

Article

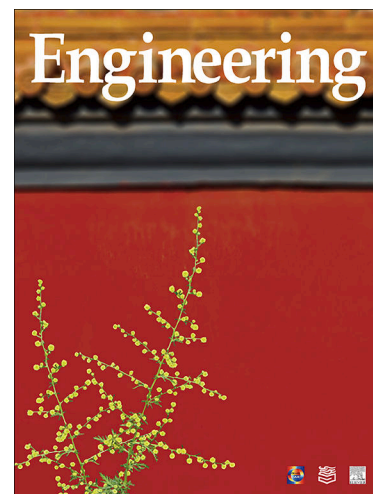
Cloud-Edge-Terminal Collaborative AI Agent for Compound Fault Detection and Diagnosis

Yiming He, Xiaoxi Hu, Weiming Shen

PII: S2095-8099(26)00174-8
DOI: <https://doi.org/10.1016/j.eng.2026.02.029>
Reference: ENG 2299

To appear in: *Engineering*

Received Date: 7 August 2025
Revised Date: 25 February 2026
Accepted Date: 28 February 2026



Please cite this article as: Y. He, X. Hu, W. Shen, Cloud-Edge-Terminal Collaborative AI Agent for Compound Fault Detection and Diagnosis, *Engineering* (2026), doi: <https://doi.org/10.1016/j.eng.2026.02.029>

This is a PDF of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability. This version will undergo additional copyediting, typesetting and review before it is published in its final form. As such, this version is no longer the Accepted Manuscript, but it is not yet the definitive Version of Record; we are providing this early version to give early visibility of the article. Please note that Elsevier's sharing policy for the Published Journal Article applies to this version, see: <https://www.elsevier.com/about/policies-and-standards/sharing#4-published-journal-article>. Please also note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Research

Intelligent Manufacturing—Article

Cloud-Edge-Terminal Collaborative AI Agent for Compound Fault Detection and Diagnosis

Yiming He ^a, Xiaoxi Hu ^b, Weiming Shen ^{a,*}^a National Center of Technology Innovation for Intelligent Design and Numerical Control, Huazhong University of Science and Technology, Wuhan 430074, China^b State Key Laboratory of Intelligent Green Vehicle and Mobility, Tsinghua University, Beijing 100084, China

* Corresponding authors.

E-mail addresses: wshen@ieee.org (W. Shen)

Abstract: With the continuous increase in the complexity and scale of intelligent systems, the demand for mechanical equipment maintenance for dynamic cross-domain versatility and human-computer collaborative intelligence has increased. The traditional static fault detection and diagnosis (FDD) paradigm, which is tailored to single scenarios and preset processes, has begun to exhibit inadequacy, inefficiency, and even unavailability. This paper proposes a cloud-edge-terminal collaborative artificial intelligence (CET-CAI) agent for system-level compound FDD (CFDD). The CET-CAI agent integrates the interactive and cognitive intelligence of a large language model with the deterministic reasoning capability of traditional fault diagnosis models. It can provide flexible dynamic task adaptability and efficient human-computer interaction with low communication costs. A novel distributed collaborative Transformer is designed and embedded in the edge system, which is specifically designed for multi-subsystem to reduce cross-subsystem signal interference and improve the generalization of unseen compound fault combinations. Experimental results on a multicomponent mechanical system dataset demonstrate that the proposed CET-CAI agent not only expands the dynamic task applicability of traditional diagnostic models, but also improves the robustness and readability of the diagnostic results.

Key words: Cloud-edge-terminal collaborative; Large language model; AI agent; Compound fault detection and diagnosis; Distributed collaborative transformer

1. Introduction

Fault detection and diagnosis (FDD) are essential tasks in intelligent manufacturing, which are crucial for ensuring system safety and reliability [1]. With the continuous increase in the scale and complexity of mechanical systems, compound FDD (CFDD) involving multiple subsystems has become inevitable. Traditional manual inspection methods are ineffective because of the complex signal transmission path and the coupling of multiple subsystem mechanisms. The feasibility of mainstream data-driven intelligent solutions has been demonstrated [2–4]. Yang et al. [5] designed a sparse autoencoder-based deep neural network and applied it to a synthetic case study and bearing vibration dataset to achieve multi-component composite fault diagnosis (FD). Zhu et al. [6] proposed a likelihood feature compositional domain generalization network, which extracts cross machine invariant fault features through feature likelihood distribution. The effectiveness of the compound FD was verified using multiple-bearing test benches. He et al. [7] proposed multiscale spatial-temporal residual capsule neural networks for the compound FD of industrial motors. Pu et al. [2] further combined Transformers with graph convolution to construct a three-stage hierarchical network that performs feature extraction, label-correlation modeling, and graph-based decoupling, achieving decoupled recognition of compound faults on gearbox and bearing datasets. However, existing studies still have some limitations:

(1) They are limited to part-level FDD: Most current research usually focuses on laboratory parts [8–10] such as bearings and gears. There is more severe feature coupling and interference in system-level signals involving multiple heterogeneous sub components, which exacerbates the risk of overfitting in deep learning (DL). In addition, the generalization scenario for

unseen compound fault combinations is inevitable because it is difficult to obtain all label combinations at the training stage, which requires higher generalization performance of zero-shot diagnosis.

(2) Lack of universality and interactivity: Most research is limited to the design of feature extraction strategies and neural network models, which only provide quantitative indicators of fault probability and cannot generate comprehensive reports that include causal analysis, risk grading, and actionable recommendations. In addition, these solutions typically target a single scenario and preset processes, and their static architecture and manual deployment hinder seamless interaction with users, other systems or dynamic environments. With the increasing demand for dynamic cross domain universality and human artificial intelligence (AI) collaborative intelligence in industrial applications, the inefficiency or unavailability of traditional paradigms has become increasingly prominent.

Recently, large-scale language models (LLMs) have received widespread attention owing to their powerful abilities and performance in logical reasoning, content generation, and human–computer interactions [11–13]. This general intelligence paradigm could solve the above-mentioned problems. Ma et al. [14] proposed a FD reasoning knowledge graph LLM, a novel FD pipeline enhanced by a knowledge graph that can assist in analyzing mechanical equipment faults and provide reliable guidance. Peng et al. [15] proposed a bearing health management framework that utilizes an LLM to unify multiple bearing-related tasks by processing user prompts and vibration signals. Lin et al. [16] proposed a multimodal LLM that maps multidimensional time series to fuzzy semantic embeddings through modal alignment, and then uses LLM to perform inference to achieve higher fault classification accuracy. Wang et al. [17] treated bearing FD as a visual question-answering task using envelope spectrum images combined with professional knowledge to enhance FD, thereby improving generalization and interpretability. These studies provide new methods and perspectives for FDD, verifying the potential of LLM in FDD. However, most of the LLM-based research mentioned above focuses on language dialogues based on knowledge graph retrieval, or on improving the performance of multimodal feature extraction networks by introducing images or text. However, these challenges have not yet been fully resolved.

Therefore, this study explores the application paradigm and potential of enabling LLM for system-level CFDD, particularly considering multisource signals from multiple subsystem devices containing unseen compound fault combinations. We focus on how to integrate the cognitive intelligence of LLMs with traditional small-scale deterministic diagnostic models to improve and enhance their diagnostic performance, multitasking adaptability, and human–computer interaction capabilities. We propose a cloud-edge-terminal collaborative AI (CET-CAI) agent framework to integrate the generalization awareness and human–AI interaction ability of an LLM with the deterministic reasoning ability of traditional intelligent FD models. In this framework, the LLM is deployed in the cloud and regarded as a "manager" and "summarizer" rather than an "analyst" to achieve task recognition and call for DL models and tools, which is different from previous research where LLM is typically used as part of feature extractors and requires additional multimodal data for retraining and fine-tuning with high computational costs. A novel distributed collaborative Transformer (DiCoT) is designed and embedded as an "analyst" in the edge system to achieve feature extraction, which is specifically designed for multi subsystem to reduce cross subsystem signal interference. Finally, the diagnostic results are fed back to the LLM for human–AI interaction. The contributions of this study are summarized as follows.

(1) The CET-CAI agent framework is proposed as a feasible LLM-enabled application for system-level CFDD. It can dynamically adapt to various tasks of fault detection, diagnosis and human–computer interaction, and reduce high development and communication costs of fine-tuning multimodal models and LLM.

(2) DiCoT is designed specifically for multiple subsystems, which particularly considers the heterogeneity and correlation between multiple subsystems, significantly improving the generalization of unseen compound fault combinations.

(3) Collaborative strategies based on structured language protocols (SLPs) and batch decision-making are designed for coordination between the LLM and traditional small-scale DL models to provide efficient, reliable, and comprehensive diagnostic reports.

The remainder of this paper is organized as follows. Section 2 presents the details of the proposed CET-CAI agent framework. Section 3 presents the experimental details and results. Finally, Section 4 discusses the study conclusions and future work.

2. Proposed CET-CAI agent

2.1. Overall framework

The overall framework of the CET-CAI agent for CFDD is shown in Fig. 1, which includes perception, brain, and action. This framework can be formally defined as a six-tuple system.

$$\text{CET-CAI} = \langle \mathcal{P}, \phi, \mathcal{A}, \mathcal{B}, \mathcal{K}, \psi \rangle \quad (1)$$

(1) Terminal layer: $\mathcal{P}$ is a perception module that includes multisource sensors for collecting industrial signals. This module is usually deployed on the terminal layer to sense information. ϕ represents application programming interface (API) tools (such as normalization, sliding window slicing, file dynamic reading, and initial layers of neural networks).

(2) Edge layer: $\mathcal{A}$ is an operational module used to perform feature extraction tasks, which includes specialized neural networks for feature extraction. This module is recommended to be deployed on edge layer to adaptively and quickly extract and compress features, reducing the amount of transmitted signals data and accelerating communication time. Edge layer can communicate with the cloud layer to obtain final diagnostic reports and decision guidance for workers to maintain or directly control equipment (such as turning on and off for maintenance).

(3) Cloud layer: $\mathcal{B}$ represents the brain module, whose core is an LLM used for reasoning decisions, scheduling, and human-computer interaction. Considering the inference and computational costs of LLMs, it is recommended to deploy it in the cloud. $\mathcal{K}$ is a domain expertise that stores detailed task elements and rules in JSON format (such as {"Diagnostic Object": "Bogie," "Source:" Train," " Model: "DiCoT," "Task": "Diagnosis,..."}); ψ represents the prior prompt, which includes role definition p_{role} and a JSON-based SLP p_{slp} that defines the interaction specification for the LLM.

