Eksploatacja i Niezawodnosc - Maintenance and Reliability
Volume 29 (2027), Issue 2
https://ein.org.pl
The Polish Maintenance Society

An interpretable fault diagnosis framework via dual-Mamba driven small model and fine-tuned LLM collaboration

Yi Ren[1,2], JiaWen Zhuang[1,*], FangJun Luan[1], Shuai Yuan[1]

[1] School of Computer Science and Engineering, Shenyang Jianzhu University, Shenyang, Liaoning, China
[2] Liaoning Provincial Key Laboratory of Big Data Management and Analysis of Urban Construction, Shenyang, Liaoning, China
* Corresponding Author: zzzwen31@163.com

Received: 23 April 2026
Revised: 16 June 2026
Accepted: 14 August 2026
Online: 7 September 2026

Abstract

With the advent of Industry 4.0/5.0, bearing fault diagnosis is crucial for intelligent manufacturing. This paper proposes DM²-Fuse, an interpretable multimodal framework based on large–small model collaboration. It integrates lightweight Mamba variants—MambaNAS (for raw vibration signals) and MambaDES (for time–frequency representations)—into MambaBearing to extract multimodal features. These are input to an LLM for diagnostic decisions and structured reports. Experiments on CWRU, JNU, and SEU show that the multimodal MambaBearing model achieves 99.41%–100% accuracy with only 146.4K parameters. MambaNAS achieves 40,009 samples/s encoder throughput (1.17× WDCNN), while MambaDES is 11.7× faster than ResNet18. Ablations confirm NAS and separable convolutions contribute +4.8% and +1.3% accuracy, respectively. An adaptive routing mechanism reduces average LLM latency by 86%, and extensive noise and cross-domain experiments validate practical robustness. The framework reduces computation, maintains high accuracy, and enhances interpretability for industrial deployment.

Keywords: b*earing fault diagnosis, dual mamba models, multimodal fusion, large–small model collaboration, interpretability*

Article citation:
Ren Y, Zhuang JW, Luan FJ, Yuan S, An interpretable fault diagnosis framework via dual-Mamba driven small model and fine-tuned LLM collaboration, Eksploatacja i Niezawodnosc – Maintenance and Reliability 2027: 29(2) http://doi.org/10.17531/ein/231129

Highlights

- A novel framework for bearing fault diagnosis via large-small model collaboration.
- Lightweight MambaNAS and MambaDES leverage NAS and depthwise convolutions for efficiency.
- Cross-attention fusion effectively integrates time-series and time-frequency information.
- Fine-tuned LLM generates interpretable diagnostic reports for industrial decision support.
- Adaptive routing mechanism and robustness validation ensure practical deployability.

1. Introduction

Bearings, as key components in rotating machinery, heavily influence the safety and stability of equipment. Therefore, fast and accurate bearing fault diagnosis is of great practical and engineering significance [1,2]. With the rapid development of intelligent manufacturing (Industry 4.0/5.0), the use of deep learning to realize automatic bearing fault identification and health assessment has been heavily studied. At the same time, the industry also cares about the lightweight of models and their interpretability [3,4], to let the model not only have high accuracy, but also be lightweight enough to deploy in resource-constrained embedded systems [5], and provide engineers with reliable decision support.

Tremendous improvements in intelligent fault diagnosis and prognosis are being made nowadays. We started to move from "handcrafted feature extraction + machine learning" to end-to-end deep learning, automatic feature extraction, and representation. Broad survey shows that Convolutional Neural Networks (CNNs), Long Short-Term Memory Networks (LSTMs), Temporal Convolutional Networks (TCNs), and Transformer-based methods have achieved certain success in noise, different operational conditions, and various diagnostic tasks, including classification, detection, and remaining useful life prediction [6]. However, as shown in Figure 1, these methods are still facing some challenges that affect their

potential applicable application in complex industrial environments.

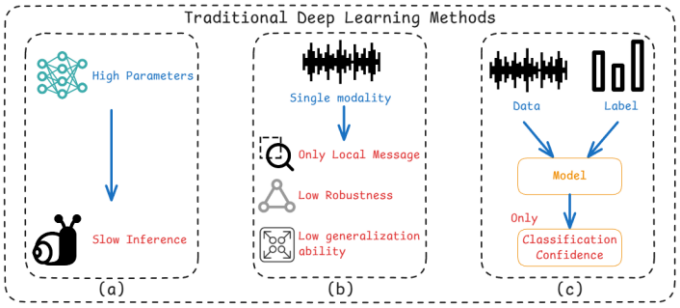

Figure 1. Challenges of traditional deep learning-based bearing fault diagnosis methods: (a) high computational cost, (b) single-modality limitation, and (c) insufficient interpretability.

A major snag is the insatiable thirst for computational and memory resources of modern deep learning. Sun et al. [7] suggest replacing attention modules with convolutional layers, replacing attentions with conv layers using depthwise separable convolutions, arriving at a model called LiteFormer. It is a lot lighter than previous designs, but still has to be manually designed, is not truly interpretable, and would be even better if it had fewer parameters. In industrial settings, these models remain difficult to deploy on edge devices with limited computational resources. Reducing the overhead of computation and storage while still maintaining diagnostic skill is an open problem in itself. In addition, most of these models rely on heavy manual tuning [8], which artificially complicates and extends the optimization.

