. The implementation of JA-MFSN is carried out using TensorFlow and Keras, and the training is performed on a high-performance server with an Xeon (R) Intel (R) CPU E5-2678v3@2.50GHz, 64 GB of main memory, and an NVIDIA Titan RTX GPU.

A. Dataset Description

The dataset used in this study originates from the HIL simulation platform for high-speed train traction control systems, jointly established by Central South University and Zhuzhou Electric Locomotive Research Institute, as depicted in Fig. 2. This simulation platform mainly comprises five components: the dSPACE real-time simulator (DRTS), power source, traction control unit (TCU), personal computer (PC), and signal conditioner. The collected data includes three predefined fault types: rotor bar break (RBB), inter-turn short circuit (ISC), and air gap eccentricity (AGE), and each type has three severity levels: mild, moderate, and severe. These fault signals are recorded through sensors monitoring the direct current-side voltage (DCV) at a sampling frequency of 2.5 kHz. Additionally, the dataset covers three distinct operating conditions of the high-speed train: 20 km/h, 160 km/h, and 280 km/h. For each operating condition, data are gathered under ten operational states, comprising nine faulty states and one normal state. To quantify the experimental scale, a sliding window segmentation strategy is applied to the recorded signals, with a window length of 1024 data points and a step size of 128 data points. This process yields a total of 5602 sample segments. Of these, 4500 samples are randomly allocated to the training set, 500 to the validation set, and 602 to the test set.

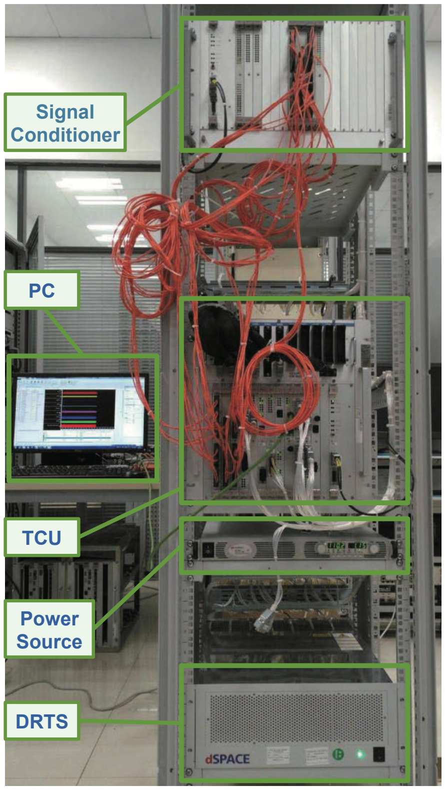

Fig. 2. Structure diagram of the HIL high-speed train traction control system simulation platform.

To provide a clear illustration of the traction system's operating status, Fig. 3 presents representative DCV signal waveforms for the ten operating states at a speed of 280 km/h. It can be observed that the DCV signal samples corresponding to different operating states exhibit distinct waveform characteristics. Normal signals display stable, regular patterns with small amplitude fluctuations and exhibit typical periodic behavior. For RBB fault signals, mild faults are characterized

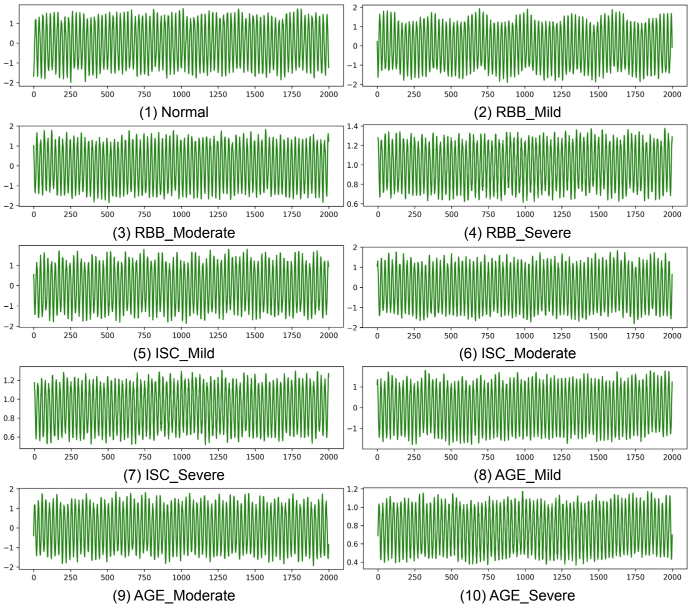
(1) Normal (2) RBB_Mild (3) RBB_Moderate (4) RBB_Severe (5) ISC_Mild (6) ISC_Moderate (7) ISC_Severe (8) AGE_Mild (9) AGE_Moderate (10) AGE_Severe

Fig. 3. Schematic diagram of the normalized DCV signals for ten fault states.

by noticeable periodic amplitude fluctuations, often manifested as localized amplitude reductions. Moderate RBB faults show more pronounced amplitude variations and stronger periodicity compared to the mild case. Severe RBB faults are distinguished by significantly increased amplitude oscillations, with more frequent and prominent periodic changes. In the case of ISC fault signals, mild faults generally maintain relatively stable waveforms, with occasional minor irregular vibrations or disturbances. Moderate ISC faults demonstrate increased signal volatility and more evident random disturbances. Severe ISC faults are characterized by frequent, sharp amplitude spikes and highly irregular waveform patterns. For AGE fault signals, mild faults display relatively regular oscillations with gradually emerging periodic fluctuations. Moderate AGE faults exhibit more distinct and regular periodic amplitude oscillations. Severe AGE faults are marked by pronounced periodic oscillations with larger amplitude swings, making them clearly distinguishable from other fault states.