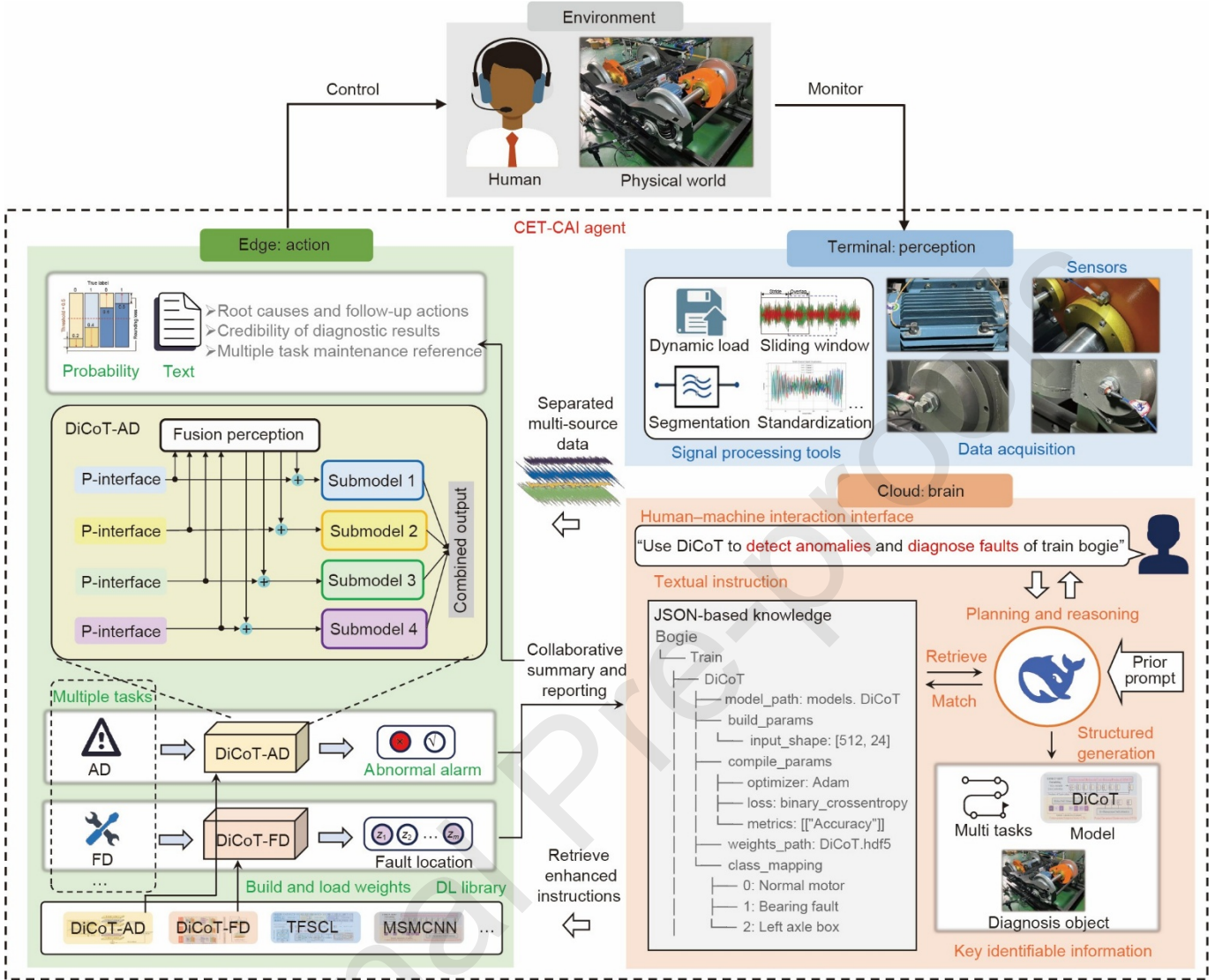


Fig. 1. Overall framework of the proposed CET-CAI agent for CFDD. AD: anomaly detection.

The perception module $\mathcal{P}$ uses multiple sensors and human-computer interaction interfaces to obtain multimodal information in the environment, such as vibration signals $\mathcal{S}$ and human text commands D_t . The signals of M subsystems are further preprocessed and standardized into structured inputs $D_s = \{D_s^m\}_{m=1}^M$ suitable for the subsequent neural networks:

$$\mathcal{P}:(\mathcal{S}, D_t) \xrightarrow{\phi} (D_s, D_t) \quad (2)$$

$$D_s = \phi_1 \circ \phi_2 \circ \dots \circ \phi_Q(\mathcal{S}) \quad (3)$$

where $\circ$ denotes the nesting of multiple APIs. $\phi_1, \phi_2, \dots, \phi_Q(\mathcal{S})$ denote the corresponding preprocessing operators, such as classic standardization and unified sliding window operations. Text data with task descriptions are transmitted to the LLM in the brain.

$$\mathcal{B}:(D_t, \mathcal{R}) \xrightarrow{p_{\text{role}}, p_{\text{slp}}} (J_{\text{json}}, \mathcal{H}) \quad (4)$$

This process mainly includes reasoning and summarizing tasks. $\mathcal{H}$ stores historical diagnostic results, and $\mathcal{R}$ denotes the diagnostic result set generated in the current round. The LLM first understands its role and SLP based on the prior role prompt, p_{role} , and protocol prompt p_{slp} , and then receives text content D_t received from the perception module and generates key information J_{json} in JSON format based on SLP.

$$J_{\text{json}} = \text{LLM}(\psi \oplus D_t) \sim \prod_{t=1}^T (v_t | v_{<t}, \psi, D_t) \cdot G(v_{t-1}, v_t) \quad (5)$$

where T is the output sequence length, v_t is the generated token indexed by t , $G(\cdot)$ is the syntax-state transition function. $v_{<t}$ represents all the token sequences generated before the token indexed by t . And each LLM prompt is generated by combining prior prompt ψ and input task text D_t , which guides the LLM to focus on key fields through prefix injection and limits the generation space. The generation process of JSON format can be represented as constraining the conditional probability of the token sequences through syntax state transition functions.

$$G(\cdot) = \begin{cases} 1 & \text{if } (v_{t-1}, v_t) \text{ conforms to JSON syntax} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Note that this process does not rely on manually designed workflows and fixed-call strategy pipelines, but is entirely determined by the LLM to organize and coordinate multiple small-scale models and tools, utilizing task-aware reasoning and language-based interaction to effectively integrate and utilize the current LLM's cognitive intelligence to cope with dynamic multitasking. Based on the key information generated by the LLM, precise and detailed JSON instructions J_{json}^* are retrieved from professional knowledge $\mathcal{K}$. The action module calls the corresponding feature extraction model f^* in the neural network library based on J_{json} to process signal data D_s from the perception module. This process is implemented using a small-scale diagnostic model with deterministic discriminative intelligence, which can alleviate the illusion risk of the LLM and avoid relying on unconstrained language reasoning in safety critical decisions. The probability obtained by the professional neural network is mapped into diagnostic results $\mathcal{R}$ based on retrieved knowledge base $\mathcal{K}$, and the final result is generated by the LLM of the brain and saved in memory:

$$\mathcal{H}^{\tau+1} = \mathcal{H}^{\tau} \cup (\mathcal{R}^{\tau+1}, \text{LLM}(\mathcal{R}^{\tau+1} \odot \mathcal{K})) \quad (7)$$

where $\mathcal{H}^{\tau}$ represents the historical results of the previous τ rounds of diagnosis, allowing users to view the results of multiple diagnoses. This design enables the LLM to provide more accurate maintenance recommendations and decisions through batch results, thereby improving diagnostic accuracy while fully utilizing interactive capabilities to simplify engineering operations. Operators can easily activate multiple AI models through dialogue to respond flexibly to various task diagnoses and avoid relying on professional AI deployment and knowledge development.

2.2. Structure of DiCoT

The design of neural networks for feature extraction is at the core of addressing the CFDD challenge, which should fully consider the characteristics of the diagnostic object and signal to achieve high performance. Unlike most previous studies on simple components, this study focuses on system-level multicomponent systems, where there is significant heterogeneity between subsystems such as bearings and gearboxes; however, there are also potential correlations owing to the rigid connections between them. Most traditional FDD solutions use a single neural network model to centralize feature extraction from multiple sensors, which may further amplify the risk of overfitting caused by the heterogeneity of multiple subsystems, particularly in more challenging generalization scenarios involving unseen compound fault combinations. Therefore, a distributed collaborative modeling architecture is proposed, as shown in Fig. 2, to embed the action module of the AI agent in the edge system, specially addressing the system-level CFDD. The distributed multibranch structure of DiCoT explicitly decouples these features, allowing independent attention mechanisms to learn component-specific representations. To demonstrate the necessity of this distributed design over centralized approaches rigorously, we provide a theoretical analysis based on optimization dynamics and probabilistic factorization.

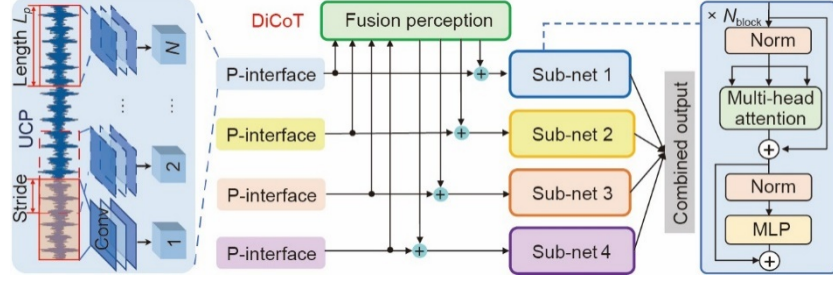


Fig. 2. Architecture of the DiCoT. Each P-interface can be deployed on the terminal for data compression, and subsequent layers can be placed on edge devices. Each subsystem corresponds to a type of component. UCP: unified convolutional patch; MLP: multilayer perceptron; Conv: convolution; N : number of temporal patches; N_{block} : number of stacked encoder blocks.

Let $x = [x_1, \dots, x_M]$ denote the multisubsystem input signals (e.g., gearbox, bearing, and axle box), where x_m belongs to the input space of the subsystem indexed by m , and let $y = [y_1, \dots, y_M]$ denote the corresponding subsystem-wise fault labels. Consider a centralized diagnostic model $f_\theta(x)$ trained by minimizing the empirical risk ($\mathcal{L}(\theta)$):

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y)}[\ell(f_\theta(x), y)] \quad (8)$$

where $\ell(\cdot)$ denotes the differentiable surrogate loss (e.g., binary cross-entropy), and $\mathbb{E}_{(x,y)}$ denotes the expectation over the data distribution.