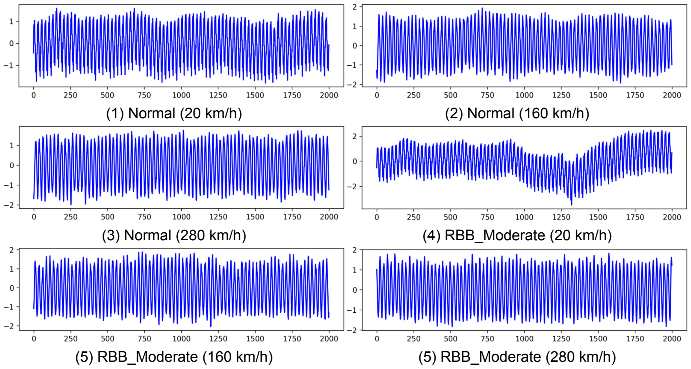
(1) Normal (20 km/h) (2) Normal (160 km/h) (3) Normal (280 km/h) (4) RBB_Moderate (20 km/h) (5) RBB_Moderate (160 km/h) (5) RBB_Moderate (280 km/h)

Fig. 4. Schematic diagram of the normalized DCV signals for three operating conditions.

To intuitively illustrate the effect of different operating conditions on the DCV signal, Fig. 4 displays representative signal waveforms for both the normal condition and the moderate RBB fault at three distinct operating speeds. For the normal condition, the signals across various speeds predominantly exhibit high-frequency oscillations with sinusoidal characteristics, demonstrating stable amplitude fluctuations and an absence of transient distortions or irregular spikes. As the operating speed increases, the fundamental frequency of the signal rises accordingly, reflecting the higher rotational speed of the traction motor. In contrast, under the moderate RBB fault condition, the waveform is marked by a distinct amplitude envelope depression in the central segment, manifesting as a cyclic "attenuation–recovery–attenuation" pattern. This characteristic is a typical indicator of rotor bar breakage, where the induced voltage periodically diminishes at certain rotor angular positions. As speed increases, these envelope fluctuations become denser; nevertheless, the fault-induced envelope depressions remain clearly discernible.

B. *Diagnosis Task Definition*

In practical fault diagnosis applications for high-speed train traction systems, accurately identifying the operating condition is essential for maintaining system safety and reliability. However, fault data acquired under real-world conditions are often limited, which poses substantial challenges for improving the performance of conventional single-task diagnostic models. To address this limitation, this study proposes a multitask learning-based fault diagnosis approach that effectively utilizes limited data resources by sharing features across related tasks. Specifically, the diagnostic framework in this work comprises two parallel tasks.

Task A. Fault Category Classification: The objective of this task is to accurately determine the operating state of the traction motor (normal, RBB, ISC, or AGE), as well as the severity level of any fault (mild, moderate, or severe), based on DCV signals collected from traction system sensors. This classification framework yields a total of ten possible states. Timely identification of both the specific fault type and its severity enables the implementation of appropriate maintenance actions, thereby preventing further escalation of faults and enhancing system reliability.

Task B. Operating Condition Classification: The objective of this task is to accurately determine the current speed state of the high-speed train's traction motor (20 km/h, 160 km/h, and 280 km/h) based on the same DCV signals. Identifying the operating condition provides valuable contextual information for condition-specific diagnostic decision-making, thereby significantly enhancing the accuracy and relevance of fault diagnosis in real-world applications.

By simultaneously addressing the two tasks described above, the proposed JA-MFSN method leverages the inherent correlations between tasks to achieve more accurate and robust fault diagnosis for traction systems. Specifically, a task-shared network is utilized to extract general fault-related features from the operating signals, while task-specific networks are designed to capture features tailored to each diagnostic objective. This architectural design not only improves diagnostic accuracy but also effectively controls model complexity, thereby reducing the risk of overfitting under limited data conditions.

C. Ablation Analysis

1) Ablation Study of JAM: To comprehensively demonstrate the impact of the proposed JAM within the JA-MFSN method, we conduct comparative experiments involving networks configured with varying numbers of JAMs. Specifically, we evaluate the deployment of 0 to 6 JAMs within each of the two task-specific networks in JA-MFSN, aiming to identify the most advantageous JAM deployment strategy. To rigorously assess the robustness of our model with respect to initialization and training variability, each configuration is independently trained ten times using the same training dataset and identical sample numbers in all repetitions. The best, average, and worst accuracies from these ten runs are presented in Table III. The optimal results in Table III are indicated in bold, while suboptimal results are underlined for clarity.

TABLE III
PERFORMANCE OF JA-MFSN WITH DIFFERENT NUMBER OF JAMS.

JAM	Fault Category			Operating Condition			Parameters
	Best	Average	Worst	Best	Average	Worst	
0	98.90%	95.55%	88.05%	96.51%	95.08%	92.58%	60, 557
1	**99.90%**	99.50%	97.81%	96.56%	96.14%	95.12%	66, 141
2	99.75%	**99.52%**	98.66%	99.60%	98.13%	94.52%	71, 725
3	99.85%	99.37%	98.01%	**100%**	99.68%	98.41%	77, 309
4	99.50%	99.24%	97.96%	**100%**	99.70%	97.31%	82, 893
5	**99.90%**	99.51%	**99.20%**	**100%**	**99.81%**	**99.20%**	88, 477
6	99.75%	99.06%	95.92%	**100%**	99.57%	98.06%	94, 061

As shown in Table III, integrating the JAMs consistently improves diagnostic accuracy compared with the baseline model without JAM. Notably, deploying five JAMs within each task-specific network achieves the best overall performance. This improvement can be attributed to JAM's ability to effectively balance model complexity with the extraction of task-sensitive features. Specifically, when fewer than five JAMs are adopted, the model may lack sufficient capacity to capture complex fault characteristics, thereby limiting diagnostic accuracy. In contrast, increasing the number of JAMs beyond five introduces redundant parameters, which slightly raises the risk of overfitting and reduces the stability of diagnostic accuracy across multiple runs, as evidenced by the marginal performance drop observed with six JAMs. Therefore, configuring the network with five JAMs provides an optimal trade-off, enabling the model to capture comprehensive fault representations while maintaining computational efficiency and mitigating overfitting. These results further confirm that excessive lightweighting of the network does not lead to improved diagnostic outcomes. The key lies in integrating the JAMs, which not only preserves model compactness but also substantially enhances multitask diagnostic accuracy, thereby validating the effectiveness and practical advantages of the proposed method.

2) Ablation Study on Single-Task Diagnosis: To further validate the motivation for adopting multitask learning, we conducted ablation experiments by applying each method to a single-task fault diagnosis scenarios. In this experiment, each model retains only the task-shared network stream and a single task-specific network stream, focusing on one task at a time without any shared representation across tasks. The hyperparameters, training epoch settings, and input data are kept consistent with those in the multitask experiments to ensure fairness. All experiments are independently repeated ten times. Table IV reports the best, average, and worst accuracies, as well as the number of trainable parameters to reflect computational resource consumption.

TABLE IV
EFFECTIVENESS ANALYSIS OF DIFFERENT METHODS IN SINGLE-TASK DIAGNOSIS.

Method	Fault Category			Parameters
	Best	Average	Worst	
M2FN	90.47%	85.60%	83.87%	287, 882
MACNN	84.07%	83.20%	81.33%	141, 002
MSFMTP	54.15%	52.09%	50.52%	1, 844, 458
MT1DCNN	96.39%	94.40%	93.07%	105, 818
MTAGN	98.84%	95.53%	90.60%	189, 570
JA-MFSN	**99.62%**	**98.09%**	**96.85%**	**72, 418**
Method	Operating Condition			Parameters
	Best	Average	Worst	
M2FN	80.56%	78.49%	77.92%	287, 427
MACNN	93.23%	90.12%	88.07%	112, 323
MSFMTP	77.21%	56.80%	34.26%	1, 844, 458
MT1DCNN	89.21%	86.42%	82.29%	98, 643
MTAGN	91.82%	88.13%	86.12%	182, 395
JA-MFSN	**99.54%**	**98.79%**	**97.51%**	**72, 299**

As shown in Table IV, JA-MFSN can consistently achieve the highest diagnostic accuracy across both fault diagnosis scenarios, demonstrating superior performance. Compared with the multitask experiments, most methods exhibit decreased classification accuracy for both fault categories and operating conditions. Moreover, the total number of parameters in the multitask setting is smaller than the sum of those in the two single-task models, indicating improved computational efficiency. Overall, these results clearly demonstrate that the multitask paradigm not only provides higher accuracy but also reduces redundant parameters through feature sharing.

D. Influence of Weight Values of Loss Function

In this subsection, we conduct experiments to investigate the impact of the hyperparameters α and β in the loss function on the diagnostic performance of JA-MFSN. Specifically, α and β are systematically varied across five discrete values: 0.2, 0.4, 0.6, 0.8, and 1.0, resulting in a total of 25 unique experimental configurations. Each experiment is independently repeated ten times, and the average accuracies for both fault category classification and operating condition classification tasks are reported in Table V. Analysis of the results in Table V indicates that variations in α and β do affect diagnostic outcomes; however, the overall impact is relatively modest. For fault category classification, the diagnostic accuracy ranges from 96.62% to 99.79%, with a narrow margin of 3.17%. Similarly, the accuracy for operating condition classification exhibits minimal variability, ranging from 97.16% to 99.94%, corresponding to a small difference of 2.78%.

The robustness and minimal sensitivity of JA-MFSN to variations in these hyperparameters can be primarily attributed

TABLE V
PERFORMANCE OF JA-MFSN WITH DIFFERENT HYPERPARAMETER SETTINGS.

Task	β / α	0.2	0.4	0.6	0.8	1.0
Fault Category	0.2	99.40%	99.51%	98.54%	96.62%	97.28%
	0.4	99.56%	99.20%	99.27%	99.37%	99.38%
	0.6	99.57%	99.73%	99.54%	99.45%	99.13%
	0.8	**99.79%**	99.68%	99.39%	99.49%	99.46%
	1.0	99.66%	99.39%	99.50%	99.58%	99.51%
Operating Condition	0.2	99.78%	**99.94%**	99.37%	99.69%	99.26%
	0.4	99.62%	99.50%	99.74%	99.91%	99.92%
	0.6	98.79%	99.61%	99.56%	99.05%	99.40%
	0.8	97.16%	99.90%	99.85%	99.89%	99.78%
	1.0	97.71%	99.56%	99.93%	99.89%	99.81%

TABLE VI
PERFORMANCE OF DIFFERENT MULTITASK MODELS FOR HIGH-SPEED TRAIN FAULT DIAGNOSIS.