For system-level CFDD, gradient amplitude dominance and conflict are the key issues. Owing to the heterogeneous signal-to-noise ratios and energy scales among the subsystems, it is common that $\|x_m\| \gg \|x_{m'}\|$ (e.g., high-energy gearbox vibrations dominate weak bearing signals). For centralized encoders with shared parameters θ (e.g., a shared convolutional kernel W), the gradient of the loss with respect to parameters scales with the input magnitude

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial h} \cdot \frac{\partial h}{\partial W} \propto x \quad (9)$$

where $\mathcal{L}$ is the empirical loss, h is the intermediate representation. This implies that gradient updates are dominated by high-energy subsystems, that is, $\|\nabla_{\theta_m} \mathcal{L}\| \gg \|\nabla_{\theta_{m'}} \mathcal{L}\|$, where $\nabla_{\theta_m} \mathcal{L}$ and $\nabla_{\theta_{m'}} \mathcal{L}$ denote the gradient components associated with subsystem indices m and m' , respectively. Moreover, heterogeneous discriminative cues across subsystems may induce gradient conflict, characterized by negative cosine similarity

$$\cos(\nabla_{\theta_m} \mathcal{L}, \nabla_{\theta_{m'}} \mathcal{L}) < 0 \quad (10)$$

leading to destructive interference in the shared parameters and suppressing the optimization of weak but informative fault features. These effects indicate that unconstrained parameter sharing is suboptimal for heterogeneous system-level CFDD, motivating the explicit architectural decoupling of the gradient propagation paths.

To address the above gradient dominance and conflict at the modeling level, DiCoT adopts a distributed multibranch structure

$$f(x) = g(h_1(x_1), \dots, h_M(x_M), h_{\text{FP}}(x)) \quad (11)$$

where $g(\cdot)$ denotes the fusion function that aggregates the subsystem-specific branches and the FP branch. This aligns with a standard factorization assumption commonly used in Bayesian diagnosis:

$$P(y|x) \approx P(y_{\text{coupling}}|x) \cdot \prod_{m=1}^M P(y_m|x_m; \theta_m) \quad (12)$$

where y_{coupling} denotes the coupling-dependent system state that captures cross-subsystem interactions.

This formulation explicitly separates subsystem-specific likelihoods from shared coupling effects, enabling each branch h_m to focus on component-specific representations, whereas the fusion module h_{FP} captures the global system interactions. Importantly, this factorization constrains the information and gradient flow across subsystems by design, rather than relying on implicit regularization.

From an optimization perspective, this design achieves decoupling optimization based on Block-diagonal Hessian matrices, effectively reducing the gradient conflicts between parameters. Let us partition the parameter set as $\theta = (\theta_1, \dots, \theta_M, \theta_{\text{FP}})$, where θ_m corresponds to the subsystem branch indexed by m , and θ_{FP} corresponds to the fusion module. Under the factorized structure above, the empirical loss can be written as

$$\mathcal{L}(\theta) \approx \sum_{m=1}^M \mathcal{L}(\theta_m, \theta_{\text{FP}}) \quad (13)$$

which implies that the cross-partial derivatives between the independent branch parameters are negligible:

$$\frac{\partial^2 \mathcal{L}}{\partial \theta_m \partial \theta_{m'}} \approx 0 \quad (14)$$

Consequently, the Hessian matrix exhibits an approximately sparse, bordered, and block-diagonal structure:

$$\mathbf{H} = \nabla_{\theta}^2 \mathcal{L} \approx \begin{bmatrix} H_1 & 0 & \cdots & C_1 \\ 0 & H_2 & \cdots & C_2 \\ \vdots & \vdots & \ddots & \vdots \\ C_1^T & C_2^T & \cdots & H_{\text{FP}} \end{bmatrix} \quad (15)$$

where H_m represents the curvature associated with subsystem index m and C_m captures its coupling with the fusion module. The absence of strong off-diagonal interactions between the subsystems indicates reduced parameter interference, which theoretically stabilizes the optimization and supports compositional generalization to unseen compound fault combinations.

Guided by this theoretical framework, we use advanced DL components to establish a DiCoT architecture that effectively captures long-term dependencies within decoupled branches. Specifically, Transformer structures have a high potential for large-scale complex information and are being applied in FDD. Developing effective patch techniques can better adapt to the discreteness of temporal signals and reduce computational consumption compared to directly applying the original Transformer. We adopt a convolutional operation with a large stride and kernel to construct a unified convolutional patch (UCP) interface, as shown in Fig. 2, which not only integrates the local perception advantages of a convolutional neural network (CNN) but also efficiently unifies heterogeneous shape inputs from different branches through feature dimensionality reduction with lower computational costs. This process can be deployed within sensors along with signal slicing and can be standardized based on fog-computing technology [18–20]. This means that sensors do not transmit large-scale raw data to the edge system, but rather transmit compressed features. In this experiment, the original dimensions of the largest subsystem were reduced from 512×10 to 32×64 and the amount of data required to be transmitted was reduced by 60%. The patch indexed by n in subsystem interface m can be represented as:

$$p_m^{(n,D)} = \{p_m^{(n,d)}\}_{d=1}^D = \{\sum_{l=1}^{L_p} \sum_c^C x^{(s(n-1)+l,c)} w_d^{(l,c)}\}_{d=1}^D \quad (16)$$

where $w_d^{(l,c)}$ is the weight of the convolution kernel, and $x^{(s(n-1)+l,c)}$ is the corresponding value of the convoluted signal. D and s are the number of convolution kernels and convolution stride, respectively. c and C denote the channel index and the total number of channels, respectively. This operation is consistent with the calculation pattern of a one-dimensional convolution, where the length of the convolution kernel is equal to the consistency of mapping range L_p corresponding to the generated patch. Its stride is set to half that of the convolution kernel.

Considering the coupling correlation between multiple subsystems, a fusion perception (FP) module is proposed to extract the correlation features between UCPs, as shown in Fig. 3. Multiple independent UCP features are connected into $p^{(N,M \times D)} = \{\{p_m^{(n,D)}\}_{m=1}^M\}_{n=1}^N$, which aligned in the time dimension. Owing to the different feature scales of different subsystems, a multiscale feature extraction strategy is adopted to enhance the comprehensiveness of the features, which can achieve scaling with a distance of r between two elements based on a dilated convolution with D_r kernels:

$$p_r^{(n,D_r)} = \{\sum_{l=1}^{L_p^r} \sum_{d=1}^{M \times D} p_m^{(s(n-1)+1+r \cdot (l-1),d)} w_{d_r}^{(l,d)}\}_{d_r=1}^{D_r} \quad (17)$$

$$\hat{p}^{(n,D_r)} = \frac{\sum_{r=1}^{N_r} p_r^{(n,D_r)}}{N_r} \quad (18)$$

where L_p^r and $\{M \times D\}$ are the length and width of the one-dimensional dilated convolution kernel, respectively. D_r is the number of convolution kernels for the dilated convolution. The stride of the dilated convolution is usually set to 1 to reduce information loss. N_r is the number of branches at different scales. $\hat{p}^{(n,D_r)}$ denotes the fused multiscale correlation feature.

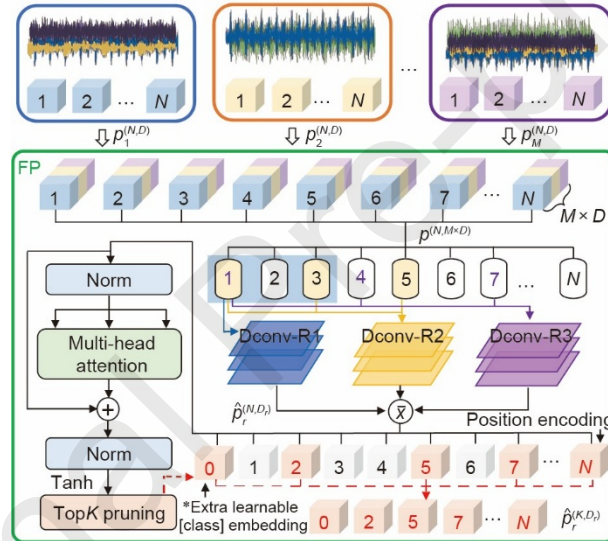


Fig. 3: The structure of FP. Pruning targets the $N + 1$ patches before the attention mechanism. Dconv: dilated convolution.

Internal external position coding [21] is added to continue the design approach of the Transformer architecture, providing learning for encoding the position information between and within patches:

$$\hat{p}_r^{(N+1,D_r)} = [p^{(0,D_r)}; \hat{p}^{(N,D_r)}] + p_{ie} \quad (19)$$

where p_{ie} denotes the internal-external positional encoding. Considering the significant interference features in the coupling information between multiple subsystems, we designed a pruning structure to obtain weight W_p , including layer normalization (LN), residual connections, and a multi-head self-attention module (MSA):

$$W_p = f_{\tanh}(f_{\ln}(f_{\text{msa}}(f_{\ln}(\hat{p}_r^{(N+1,D_r)})) + \hat{p}_r^{(N+1,D_r)})) \quad (20)$$

where $f_{\ln}(\cdot)$, $f_{\text{msa}}(\cdot)$, and $f_{\tanh}(\cdot)$ denote layer normalization, multi-head self-attention, and Tanh activation, respectively.

This operation through the attention mechanism is equivalent to measuring the degree of mutual correlation within different time blocks to obtain weight $W_p = \{w_p^{(n,d_r)}\}_{d_r=1}^{D_r} \}_{n=1}^{N+1}$. The larger the value of W_p through the attention mechanism, the more significant the common global correlation features involved in this feature. The weight W_s^n of the temporal patch indexed by n is calculated through a summation operation on the dimension of the patch:

$$W_s^n = \sum_{d_r=1}^{D_r} w_p^{(n,d_r)} \quad (21)$$

The patch indices, $I_k = \{n, W_s^n \in \text{Top}K(W_s)\}$, corresponding to the first K values in the weight vector are exported to guide the retention of the key patches:

$$\hat{p}_r^{(K,D_r)} = \hat{p}_r^{(N+1,D_r)}[I_k] = \{\hat{p}_r^{(k,D_r)}\}_{k=1}^K \quad (22)$$

The correlation features, $\hat{p}_r^{(K,D_r)}$, of the subsystems after pruning are connected to the independent features extracted by each UCP and then synchronously input into the subsequent subnetworks as independent branches. The subnetworks continue the design of the Transformer encoder structure, including LN, a MSA mechanism, fully connected layers, and a residual structure, as shown in Fig. 2.