Method	Fault Category			Operating Condition			Parameters
	Best	Average	Worst	Best	Average	Worst	
M2FN	92.88%	91.99%	90.24%	85.71%	82.57%	80.98%	566,925
MACNN	85.11%	84.00%	82.87%	97.66%	95.69%	90.29%	306,573
MSFMTP	49.85%	47.95%	46.66%	84.91%	51.33%	47.46%	2,512,749
MT1DCNN	98.36%	97.82%	95.77%	92.63%	90.70%	88.79%	188,029
MTAGN	99.20%	96.20%	92.38%	93.43%	92.25%	90.04%	342,941
JA-MFSN	**99.90%**	**99.51%**	**99.20%**	**100%**	**99.81%**	**99.20%**	**88,477**

to its inherent capability to dynamically balance attention and non-attention mechanisms through the integration of multiple JAMs. Specifically, JAM ensures the effective extraction and preservation of critical fault features, regardless of minor fluctuations in the loss function weights. Notably, configuring the hyperparameters to $\alpha = 0.8$ and $\beta = 0.2$ yields the highest accuracy for fault category classification, while $\alpha = 0.2$ and $\beta = 0.4$ achieve optimal accuracy for operating condition classification, indicating subtle preferences for task-specific weighting strategies. For practical applications, and to ensure consistency and fairness in method comparisons, both hyperparameters are set to $\alpha = 1.0$ and $\beta = 1.0$ in subsequent comparative experiments. This configuration provides balanced prioritization between the two classification tasks and establishes a fair evaluation baseline for assessing the performance of JA-MFSN against existing approaches.

E. Performance Comparison With Existing Methods

To verify the superior performance of the JA-MFSN method in high-speed train fault diagnosis, several recently published studies have been used in comparison experiments. These studies include the multitask multisensor fusion network (M2FN) [6], multi-tasking atrous CNN (MACNN) [35], multi-scale feature fusion and multitask parallel learning (MSFMTP) [54], multitask one-dimensional CNN (MT1DCNN) [22], and multitask attention guided network (MTAGN) [38]. These multitask models are applied to simultaneously handle both the fault category classification task and the operating condition classification task. It is noteworthy that to ensure the fairness of the experiments, the hyperparameters α and β in the loss functions of all methods are both set to 0.5. Furthermore, based on the results of ablation experiments, we configure 5 JAMs in the task-specific network of the proposed JA-MFSN to achieve supreme performance. To prevent performance degradation in larger models caused by insufficient training, all methods employed the same optimizer, batch size, early stopping technique, and maximum number of epochs. These controls prevent under-training of larger models and ensure capacity–performance comparisons are meaningful. To account for potential uncertainty due to neural network randomness, all comparative experiments are independently repeated ten times. The best, average, and worst accuracies are used to provide a comprehensive assessment of the fault diagnosis performance of these methods, and the number of model parameters is listed to highlight the advantages of the proposed JA-MFSN method in terms of model lightweight and overfitting avoidance. The experimental results are summarized in Table VI, where the optimal outcomes are highlighted in bold and the sub-optimal results are underlined for ease of interpretation.

The results presented in Table VI clearly demonstrate the superior multitask classification performance of the proposed JA-MFSN method in high-speed train fault diagnosis compared to several other efficient methods. For the fault category classification task, the JA-MFSN method demonstrates remarkable diagnostic accuracy over ten repeated experiments, outperforming the competing methods by margins ranging from 0.7% to 50.05% in the best accuracy, 1.69% to 51.56% in the average accuracy, and 3.43% to 52.54% in the worst accuracy. Similarly, for the operating condition classification task, the JA-MFSN method achieves substantial performance gains, with best accuracy increasing by 2.34% to 15.09%, average accuracy improving by 4.12% to 17.24%, and worst accuracy rising by 8.91% to 51.75%. The experimental results indicate that the proposed JA-MFSN method can effectively handle different classification tasks and outperform the other five eminent multitask fault diagnosis methods in both fault category classification and operating condition classification tasks.

Regarding model parameters, although other multitask learning–based methods can achieve satisfactory classification performance, their large number of model parameters often increases the risk of overfitting, thereby reducing overall effectiveness. This issue is particularly evident in the MSFMTP method, where an excessive number of redundant parameters severely compromises classification accuracy. In addition, some efficient models, such as MACNN, MT1DCNN, and MTAGN, despite having heterogeneous parameter sizes, have each achieved promising results in different tasks. These observations indicate that merely reducing (or increasing) the number of model parameters cannot effectively improve diagnostic accuracy. In other words, the superiority of the proposed approach does not stem solely from parameter reduction, but rather from its architecture, which integrates a task-sharing network with JAMs-based task-specific networks. This design not only enables effective multitask feature sharing but also preserves strong feature extraction capability.

In summary, by achieving an optimal balance between model complexity and performance, JA-MFSN addresses the inherent trade-off problem in multitask learning models and demonstrates outstanding efficiency in real-world high-speed train fault diagnosis. The proposed method ensures lightweight

Citation information: DOI: 10.1109/TIM.2025.3627341, IEEE Transactions on Instrumentation and Measurement

design while maintaining strong feature extraction capability, which can be attributed to several advantages. First, the task-shared network in JA-MFSN can effectively extract common features across different tasks, thereby improving the efficiency of feature learning. Second, JAMs can adaptively activate or suppress attention, this adaptive gating mechanism enhances the extraction of critical features while preventing feature loss. Moreover, the task-specific networks based on JAMs preserve the unique characteristics of each task and suppress cross-task interference, as evidenced by the ablation studies. Finally, the overall framework of JA-MFSN remains lightweight, where reduced redundant computation and adaptive suppression of unnecessary attention activations yield superior diagnostic accuracy and computational efficiency, while effectively mitigating the risk of overfitting.

F. *Efficiency Comparison of Edge Deployment*

```
jtop 15W|CPU 7.7%|GPU 0.0%
Model: NVIDIA Orin NX Developer Kit - Jetpack 5.1.1 [L4T 35.3.1]
 Temperatures: GPU  45.94C
 iGPU ga10b   0% - gov: nvhost_podgov                 GPU Shared RAM 93.3M/7.2GB
                                            100%                                     7.2G
                                             91%                                     6.6G
                                             83%                                     6.0G
                                             75%                                     5.4G
                                             66%                                     4.8G
                                             58%                                     4.2G
                                             50%                                     3.6G
                                             41%                                     3.0G
                                             33%                                     2.4G
                                             25%                                     1.8G
                                             16%                                     1.2G
                                              8%                                     0.6G
      -8s      -6s      -4s      -2s      0 time    -8s      -6s      -4s      -2s   0 time
  3D scaling:[Active]      Railgate: Disable       Power ctrl: auto       TPC PG: OFF
  Frq [306MHz::::::::::::::::::::::::612MHz] 612MHz GPC0 — RUNNING ——◆ 612MHz
  PID    USER     GPU  TYPE  PRI  S  CPU%  MEM     [GPU MEM]  Command
  2569   zackzou  I    G     20   S  2.2   66.9M   58.2M      gnome-shell
  1607   zackzou  I    G     20   S  3.7   11.8M   35.1M      Xorg
  1024   root     I    G     20   S  0.0   2.6M    0k         nvargus-daemon

Show Applications
 1ALL 2GPU 3CPU 4MEM 5ENG 6CTRL 7INFO Quit                                     (c) 2024, RB
```

Fig. 5. Schematic diagram of Nvidia Jetson Orin NX 8G Super device.

TABLE VII
EFFICIENCY COMPARISON OF DIFFERENT METHODS ON EDGE DEVICE.

Method	Power	Inference Latency	Memory Consumption CPU	GPU
M2FN	10W	15.92 ms	1.22 MB	237.97 MB
MACNN		15.16 ms	0 MB	283.09 MB
MSFMTP		21.79 ms	0 MB	375.97 MB
MT1DCNN		12.75 ms	0 MB	185.48 MB
MTAGN		16.75 ms	0 MB	232.80 MB
JA-MFSN		10.84 ms	0 MB	148.10 MB
M2FN	20W	12.94 ms	0 MB	237.97 MB
MACNN		13.78 ms	0 MB	283.09 MB
MSFMTP		18.72 ms	0.72 MB	375.97 MB
MT1DCNN		10.22 ms	0 MB	185.48 MB
MTAGN		13.32 ms	0 MB	232.80 MB
JA-MFSN		9.19 ms	0 MB	148.10 MB
M2FN	40W (Max)	10.27 ms	0 MB	237.97 MB
MACNN		10.75 ms	0 MB	283.09 MB
MSFMTP		15.08 ms	1.89 MB	375.97 MB
MT1DCNN		8.59 ms	1.70 MB	185.48 MB
MTAGN		9.37 ms	0 MB	232.80 MB
JA-MFSN		7.02 ms	0 MB	148.10 MB

To further validate the practical applicability and efficiency of the proposed JA-MFSN method in real-world deployment scenarios, we implement and evaluate the JA-MFSN model on an Nvidia Jetson Orin NX 8G Super edge device, as exhibited in Fig. 5. The comparative analysis of computational efficiency with several advanced multitask fault diagnosis methods under different power constraints (10W, 20W, and 40W (Max)) is summarized in Table VII. Specifically, under 10W, 20W, and 40W conditions, JA-MFSN achieves the lowest inference latencies of 10.84 ms, 9.19 ms, and 7.02 ms, respectively. Compared with the second-best-performing method (MT1DCNN), JA-MFSN reduces latency by approximately 15.0%, 10.1%, and 18.3% at the corresponding power levels. These results demonstrate the superior computational efficiency and real-time performance of JA-MFSN, which are essential for practical deployment on edge devices. Overall, the experimental findings confirm that JA-MFSN not only achieves excellent diagnostic performance but also excels in computational efficiency when deployed on resource-constrained edge hardware. Its reduced latency and memory requirements further demonstrate its feasibility for real-time fault diagnosis in high-speed train traction systems.