Each subnetwork independently outputs the predicted results of the corresponding subsystem based on a multilabel classifier to avoid interference between the final decision features of different structural components.

$$L^m(y_i^m, z_i^m) = -\frac{1}{n_c} \sum_{n_c} [y_i^m \cdot \log(z_i^m) + (1 - y_i^m) \cdot \log(1 - z_i^m)] \quad (23)$$

where z_i^m and y_i^m are the predicted results and the corresponding true labels for the sample indexed by i in the subsystem indexed by m , respectively. n_c denotes the number of categories corresponding to this subsystem. L^m is the loss of the sub-network indexed by m . For each round of DiCoT training on the sample indexed by i , a collaborative loss L_{col} is output as follows:

$$L_{\text{col}}(\{y_i^m\}_{m=1}^M, \{z_i^m\}_{m=1}^M) = \frac{1}{M} \sum_{m=1}^M L^m(y_i^m, z_i^m) \quad (24)$$

2.3. Collaborative strategies

Unlike the traditional fixed neural network output probability pattern, the proposed AI agent can utilize the LLM to call diverse tools and functions to achieve collaborative summarization and reporting. Traditional diagnostic decision-making models use sample-predicted values to obtain results with high real-time performance. However, the limited available information within a single window makes it prone to overfitting and false positives because not all windows in the signal segment contain fault information. Therefore, batch decision-making is beneficial for improving the accuracy of diagnosis and reducing false positives. Let the signal output in a batch be

$$P_b = \{z_i\}_{i=1}^b = \begin{bmatrix} z_1^1 & \cdots & z_1^{N_c} \\ \vdots & \ddots & \vdots \\ z_b^1 & \cdots & z_b^{N_c} \end{bmatrix} \quad (25)$$

where N_c is the total value of the state categories that contain all the subsystems, z_i is the prediction vector indexed by i , and b is the batch number of decision samples. After rounding, the batch sliding window samples from the same signal source were averaged to obtain the final decision probability R .

$$R = \frac{1}{b} \sum_{i=1}^b \text{Round}(P_b) = [\hat{z}_1, \hat{z}_2, \dots, \hat{z}_{N_c}] \quad (26)$$

where $\text{Round}(\cdot)$ denotes element-wise rounding, and $\hat{z}_{N_c}$ is the final batch-averaged decision vector.,

This batch decision-making strategy can improve the stability of the diagnostic results and reduce false alarm rates. In addition, the LLM can integrate the results of multiple tasks for batch judgment. For example, when abnormal detection and FD problems occur simultaneously, the LLM receives two prediction results, R_{AD} and R_{FD} , and then provides comprehensive recommendations based on these results

$$(R_{AD}, R_{FD}) \xrightarrow{\mathcal{K}} (\mathcal{R}_{AD}, \mathcal{R}_{FD}, D_t) \xrightarrow{\text{LLM}} \mathcal{H} \quad (27)$$

where $\mathcal{R}_{AD}$ and $\mathcal{R}_{FD}$ are the readable texts of the AD and FD models mapped by the knowledge base, respectively. These multitasking results are uniformly input into the LLM to obtain comprehensive multitasking reports and are saved as $\mathcal{H}$. Through the above paradigm, the LLM collaborates with professional small-scale models to address diverse CFDD challenges and provide readable and comprehensive reports, including a summary and analysis of small-scale specialized model (SSM) diagnosis results, feasibility evaluation of the model and diagnosis results, further root cause analysis, as well as subsequent maintenance strategies and recommendations.

Table 1

Label description.

Component	Label	State
Motor	M0	Normal
	M1	Short circuit
	M2	Broken rotor bar
	M3	Bearing fault
	M4	Bowed axis
Gear	G0	Normal
	G1	Gear cracked tooth
	G2	Gear worn tooth
	G3	Gear missing tooth
	G4	Gear chipped tooth
	G5	Bearing inner race fault

Component	Label	State
	G6	Bearing outer race fault
	G7	Bearing rolling element fault
	G8	Bearing cage fault
	LA0	Normal
	LA1	Bearing inner race fault
Axle box (left)	LA2	Bearing outer race fault
	LA3	Bearing rolling element fault
	LA4	Bearing cage fault
	RA0	Normal
Axle box (right)	RA1	Bearing inner race fault

3. Experiment

3.1. Experimental setup

A system-level dataset of a multicomponent mechanical system is used to validate the effectiveness of the proposed solution. It is constructed as a proportionally replicated system-level platform with the same mechanical topology as that of a real industrial bogie transmission system [22,23]. The experimental platform includes motors, gears, and axle boxes, all of which are representative core transmission components that operate synchronously and are coupled. It contains synchronized subsystems, coupled transmission paths, and heterogeneous multisensor measurements collected under controlled operating conditions. The detailed fault categories and corresponding labels are shown in Table 1. Unlike previous single-fault datasets, this dataset includes complex component-level and system-level-associated compound faults. The compound fault scenarios in this dataset are not synthesized by mathematically concatenating the single-fault signals. Instead, they are generated through simultaneous physical fault injection on the high-speed train bogie system. Consequently, the collected vibration signals capture the nonlinear mechanical coupling and cross-subsystem structural interference inherent to real-world machinery. The data for each system state were obtained from 24 sensors distributed across various subcomponents, with a sampling frequency of 64 kHz. We chose the operating condition of driving the motor at a speed of 20 Hz for verification, and the specific dataset division is shown in Table 2 and Fig. 4. Notably, the setting of the test samples does not focus on source domain diagnosis or different operating condition diagnosis of the same category as in previous works, but rather on a more realistic challenge of locating unseen compound fault combinations. This approach is more closely aligned with engineering requirements because it is difficult to ensure that a combination of compound faults occurs during training.

The experiment comprehensively demonstrated the effectiveness and superiority of the proposed approach in multiple dimensions:

- (1) Diagnostic performance of the proposed DiCoT.
- (2) Function expansion and improvement for the AI agent.
- (3) Communication costs of the collaborative framework.

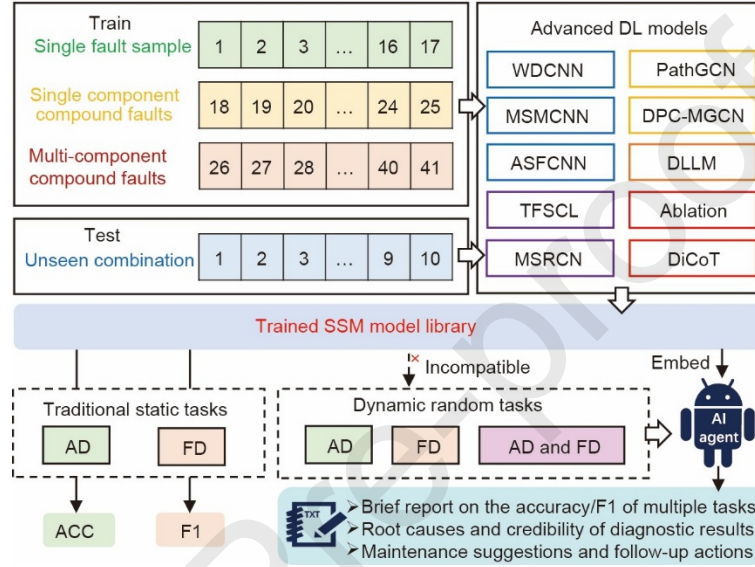


Fig. 4. Experimental setup. The fault types of the samples in the training and testing sets are different, which are independent of each other without data leakage.

The comparative experiment consisted of two tasks: AD and FD. AD is mainly used for the coarse-grained determination of abnormalities in the components of the bogie system, whereas FD is used to directly diagnose the type of fault. For example, the label of M0_G0_LA1 + LA2 + LA3 + LA4_RA0 in AD is [0,0,1,0], and the label in FD is [1,0,0,0,0, 1,0,0,0,0,0,0, 0,1,1,1,0]. The hyperparameters of DiCoT are listed in Table 3, inheriting a lightweight design [21]. Some advanced methods, including WDCNN [24], TFSCN [25], MSMCNN [26], ASFCNN [27], MSRCN [7], PathGCN [28], DPC-MGCN [29], data-based LLM FD (DLLM) [30], are selected for training to compare their performance on independent tasks with DiCoT. In this dataset, the signals of each composite fault type are collected in an independent file. Each file is randomly sampled based on a sliding window of length 512 to obtain 500 training samples. The total number of training samples is 20500. All the models are trained using the Adam algorithm for 20 epochs with a learning rate of 0.001. The proposed DiCoT model has parameters of 3.3×10^6 , which can be considered a small lightweight model. To avoid randomness in training the neural networks, we repeat the complete experimental process five times, including data slicing, model training, and the evaluation of a test set containing 5000 samples.

Table 2

Dataset settings.

Role	Number	Type
Train	1	M0_G0_LA0_RA0

Role	Number	Type
	2	M1_G0_LA0_RA0
	3	M2_G0_LA0_RA0
	4	M3_G0_LA0_RA0
	5	M4_G0_LA0_RA0
	6	M0_G1_LA0_RA0
	7	M0_G2_LA0_RA0
	8	M0_G3_LA0_RA0
	9	M0_G4_LA0_RA0
	10	M0_G5_LA0_RA0
	11	M0_G6_LA0_RA0
	12	M0_G7_LA0_RA0
	13	M0_G8_LA0_RA0
	14	M0_G0_LA1_RA0
	15	M0_G0_LA2_RA0
	16	M0_G0_LA3_RA0
	17	M0_G0_LA4_RA0
	18	M0_G4 + G5_LA0_RA0

Role	Number	Type
	19	M0_G2 + G5_LA0_RA0
	20	M0_G3 + G5_LA0_RA0
	21	M0_G1 + G5_LA0_RA0
	22	M0_G0_LA1 + LA2_RA0
	23	M0_G0_LA1 + LA2 + LA3 + LA4_RA0
	24	M0_G0_LA2 + LA4_RA0
	25	M0_G0_LA1 + LA2+LA3_RA0
	26	M2_G0_LA1_RA1
	27	M2_G3_LA0_RA0
	28	M1_G3_LA0_RA0
	29	M1_G5_LA0_RA0
	30	M4_G5_LA0_RA0
	31	M1_G0_LA1_RA0
	32	M3_G0_LA1_RA0
	33	M1_G0_LA1_RA1
	34	M0_G0_LA1_RA1
	35	M0_G3_LA1_RA0

Role	Number	Type
	36	M1_G5_LA1_RA1
	37	M4_G3_LA1_RA1
	38	M4_G5_LA1_RA0
	39	M2_G5_LA1_RA0
	40	M1_G3_LA1_RA0
	41	M4_G0_LA1_RA0
	1	M0_G0_LA1 + LA2 + LA4_RA0
	2	M0_G0_LA2 + LA3_RA0
	3	M1_G5_LA1_RA0
	4	M2_G0_LA1_RA0
	5	M2_G3_LA1_RA0
Test	6	M2_G5_LA0_RA0
	7	M3_G3_LA0_RA0
	8	M3_G5_LA0_RA0
	9	M4_G3_LA0_RA0
	10	M4_G3_LA1_RA0

Table 3

Experimental hyperparameters.

Hyperparameter	Description	Value
L	Length of each sample channel	512
M	Number of subsystems	4
D	Number of convolution kernels	64
L_p	Length of UCP	32
s	Stride of UCP	16
D_r	Number of dilated convolution kernels	64
L_p^r	Length of dilated convolution kernels	3
N_r	Number of branches at different scales	3
N_{block}	Number of stacked encoder blocks	5
K	Number of pruned patches	10
b	Batch number of decision samples	100

However, traditional SSM relies on preset static processes and fixed output labels, which cannot meet the requirements of dynamic random tasks. Traditional SSM can be seamlessly integrated into the action module of the AI agent frameworks with greater flexibility. DeepSeeker R1-8B is selected as the LLM. The proposed AI agent-based framework can enhance the performance of independent tasks with batch decision-making mechanisms as well as meet the requirements of dynamic random tasks based on text prompts, as shown in Fig. 4.