IV. Conclusion

In pursuit of parallel processing of different classification tasks and improved classification accuracy in high-speed train fault diagnosis, this study proposes a novel JA-MFSN method designed to simultaneously accomplish high-accuracy fault category classification and operating condition classification tasks. Specifically, the proposed method can effectively reduce the trainable parameters in multitask models, thereby mitigating the overfitting challenge caused by parameter redundancy and enhancing model performance. Additionally, the proposed JAM can efficiently extract sensitive fault features with limited parameters, avoiding feature loss issue. Ablation experiments demonstrate the effectiveness of JAM, showing that this module can retain useful features for specific tasks while avoiding feature loss due to redundant attention mechanisms. The proposed method is tested on the HIL high-speed train traction control system simulation platform, with the experimental results clearly indicating the proposed JA-MFSN method's significant advantages over five efficient multitask learning-based methods. In practical industrial deployments, deep learning-based approaches commonly encounter challenges such as limited fault data, constrained computational resources, and real-time processing requirements, which may impact their generalizability. Future research will focus on addressing these challenges by exploring data augmentation techniques and lightweight model architectures tailored for real-world applications. In addition, the proposed multitask learning strategy will be extended to a broader range of mechanical fault diagnosis scenarios, including diagnostic strategies for power converters [1], train bogies [37], and autonomous underwater vehicles [52].

References

[1] J. Cai, Y. Wang, W. Ding, A. David Cheok, Y. Yan, and X. Zhang, "Real-Time Power Switch Short-Circuit Fault Diagnosis for the Power

Citation information: DOI: 10.1109/TIM.2025.3627341, IEEE Transactions on Instrumentation and Measurement

Converter in SRM Drives," IEEE Transactions on Instrumentation and Measurement, vol. 74, pp. 1-14, 2025.

[2] Y. Cai, X. Yang, Y. Yang, and Q. Liu, "Leader-Following Privacy-Preserving Consensus Control of Nonlinear Multi-Agent Systems: A State Decomposition Approach," International Journal of Systems Science, vol. 56, no. 10, pp. 2284-2295, 2025.

[3] F. Chen, S. Li, J. Han, F. Ren, and Z. Yang, "Review of Lightweight Deep Convolutional Neural Networks," Archives of Computational Methods in Engineering, vol. 31, no. 4, pp. 1915-1937, 2023.

[4] H. Chen, R. Wu, C. Tao, W. Xu, H. Liu, C. Xu, and M. Jian, "Multi-Scale Class Attention Network for Diabetes Retinopathy Grading," International Journal of Network Dynamics and Intelligence, vol. 3, no. 2, 2024.

[5] X. Chen, R. Yang, Y. Xue, M. Huang, R. Ferrero, and Z. Wang, "Deep Transfer Learning for Bearing Fault Diagnosis: A Systematic Review Since 2016," IEEE Transactions on Instrumentation and Measurement, vol. 72, 2023.

[6] J. Cui, P. Xie, X. Wang, J. Wang, Q. He, and G. Jiang, "M2FN: An End-to-End Multi-Task and Multi-Sensor Fusion Network for Intelligent Fault Diagnosis," Measurement, vol. 204, 2022.

[7] W. Cui, P. Li, and R. Chi, "Data-Driven Predictive ILC for Nonlinear Nonaffine Systems," International Journal of Systems Science, vol. 55, no. 9, pp. 1868-1881, 2024.

[8] F. Deng, Y. Ming, and B. Lyu, "CCE-Net: Causal Convolution Embedding Network for Streaming Automatic Speech Recognition," International Journal of Network Dynamics and Intelligence, 2024.

[9] T. Gao, H. Pan, Z. Wang, and H. Gao, "A CRF-Based Framework for Tracklet Inactivation in Online Multi-Object Tracking," IEEE Transactions on Multimedia, vol. 24, pp. 995-1007, 2022.

[10] G. Geetha and P. Geethanjali, "An Efficient Method for Bearing Fault Diagnosis," Systems Science & Control Engineering, vol. 12, no. 1, 2024.

[11] G. Geetha and P. Geethanjali, "Computational Intelligence to Detect Bearing Faults Using Optimal Features From Motor Current Signals," Systems Science & Control Engineering, vol. 12, no. 1, 2024.

[12] C. Guan, R. Shang, F. Yang, A.-x. Shao, and S. Zhang, "A Priori Information-Guided Generative Adversarial Network for Data Augmentation: Application to Pipeline Fault Diagnosis," Systems Science & Control Engineering, vol. 12, no. 1, 2024.

[13] L. Gong, C. Pang, G. Wang, and N. Shi, "Lightweight Bearing Fault Diagnosis Method Based on Improved Residual Network," Electronics, vol. 13, no. 18, 2024.

[14] C. Han and Z. Xu, "Pattern-Moving Based Data-Driven Control for Multi-Input Continuous-Time Non-Newtonian Mechanical Systems," International Journal of Systems Science, vol. 56, no. 10, pp. 2406-2430, 2025.

[15] S. He, W. K. Ao, and Y.-Q. Ni, "A Unified Label Noise-Tolerant Framework of Deep Learning-Based Fault Diagnosis via a Bounded Neural Network," IEEE Transactions on Instrumentation and Measurement, vol. 73, pp. 1-15, 2024.

[16] X. He, Z. Wang, C. Gao, and D. Zhou, "Consensus Control for Multiagent Systems Under Asymmetric Actuator Saturations With Applications to Mobile Train Lifting Jack Systems," IEEE Transactions on Industrial Informatics, vol. 19, no. 10, pp. 10224-10232, 2023.

[17] J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans, "Axial Attention in Multidimensional Transformers," arXiv preprint, arXiv:1912.12180, 2019.

[18] S. Hu, J. Lu, and S. Zhou, "Learning Regression Distribution: Information Diffusion from Template to Search for Visual Object Tracking," International Journal of Network Dynamics and Intelligence, 2024.