Table 4

Experimental diagnostic results. SD represents the standard deviation.

Method	AD	FD
--------	----	----

	Accuracy (%)	SD	F1 (%)	SD
WDCNN	86.71	1.476	59.48	0.0117
MSMCNN	84.84	0.992	57.30	0.0155
MSRCN	89.92	1.096	67.60	0.0144
ASFCNN	82.96	1.425	48.99	0.0097
TFSCCL	85.16	2.123	70.99	0.1357
PathGCN	83.81	2.493	50.97	0.0556
DPC-MGCN	81.54	2.498	54.63	0.0396
DLLM	89.00	0.890	67.34	0.0038
DiCoT-FP	87.36	0.666	72.28	0.0061
DiCoT-Di	86.55	1.825	68.37	0.0057
DiCoT	91.37	0.022	75.94	0.0059

The bold font represents the best result in each column.

3.2. Comparison of diagnostic performance

The average values of the five results are summarized in Table 4. The experimental results show that both AD and FD tasks for unseen compound fault generalization are challenging. Some advanced models can achieve 100% accuracy on the training set and over 95% generalization under tasks of varying speeds (but consistent diagnostic categories). However, the feasibility and generalization performance of the results decrease rapidly for samples with unseen category combinations.

For the AD tasks, the accuracy of most models does not exceed 90%. Some models with hybrid feature extraction strategies perform better than models with CNN architectures. For example, MSRCN based on capsule neural networks achieved an accuracy of 89.92%, and DLLM based on the Transformer architecture achieved an accuracy of 89%. The accuracy of the proposed DiCoT reaches 91.37% with a low standard deviation of only 0.022, which is superior to that of all the comparison methods. We conducted ablation experiments by removing the proposed FP module and the distributed architecture. The performance of the ablated models decreased to 87.36% and 86.55%, verifying the effectiveness of both designs.

For challenging FD tasks, the performance degradation of most models was significant. Some models had F1 scores lower than 60%. Models with special design strategies exhibited better performance. For example, the signal processing-informed

TFSCCL model achieved an F1 score of 70.99%, which is better compared to those of most CNN-based models such as WDCNN with 59.48% and MSMCNN with 57.30%. The F1 scores of MSRCN based on the capsule neural network and DLLM based on the Transformer architecture in FD task are 67.60% and 67.34%, respectively. The proposed DiCoT model achieved the highest F1 score of 75.94% among all methods. After removing the FP module, the F1 score of DiCoT-FP is 72.28%, which is still higher than that of TFSCCL. After removing the distributed design architecture, the F1 score of DiCoT-Di is 68.37%. This indicates that the integration of perception modules and the distributed architecture effectively improves the performance of FD tasks. To facilitate a deeper understanding of model performance across different mechanical components, Table 5 presents the subsystem-level evaluation metrics, including F1-score, precision, and recall. It can be observed that the diagnostic difficulty of different components varies, as there are significant differences in label space categories, specifically for the compositional generalization of gear systems. Most DL models struggle to extract effective features, with F1 scores exceeding 30%. For the left axle box, WDCNN and the other methods achieved only 59.34% F1. However, the DiCoT series maintained high robustness, exceeding the baseline by more than 20%. The proposed DiCoT maintains the most balanced and consistent performance among all four subsystems, and this distributed structure can alleviate the interference between different subsystems in a multicomponent composite FD, thereby reducing the risk of overfitting.

Table 5

Comparison of subsystem level indicator results.

Method	Motor			Gear			Left Axle Box			Right Axle Box		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
WDCNN	51.85	42.00	46.41	39.56	36.01	37.70	59.34	53.99	56.54	94.95	93.99	94.47
MSMCNN	81.08	30.00	43.80	31.08	23.01	26.44	73.68	41.99	53.50	100	100	100
MSRCN	68.13	61.99	64.92	46.94	45.99	46.46	69.39	68.00	68.69	91.43	96.01	93.66
ASFCNN	38.30	17.99	24.49	30.51	17.99	22.64	37.10	23.01	28.40	99.00	99.00	99.00
TFSCCL	88.31	68.01	76.84	41.89	30.99	35.63	70.11	61.01	65.24	100	100	100
PathGCN	42.11	23.99	30.57	39.76	33.01	36.07	56.63	47.02	51.37	95.15	97.99	96.55
DPC-MGCN	55.93	33.02	41.51	29.27	23.99	26.37	50.00	46.99	48.45	92.93	91.99	92.46
DLLM	76.47	51.99	61.90	34.07	30.99	32.46	73.03	64.99	68.78	95.00	95.00	95.00
DiCoT-FP	75.26	72.99	74.11	35.05	34.01	34.52	92.71	89.01	90.82	100	100	100

DiCoT-Di	69.47	65.99	67.69	40.22	36.99	38.54	66.35	69.01	67.65	91.92	91.01	91.46
DiCoT	75.51	74.01	74.75	44.34	46.99	45.63	87.88	87.04	87.44	100	100	100

The bold font represents the best result in each column.

To further demonstrate the impact of the model structure on performance, visualization based on t -distributed stochastic neighbor embedding (t -SNE) is performed, as shown in Fig. 5. The decision layer and final output features of the three DiCoT subnets are visualized on the AD task. It can be observed that different subnets focus only on the label features related to the subsystems. For example, L0 and L1 in Fig. 5(a) form clusters. The motors in these two samples are normal, whereas those in the remaining samples are abnormal. The L5 to L8 samples shown in Fig. 5(c) form clusters. The axle boxes in these samples are normal, whereas those in the remaining samples are abnormal. This phenomenon indicates that the distributed structure can focus on the corresponding subsystem features, thereby avoiding feature interference caused by significant structural differences between the different subsystems. This further demonstrates the effectiveness of the proposed module, which is one of the main reasons for the superiority of the distributed architecture. In addition, distributed networks decouple compound labels through a structural design to ensure independence between sublabels and reduce the coupling interference caused by changes in label combinations. The final output successfully classified all the samples, forming three major clusters: $[1,1,0,0]$, $[0,0,1,0]$, and $[1,1,1,0]$.

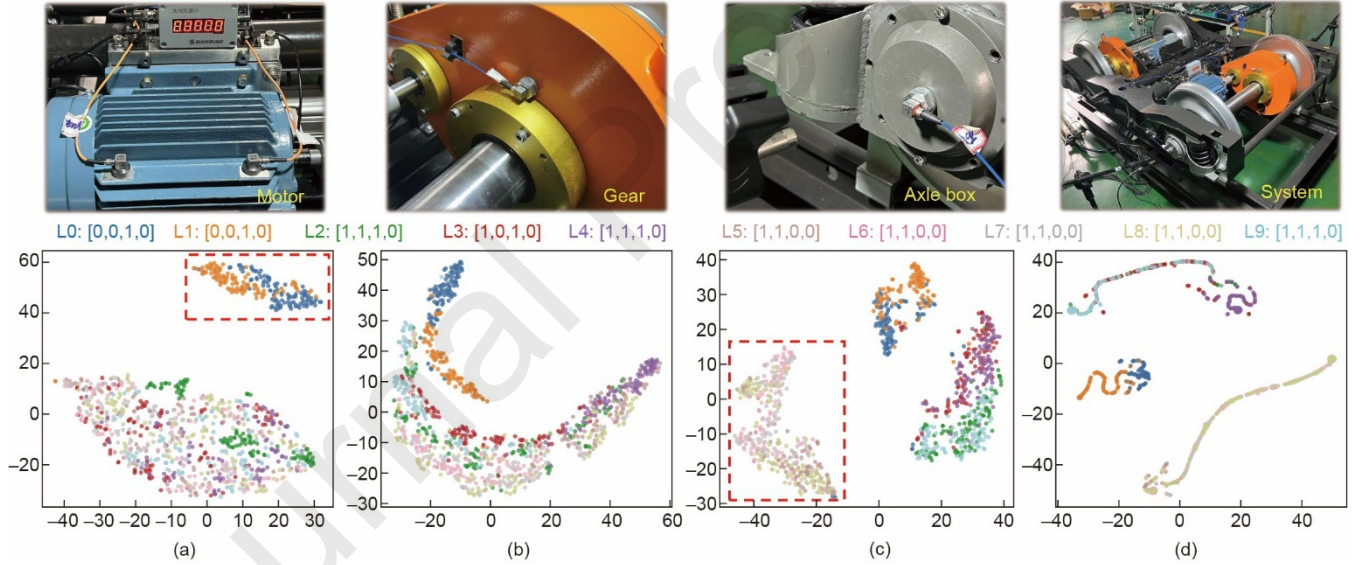


Fig. 5. Visualization of t -SNE for decision layer features. (a) Motor; (b) Gear; (c) Axle box; (d) System. L0–L9 represents nine types of samples. $[1,0,1,0]$ represents abnormalities in the motor and left turn axle box.

3.3. Function improvement empowered by AI agent

The proposed AI agent can directly improve the generalization and diagnostic stability of SSMs through batch decision-making. The mean of the five experimental results is listed in Table 6. It can be observed that the performance of all SSMs improved under the architecture of the AI agents. For example, in AD tasks, the accuracy of WDCNN and MSRCN increased by 6.29% and 3.08%, respectively, both reaching 93.0%. On the more challenging FD tasks, the improvement is more significant, with PathGCN's F1 score increasing by 20.19% and DPC-MGCN's F1 score increasing by 20.48%. TFSCN and DLLM, which originally performed well on FD tasks, achieved growth rates of 9.78% and 13.95%, respectively, under LLM coordination. The proposed DiCoT model achieved the best results, with an accuracy of 93.5% and an F1 score

of 82.85%. This further demonstrates the superiority of the designed DiCoT network and the effectiveness of LLM's global perspective and batch decision-making.

Table 6

Performance improvement results of LLM-based batch decision-making.

Method	AD	FD		
	Accuracy (%)	GR (%)	F1 (%)	GR (%)
WDCNN	93.0	+6.29	77.47	+17.99
MSMCNN	85.5	+0.66	71.86	+14.56
MSRCN	93.0	+3.08	79.48	+11.88
ASFCNN	88.5	+5.54	67.79	+18.80
TFSCCL	85.5	+0.34	80.77	+9.78
PathGCN	90.0	+6.19	71.16	+20.19
DPC-MGCN	89.0	+7.46	75.11	+20.48
DLLM	92.5	+3.50	81.29	+13.95
DiCoT-FP	87.5	+0.14	78.81	+6.53
DiCoT-Di	90.0	+3.45	80.94	+12.57
DiCoT	93.5	+2.13	82.85	+6.91

GR: the growth rate compared to SSM static tasks.