[19] X. Jiang, X. Li, Q. Wang, Q. Song, J. Liu, and Z. Zhu, "Multi-Sensor Data Fusion-Enabled Semi-Supervised Optimal Temperature-Guided PCL Framework For Machinery Fault Diagnosis," Information Fusion, vol. 101, 2024.

[20] P. Kong, M. Wang, H. Yan, Z. Li, and Y. Lv, "Data-Driven Control for Linear Discrete-Time Systems With Time-Varying Delays," International Journal of Systems Science, 2025.

[21] Y. Li, S. Wang, J. Xie, T. Wang, J. Yang, T. Pan, and B. Yang, "A Lightweight Dual-Compression Fault Diagnosis Framework for High-Speed Train Bogie Bearing," IEEE Transactions on Instrumentation and Measurement, vol. 73, pp. 1-14, 2024.

[22] Z. Liu, H. Wang, J. Liu, Y. Qin, and D. Peng, "Multitask Learning Based on Lightweight 1DCNN for Fault Diagnosis of Wheelset Bearings," IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-11, 2021.

[23] D. Lyu, C. Li, Z. Han, Z. Song, Z. Yang, Y. Ma, and J. Hong, "Position-Agnostic Aeroengine Intershaft Bearing Fault Diagnosis via Condition-Guided Multitask Learning," IEEE Transactions on Instrumentation and Measurement, vol. 74, pp. 1-17, 2025.

[24] G. Ma, Z. Wang, W. Liu, J. Fang, Y. Zhang, H. Ding, and Y. Yuan, "Estimating the State of Health for Lithium-ion Batteries: A Particle Swarm Optimization-Assisted Deep Domain Adaptation Approach," IEEE/CAA Journal of Automatica Sinica, vol. 10, no. 7, pp. 1530-1543, 2023.

[25] L. Ma, P. Yang, and K. Peng, "Multitask Learning Based Collaborative Modeling of Heterogeneous Data for Compound Fault Diagnosis in Manufacturing Processes," IEEE Transactions on Industrial Informatics, vol. 20, no. 12, pp. 14174-14183, 2024.

[26] F. Mihalič, M. Truntič, and A. Hren, "Hardware-in-the-Loop Simulations: A Historical Overview of Engineering Challenges," Electronics, vol. 11, no. 15, 2022.

[27] G. Niu, E. Liu, X. Wang, P. Ziehl, and B. Zhang, "Enhanced Discriminate Feature Learning Deep Residual CNN for Multitask Bearing Fault Diagnosis With Information Fusion," IEEE Transactions on Industrial Informatics, vol. 19, no. 1, pp. 762-770, 2023.

[28] H. Shao, Y. Tan, J. Li, H. Gao, H. Yin, and H. Gao, "A Non-Local Adaptive Network for Cross-Domain Intelligent Fault Diagnosis Leveraging Multi-Source IOT Data," Signal, Image and Video Processing, vol. 19, no. 4, 2025.

[29] Y. Shen, Z. Wang, H. Dong, H. Liu, and Y. Chen, "Set-Membership State Estimation for Multirate Nonlinear Complex Networks Under FlexRay Protocols: A Neural-Network-Based Approach," IEEE Transactions on Neural Networks and Learning Systems, vol. 36, no. 3, pp. 4922-4933, 2025.

[30] Q. Song, X. Jiang, J. Liu, J. Shi, and Z. Zhu, "Contrast-Assisted Domain-Specificity-Removal Network for Semi-Supervised Generalization Fault Diagnosis," IEEE Transactions on Neural Networks and Learning Systems, vol. 36, no. 3, pp. 5403-5416, 2025.

[31] C. Wang, Z. Wang, and H. Dong, "A Novel Prototype-Assisted Contrastive Adversarial Network for Weak-Shot Learning With Applications: Handling Weakly Labeled Data," IEEE/ASME Transactions on Mechatronics, vol. 29, no. 1, pp. 533-543, 2024.

[32] D. Wang, Y. Zhang, H. Zhang, Y. Zhuang, S. Gao, and Y. Li, "Bearing Fault Diagnosis Method Based on Multisensor Hybrid Feature Fusion," IEEE Transactions on Instrumentation and Measurement, vol. 74, pp. 1-11, 2025.

[33] H. Wang, Z. Liu, D. Peng, M. Yang, and Y. Qin, "Feature-Level Attention-Guided Multitask CNN for Fault Diagnosis and Working Conditions Identification of Rolling Bearing," IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 9, pp. 4757-4769, 2022.

[34] Y. Wang, C. Wen, and X. Wu, "Fault Detection and Isolation of Floating Wind Turbine Pitch System Based on Kalman Filter and Multi-Attention 1DCNN," Systems Science & Control Engineering, vol. 12, no. 1, 2024.

[35] Z. Wang, Y. Yin, and R. Yin, "Multi-Tasking Atrous Convolutional Neural Network for Machinery Fault Identification," The International Journal of Advanced Manufacturing Technology, vol. 124, no. 11-12, pp. 4183-4191, 2022.

[36] Q. Wei, X. Tian, L. Cui, F. Zheng, and L. Liu, "WSAFormer-DFFN: A Model for Rotating Machinery Fault Diagnosis Using 1D Window-Based Multi-Head Self-Attention and Deep Feature Fusion Network," Engineering Applications of Artificial Intelligence, vol. 124, 2023.