To further demonstrate the effectiveness of global batch decision-making, we visualized the prediction results of a batch decision-making SSM for gearbox faults, as shown in Fig. 6. It can be observed that out of 100 consecutive diagnoses, 77 of the SSM diagnostic results had a fault probability exceeding 0.5, indicating correct judgment. There were 23 diagnostic results with a probability below 0.5 that failed to detect abnormalities, resulting in false alarms. This may be because the fault information does not appear at every moment in a signal, which is common in industrial practice. By utilizing the

global decision-making of AI agents, these impacts can be further reduced to improve the credibility and precision of the decisions.

In addition to intuitive improvement in decision-making performance, the core advantage of the CET-CAI agent framework is its flexibility and universality, which can provide more humanized and intelligent suggestion generation capabilities for diagnostic tasks. Traditional feature extraction models can only handle static preset tasks through manual deployment and invocations. By seamlessly integrating SSM into the proposed AI agent architecture, more flexible and comprehensive functionality can be achieved to address a wider range of tasks. We develop a software interface as a case study based on the CET-CAI agent, as shown in Fig. 7. It can be observed that the AI agent can dynamically organize and call multiple SSMs synchronously for multitask diagnosis and collaborative decision-making by processing unstructured language instructions from humans, without being limited to specific preset scenarios or task types. Moreover, the output of the system is not just a simple probability of failure but also includes a wider range of textual reports, including potential root cause analysis, reliability of diagnostic results, and subsequent technical optimization and maintenance recommendations. Therefore, this architecture with higher human-like cognitive intelligence provides a powerful and scalable platform for solving complex problems that require multiple models, multistage collaboration, and human–computer interaction. Its application potential significantly exceeds the current scope of anomaly detection and FD. Notably, CET-CAI agents are positioned as auxiliary decision supports in industrial practice, rather than as substitutes for human expert decision-making. This follows the positioning of most current intelligent diagnostic technologies. In several real-world diagnostic scenarios, human operators combine model outputs with domain knowledge, environmental operating conditions, and safety regulations. In addition, the quantitative evaluation of user experience, operator workflow delays, and technicians' trust in the maintenance recommendations derived from this technology are topics worthy of future research. More attention should be paid to these human-computer interaction challenges.

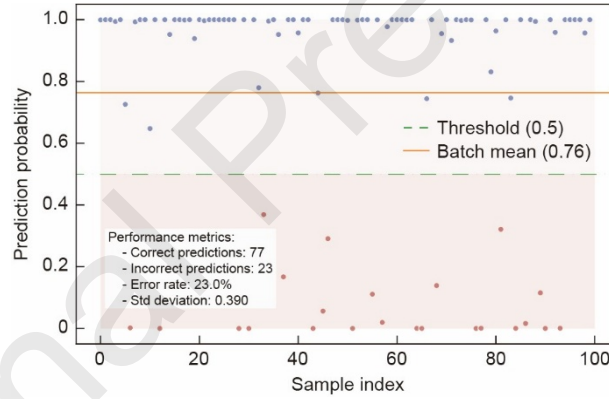


Fig. 6. Batch decision results. Std: standard deviation.

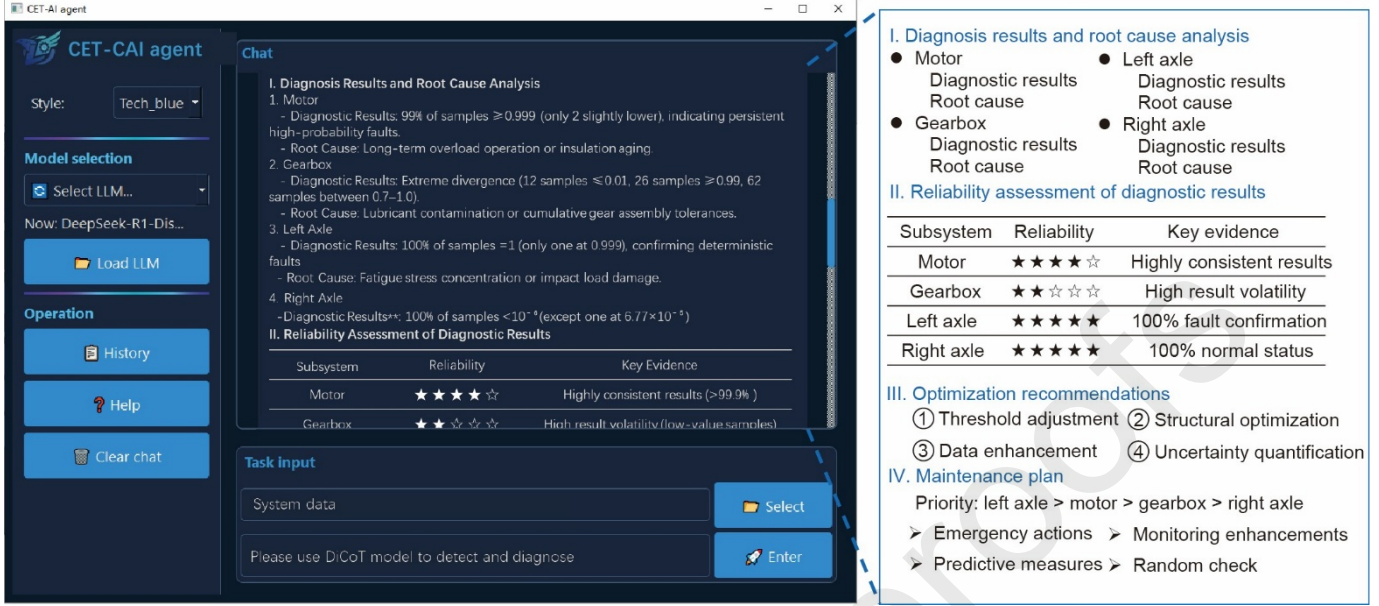


Fig. 7. The software system interface of CET-CAI agent.

3.4. Communication costs of CET-CAI agent

The main task of the CET-CAI agent is to address the precise business response, real-time response, and dynamic scenario generalization requirements in industrial general intelligence, which is different from traditional cloud computing and general AI agents used for daily human business. For intelligent FD methods, most DL-based models directly process raw vibration signals in an end-to-end manner, particularly deep convolution and Transformer-based models that rely on high-resolution time series inputs. Domain-adaptive diagnostic methods and LLM related methods require access to target domain signals for additional training, fine-tuning, or feature extraction. These settings typically require cloud deployment; additionally, they incur significant communication and deployment costs. The proposed CET-CAI intelligent agent reconstructs the diagnostic paradigm through cloud-edge collaboration, with special consideration given to the data transmission costs and real-time requirements from the design of DiCoT. The relevant parameter settings are as follows: set the original vibration data of the terminal to 512×24 sampling points (corresponding to 49152 bytes), and compress it into a 20×64 feature matrix (corresponding to 5120 bytes) at the terminal using Eq. (16). The transmission from the terminal to edge devices uses 1 Mbps broadband with an additional 10% data encapsulation overhead, whereas the transmission from edge devices to the cloud uses a 0.5 Mbps network with an additional 15% data encapsulation overhead. The performance comparison of the frameworks is presented in Table 7.

Table 7

Framework performance comparison for 100 diagnostic results.

Stage	Traditional cloud computing		Proposed CET-CAI agent		Improvement
	Expression	Value	Expression	Value	
Terminal collection (ms)	100×8	800	100×8	800	0

Terminal to edge (ms)	—	—	$100 \times (5120 \times 1.1)/(1 \text{ Mbps})$	5632	—
Edge computing (ms)	—	—	10×100	1000	—
Edge to cloud (ms)	$100 \times (49152 \times 1.15)/(0.5 \text{ Mbps})$	11311	$1 \times (8400 \times 1.15)/(0.5 \text{ Mbps})$	193	98.29%
Reaction time (s)	$(800 + 11311) \times 10^{-3}$	12.104	$(800 + 5632 + 10 + 193) \times 10^{-3}$	1.382	88.58%
Transmission (KB)	$100 \times (49152 \times 1.15)$	5652.5	$100 \times 5120 \times 1.1 + 1 \times 8400 \times 1.15$	572.86	89.86%

Mbps: megabits per second.

It can be observed that in 100 diagnostic decision tasks, traditional cloud computing requires a data transmission cost of 5652.5 KB, whereas the proposed CET-CAI agent architecture decomposes the transmission burden into lightweight feature streams from the terminal to the edge. Because raw data no longer need to be transmitted to the cloud, only a decision vector with a length of 21 needs to be transmitted, and the transmission time to the cloud significantly decreased from 11311 ms to 193 ms, with a drastic decline of 98.29% in transmission time. In addition, because of the independent stages of signal data acquisition, SSM model inference, and data transmission, which can be processed in parallel, the single-inference time is shorter than the data transmission time, further reducing the system response time. Compared to traditional cloud computing models, the response time of the proposed cloud-edge-terminal collaboration system has been compressed from 12.10 to 1.38 s, a decrease of up to 88.58%. The transmission cost is only 572.86 KB, and the total traffic is reduced by 89.86%. To further validate the robustness of the proposed system, we conducted a sensitivity analysis of the network bandwidth, as shown in Fig. 8. As the bandwidth decreases from 1.0 to 0.1 Mbps, the response time of the traditional solutions increases exponentially, with peaks exceeding 57 s. In contrast, the proposed CET-CAI agent can maintain a stable response time of approximately 2 s even under such a limited bandwidth, demonstrating strong robustness. Unlike the cloud-centric paradigm, where external transmission accounts for over 90% of the time cost, the proposed architecture successfully shifts the computing load to the local edge. This decoupling ensures that the real-time performance of the system is determined by controllable local resources rather than by unpredictable external network quality.

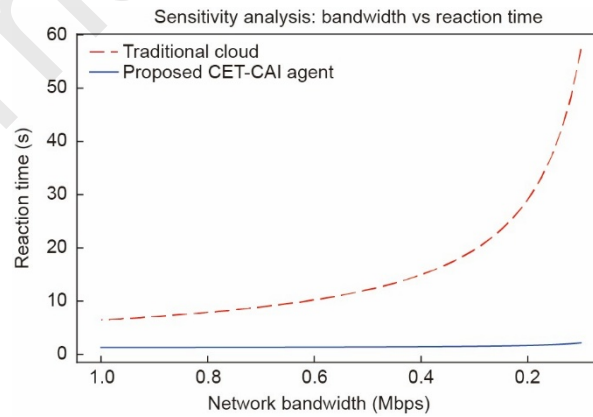


Fig. 8. Sensitivity analysis on network bandwidth.