[37] J. Xie, S. Wang, Y. Li, T. Wang, J. Yang, and T. Pan, "A Simulated-to-Real Transfer Fault Diagnosis Method Based on Prototype Clustering Subdomain Adversarial Adaptation Network for HST Bogie Bearing," IEEE Transactions on Instrumentation and Measurement, vol. 73, pp. 1-13, 2024.

[38] Z. Xie, J. Chen, Y. Feng, K. Zhang, and Z. Zhou, "End to End Multi-Task Learning with Attention for Multi-Objective Fault Diagnosis Under Small Sample," Journal of Manufacturing Systems, vol. 62, pp. 301-316, 2022.

[39] Y. Xue, R. Yang, X. Chen, Z. Tian, and Z. Wang, "A Novel Local Binary Temporal Convolutional Neural Network for Bearing Fault Diagnosis," IEEE Transactions on Instrumentation and Measurement, vol. 72, 2023.

[40] Y. Xue, R. Yang, X. Chen, W. Liu, Z. Wang, and X. Liu, "A Review on Transferability Estimation in Deep Transfer Learning," IEEE Transactions on Artificial Intelligence, vol. 5, no. 12, pp. 5894-5914, 2024.

[41] Y. Xue, R. Yang, X. Chen, B. Song, and Z. Wang, "Separable Convolutional Network-Based Fault Diagnosis for High-Speed Train:

Citation information: DOI: 10.1109/TIM.2025.3627341, IEEE Transactions on Instrumentation and Measurement

A Gossip Strategy-Based Optimization Approach," IEEE Transactions on Industrial Informatics, vol. 21, no. 1, pp. 307-316, 2024.

[42] X. Yan, C. Wang, and Y. Jin, "Federated Bimodal Graph Neural Networks for Text-Image Retrieval," International Journal of Network Dynamics and Intelligence, vol. 4, no. 2, 2025.

[43] A. Yang, H. Wang, Q. Sun, and M. Fei, "Moving Horizon Estimation Based on Distributionally Robust Optimisation," International Journal of Systems Science, vol. 55, no. 7, pp. 1363-1376, 2024.

[44] M. Ye and J. Zhang, "Mobip: A Lightweight Model for Driving Perception Using MobileNet," Frontiers in Neurorobotics, vol. 17, p. 1291875, 2023.

[45] L. Yin, C. Luo, L. Liu, J. Cui, Z. Liu, and G. Sun, "A Hardware-in-the-Loop Simulation Platform for a High-Speed Maglev Positioning and Speed Measurement System," Technologies, vol. 13, no. 3, 2025.

[46] J. You, R. Yang, Y. Zhan, B. Song, Y. Zhang, and Z. Wang, "BR-MTFL: A Novel Byzantine Resilience-Enhanced Multitask Federated Learning Framework for High-Speed Train Fault Diagnosis," IEEE Transactions on Instrumentation and Measurement, vol. 74, pp. 1-13, 2025.

[47] N. Zeng, H. Li, and Y. Peng, "A New Deep Belief Network-based Multi-Task Learning for Diagnosis of Alzheimer's Disease," Neural Computing and Applications, vol. 35, no. 16, pp. 11599-11610, 2021.

[48] F. Zhan, L. Hu, W. Huang, Y. Dong, H. He, and G. Wu, "Category Knowledge-Guided Few-Shot Bearing Fault Diagnosis," Engineering Applications of Artificial Intelligence, vol. 139, 2025.

[49] Y. Zhang, N. Qin, D. Huang, A. Yang, X. Jia, and J. Du, "Generalized Zero-Shot Approach Leveraging Attribute Space for High-Speed Train Bogie," IEEE Transactions on Instrumentation and Measurement, vol. 73, pp. 1-12, 2024.

[50] Y. Zhang, L. Qiao, and M. Zhao, "Fault Diagnosis for Wind Turbine Generators Using Normal Behavior Model Based on Multi-Task Learning," IEEE Transactions on Automation Science and Engineering, pp. 1-13, 2024.

[51] Y. Zhang, X. Zhang, D. Miao, and H. Yu, "Real-Time Semantic Segmentation of Road Scenes via Hybrid Dilated Grouping Network," International Journal of Network Dynamics and Intelligence, vol. 4, no. 1, 2025.

[52] Z. Zhang, C. Wei, S. Xie, W. Zhang, and L. Wen, "A New Multisensor Feature Fusion KAN Network for Autonomous Underwater Vehicle Fault Diagnosis," IEEE Transactions on Instrumentation and Measurement, vol. 74, pp. 1-11, 2025.

[53] K. Zhou, N. Lu, B. Jiang, and Z. Ye, "FEV-Swin: Multi-Source Heterogeneous Information Fusion Under a Variant Swin Transformer Framework for Intelligent Cross-Domain Fault Diagnosis," Knowledge-Based Systems, vol. 310, 2025.

[54] L. Zhou, H. Wang, and S. Xu, "Aero-Engine Prognosis Strategy based on Multi-Scale Feature Fusion and Multi-Task Parallel Learning," Reliability Engineering & System Safety, vol. 234, 2023.

[55] D. Zhu and N. Cui, "Design of Intelligent Control for Flexible Linear Double Inverted Pendulum based on Particle Swarm Optimization Algorithm," Systems Science & Control Engineering, vol. 12, no. 1, 2024.