4. Conclusion

This paper presents a novel CET-CAI agent for CFDD in complex mechanical systems. By leveraging an LLM as a cloud-based coordinator and deploying a DiCoT at the edge system, the proposed framework enables dynamic task recognition, subsystem-aware feature extraction, and human-AI interaction. This framework provides a feasible collaborative model between LLM and small-scale models, endowing traditional diagnostic architectures with higher flexibility and cognitive reasoning capabilities. Experimental evaluations demonstrate that the CET-CAI agent not only achieves superior diagnostic accuracy and generalization on unseen compound faults but also significantly reduces communication overhead through feature-level compression and hierarchical processing. Moreover, the proposed CET-CAI agent subverts the traditional diagnostic mode, and its output is not only the probability of faults but also includes cognitive intelligence analysis results, such as batch decision-making and root cause analysis, result reliability analysis, as well as subsequent maintenance recommendations. The orchestration, scheduling, and decision-making in the entire architecture require the use of LLM to understand the tasks of instructions, autonomously generate decision instructions to select and enable models, as well as analyze the diagnostic results of the models to make fuzzy decisions. The proposed architecture effectively bridges the gap between traditional deep models and human-centric industrial intelligence, significantly expanding the performance and functional capabilities of traditional solutions, which provides a highly feasible solution and potential for LLM-based industrial general intelligence.

Future work will explore the integration of uncertainty quantification, lifelong learning, multiple AI agent technologies, and broader industrial applications to further enhance the adaptability and reliability of CET-CAI agents. In addition, we will deploy this solution to practical industrial scenarios for long-term operational testing and exploration, including systematically addressing edge device failures, preventing erroneous or misleading maintenance recommendations, and comprehensively evaluating operator trust and human factors.

References

- [1] Liu Y, Zhou Y, Liu Y, Xu Z, He Y. Intelligent fault diagnosis for CNC through the integration of large language models and domain knowledge graphs. *Engineering* 2025;53:311–22.
- [2] Pu H, Teng S, Xiao D, Xu L, Qin Y, Luo J. Compound fault diagnosis of rotating machine through label correlation modeling via graph convolutional neural network. *IEEE Trans Instrum Meas* 2024;73:1–10.
- [3] He Y, Shen W. An individual generalization framework based on independent samples towards a more reasonable fault diagnosis benchmark. *Comput Ind* 2025;173:104359.
- [4] Li W, Lan H, Chen J, Feng K, Huang R. WavCapsNet: an interpretable intelligent compound fault diagnosis method by backward tracking. *IEEE Trans Instrum Meas* 2023;72:1–11.
- [5] Yang Z, Baraldi P, Zio E. A method for fault detection in multi-component systems based on sparse autoencoder-based deep neural networks. *Reliab Eng Syst Saf* 2022;220:108278.
- [6] Zhu Y, Zi Y, Zhang M, Xu J. LFC-DGNet: a likelihood feature compositional domain generalization network from single-fault to unseen multi-component compound fault diagnosis across machines. *Adv Eng Inform* 2025;64:103037.
- [7] He Y, Shen W. MSRCN: a cross-machine diagnosis method for the CNC spindle motors with compound faults. *Expert Syst Appl* 2023;233:120957.
- [8] Han S, Shao H, Cheng J, Yang X, Cai B. Convformer-NSE: a novel end-to-end gearbox fault diagnosis framework under heavy noise using joint global and local information. *IEEE/ASME Trans Mechatron* 2023;28(1):340–9.
- [9] Liu B, Yan C, Liu Y, Lv M, Huang Y, Wu L. ISEANet: an interpretable subdomain enhanced adaptive network for unsupervised cross-domain fault diagnosis of rolling bearing. *Adv Eng Inform* 2024;62(Pt A):102610.
- [10] Pang B, Liu Q, Xu Z, Sun Z, Hao Z, Song Z. Fault vibration model driven fault-aware domain generalization framework for bearing fault diagnosis. *Adv Eng Inform* 2024;62(Pt A):102620.
- [11] Wu F, Shen T, Bäck T, Chen J, Huang G, Jin Y, et al. Knowledge-empowered, collaborative, and co-evolving AI models: the post-LLM roadmap. *Engineering* 2025;44:87–100.

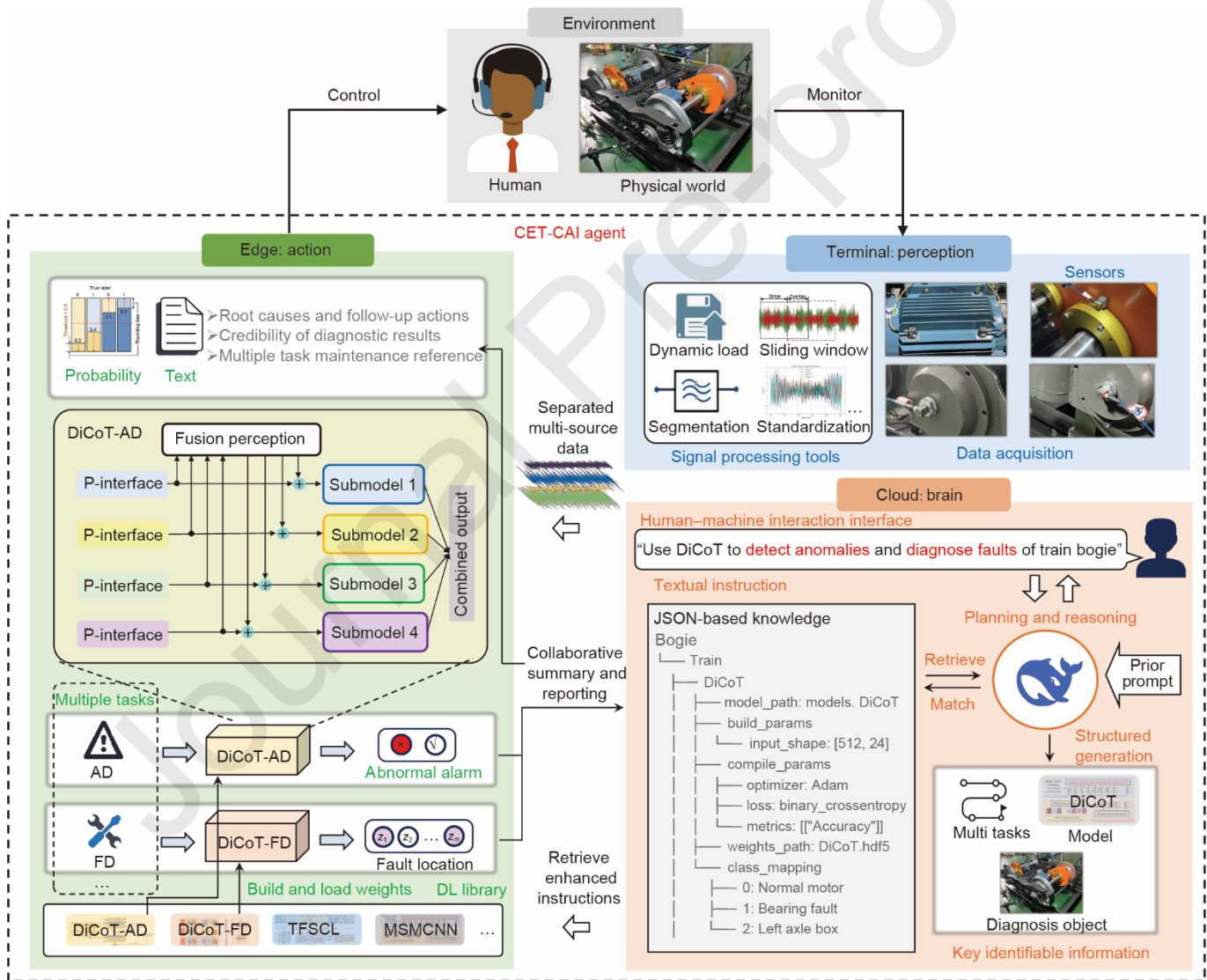
- [12] Jiang P, Sonne C, Li W, You F, You S. Preventing the immense increase in the life-cycle energy and carbon footprints of LLM-powered intelligent chatbots. *Engineering* 2024;40:202–10.
- [13] You Y, Tan K, Jiang Z, Zhang L. Developing a predictive platform for salmonella antimicrobial resistance based on a large language model and quantum computing. *Engineering* 2025;48:174–84.
- [14] Ma Y, Zheng S, Yang Z, Pan H, Hong J. A knowledge-graph enhanced large language model-based fault diagnostic reasoning and maintenance decision support pipeline towards industry 5.0. *Int J Prod Res* 2025;0:1–22.
- [15] Peng H, Liu J, Du J, Gao J, Wang W. BearLLM: a prior knowledge-enhanced bearing health management framework with unified vibration signal representation. *Proc Conf AAAI Artif Intell* 2025;39(19):19866–74.
- [16] Lin L, Zhang S, Fu S, Liu Y. FD-LLM: large language model for fault diagnosis of complex equipment. *Adv Eng Inform* 2025;65(Pt A):103208.
- [17] Wang J, Li T, Yang Y, Chen S, Zhai W. DiagLLM: multimodal reasoning with large language model for explainable bearing fault diagnosis. *Sci China Inf Sci* 2025;68(6):160103.
- [18] Fabre W, Haroun K, Lorrain V, Lepecq M, Sicard G. From near-sensor to in-sensor: a state-of-the-art review of embedded ai vision systems. *Sensors* 2024;24(6):5446.
- [19] Asghari A, Sohrabi MK. Server placement in mobile cloud computing: a comprehensive survey for edge computing, fog computing and cloudlet. *Comput Sci Rev* 2024;51:100616.
- [20] Huaranga-Junco E, González-Gerpe S, Castillo-Cara M, Cimmino A, García-Castro R. From cloud and fog computing to federated-fog computing: a comparative analysis of computational resources in real-time iot applications based on semantic interoperability. *Future Gener Comput Syst* 2024;159:134–50.
- [21] He Y, Shen W. CFSPT: a lightweight cross-machine model for compound fault diagnosis of machine-level motors. *Inf Fusion* 2024;111:102490.
- [22] Qin Y, Wang Y, Li Zhaojun S, Wang B, Ding A, Wang C, et al. An in-depth tutorial on BJTU-RAO bogie datasets for fault diagnosis. *IEEE Access* 2025;13:60879–88.
- [23] Ding A, Qin Y, Wang B, Guo L, Jia L, Cheng X. Evolvable graph neural network for system-level incremental fault diagnosis of train transmission systems. *Mech Syst Signal Process* 2024;210:111175.
- [24] Zhang W, Peng G, Li C, Chen Y, Zhang Z. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors* 2017;17(2):425.
- [25] He Y, Zhao C, Shen W. Cross-domain compound fault diagnosis of machine-level motors via time–frequency self-contrastive learning. *IEEE Trans Ind Inf* 2024;20(7):9692–701.
- [26] He Y, Chen J, Zhou X, Huang S. *In-situ* fault diagnosis for the harmonic reducer of industrial robots via multi-scale mixed convolutional neural networks. *J Manuf Syst* 2023;66:233–47.
- [27] Zhou H, Yang J, Zhu Q, Chen J. Cross-speed spindle motor bearings fault diagnosis combined with multi-space variable scale adaptive filter and feedforward hybrid strategy. *ISA Trans* 2025;157:368–80.
- [28] Li T, Zhou Z, Li S, Sun C, Yan R, Chen X. The emerging graph neural networks for intelligent fault diagnostics and prognostics: a guideline and a benchmark study. *Mech Syst Signal Process* 2022;168:108653.
- [29] Zhang Y, Zhang S, Zhu Y, Ke W. Cross-domain bearing fault diagnosis using dual-path convolutional neural networks and multi-parallel graph convolutional networks. *ISA Trans* 2024;152:129–42.
- [30] Tao L, Liu H, Ning G, Cao W, Huang B, Lu C. LLM-based framework for bearing fault diagnosis. *Mech Syst Signal Process* 2025;224:112127.

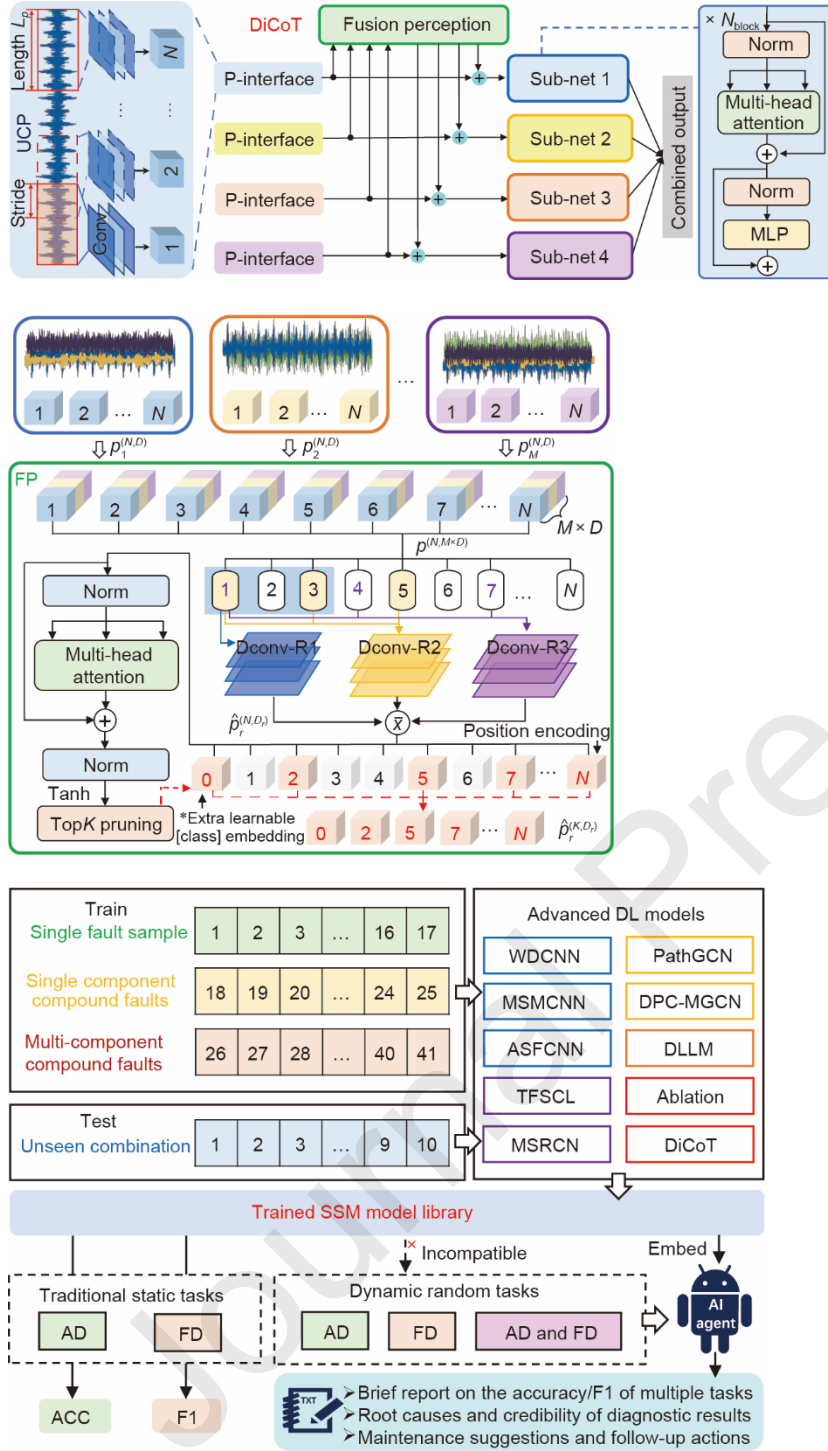
Declaration of interests

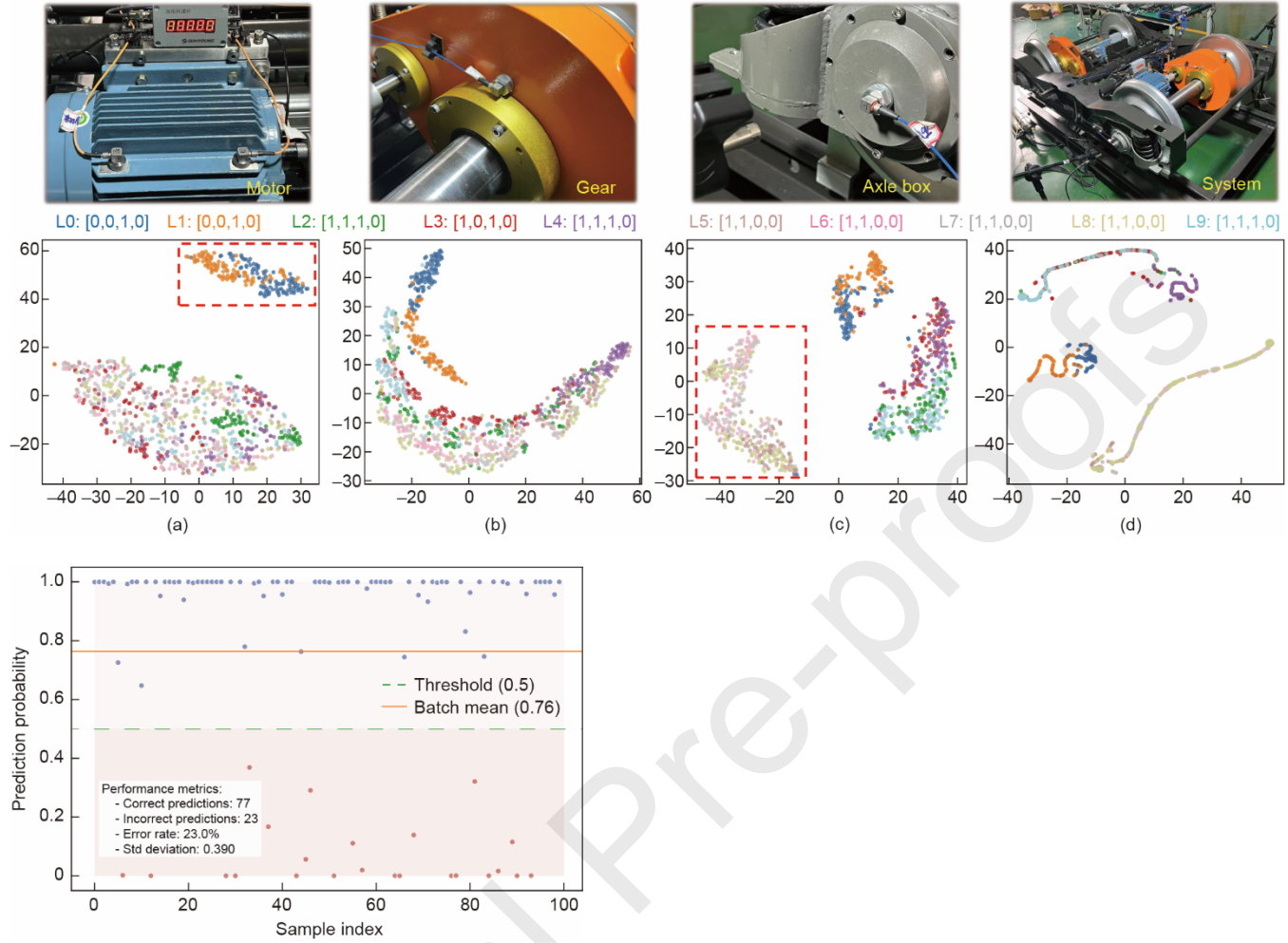
☒ The authors declare that they have no known competing financial interests or personal

relationships that could have appeared to influence the work reported in this paper.

□ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:







CET-CAI agent

Style: Tech_blue

Model selection

Select LLM...

Now: DeepSeek-R1-Dis...

Load LLM

Operation

History

Help

Clear chat

Chat

I. Diagnosis Results and Root Cause Analysis

- Motor
 - Diagnostic Results: 99% of samples ≥ 0.999 (only 2 slightly lower), indicating persistent high-probability faults.
 - Root Cause: Long-term overload operation or insulation aging.
- Gearbox
 - Diagnostic Results: Extreme divergence (12 samples ≤ 0.01 , 26 samples ≥ 0.99 , 62 samples between 0.7-1.0).
 - Root Cause: Lubricant contamination or cumulative gear assembly tolerances.
- Left Axle
 - Diagnostic Results: 100% of samples ≈ 1 (only one at 0.999), confirming deterministic faults.
 - Root Cause: Fatigue stress concentration or impact load damage.
- Right Axle
 - Diagnostic Results: 100% of samples $< 10^{-4}$ (except one at 5.77×10^{-4}).

II. Reliability Assessment of Diagnostic Results

Subsystem	Reliability	Key Evidence
Motor	★★★★☆	Highly consistent results (>99.9%)
Gearbox	★★★☆☆	High result volatility (low-value samples)

Task input

System data

Please use DiCoT model to detect and diagnose

Select

Enter

I. Diagnosis results and root cause analysis

- Motor
 - Diagnostic results
 - Root cause
- Gearbox
 - Diagnostic results
 - Root cause
- Left axle
 - Diagnostic results
 - Root cause
- Right axle
 - Diagnostic results
 - Root cause

II. Reliability assessment of diagnostic results

Subsystem	Reliability	Key evidence
Motor	★★★★☆	Highly consistent results
Gearbox	★★★☆☆	High result volatility
Left axle	★★★★★	100% fault confirmation
Right axle	★★★★★	100% normal status

III. Optimization recommendations

- Threshold adjustment
- Structural optimization
- Data enhancement
- Uncertainty quantification

IV. Maintenance plan

- Priority: left axle > motor > gearbox > right axle
- Emergency actions
 - Monitoring enhancements
 - Predictive measures
 - Random check