Another important issue is the lack of modeling for long-range temporal dependencies: although architectures like LSTMs and TCNs seem to capture short-to-medium temporal correlations well enough, they lose all their utility as soon as they confront long sequences due to vanishing or exploding gradients [9]. A1D LSTM-regulated residual network was proposed by Mohammad-Alikhani et al. [10] for motor-driven fault diagnosis, though it is not quite straightforward enough to reliably operate under variable operating conditions, nor does it scale particularly cross-domain readily to new domains. Recently, State Space Models (SSMs), particularly Mamba [11], have emerged as a promising alternative. Mamba achieves linear time complexity "O(L)" , in contrast to the quadratic complexity "O(" "L" ^"2" ")" of Transformers, while effectively modeling long-range dependencies through selective state updates. However, the direct application of Mamba to bearing fault diagnosis, without further architectural optimization and multimodal integration, remains insufficient for industrial deployment. This limitation hinders its ability to model long-sequence faults, especially under variable load and speed conditions in real-world bearings.

Furthermore, deep learning models are often criticized as "black boxes," making it difficult for engineers to interpret their diagnostic reasoning. Interpretability is particularly crucial in fault diagnosis [12]. Most existing models provide only fault categories and confidence levels, offering no insights into the underlying causes or contributing factors, which limits their usefulness in maintenance decision-making. Recent studies have started exploring the integration of large language models (LLMs) to improve interpretability [13]. For example, Tao et al. [14] introduced an LLM-based fault diagnosis framework using feature textualization and parameter-efficient fine-tuning to enhance the interpretability of model outputs. However, this research direction is still in its early stages.

Lastly, methods that only use single-modality data (e.g., time-domain vibration) rather than multimodal data with complementary information. In real-world settings, vibration data from single modalities might be too noisy or insufficient in terms of capturing signals of incipient fault [15]. Yi et al. [16] devise VibrMamba, a lightweight fault diagnosis model based on SSMs, with decent precision and lower overhead cost, but it's only on signal domain data and is not validated from many angles. In search of strong robustness and precision, more researchers begin to utilize multimodal information(e.g., time sequence, acoustic signal, time-frequency representation) [17,18]. Heterogeneous modalities'integration with learning efficiency, adaptivity, and complementarity of features remains a significant challenge in multimodal learning.

To address these issues, this paper introduces an interpretable fault diagnosis framework called DM^2-Fuse, featuring a Dual-Mamba-driven small model and a fine-tuned LLM in a collaborative large-small mechanism. This framework prioritizes lightweight design and interpretability, employing a pipeline collaboration strategy [19] where the small model preprocesses and extracts multimodal features, and the fine-tuned LLM performs interpretable diagnostic reasoning. This design combines performance, efficiency, and explainability through the end-to-end cooperation of specialized small models

and general-purpose large models.

The main contributions of this work are as follows:

2. Large-small collaborative mechanism: A pipeline collaboration strategy is introduced between the lightweight small model and the fine-tuned LLM, combining their strengths—efficient feature extraction and interpretable decision-making. An adaptive routing mechanism further balances latency and interpretability by dynamically invoking the LLM only for low-confidence or complex cases.
3. Lightweight small-model design: A NAS-optimized Mamba temporal network and a depthwise separable convolution-enhanced Mamba vision module are proposed to extract dynamic and complementary features from vibration signals and CWT-based time-frequency maps. This dual-Mamba small model reduces parameters and computational cost while preserving strong long-term dependency modeling.
4. Multimodal feature fusion: A cross-attention mechanism integrates signal and image features, enabling the model to capture potential correlations and complementary information across modalities.
5. Interpretable diagnostic reasoning: The fused features, along with statistical prompts, are embedded into a fine-tuned LLM, enabling the framework to provide both diagnostic results and natural language explanations, thereby improving transparency and trustworthiness.

2. A review of deep learning and multimodal large-model–driven bearing fault diagnosis

Deep learning methods, particularly CNNs, LSTMs, Recurrent Neural Networks (RNNs), Transformer-based models, and their various variants, have been widely applied in bearing fault diagnosis. Xu et al. [20] proposed a fault diagnosis model based on CNN-LSTM-Attention, which effectively handles common faults such as short circuits, permanent magnet demagnetization, and rotor eccentricity. This model leverages the feature extraction capability of CNNs, the sequential data processing power of LSTMs, and the weighted feature selection provided by attention mechanisms. You et al. [21] designed a hybrid model combining QNNs and Bi-LSTM, which they proved to be a fast and efficient model used for diagnosing rolling bearing faults. Du et al. [22] proposed a joint 1D-2D convolutional neural network (JCNN) for rotating machinery fault diagnosis. This approach allows for feature vectors to be adaptively extracted from vibration signals by one-dimensional convolution, and the resulting feature vectors are transformed into two-dimensional images as inputs to a two-dimensional CNN, which demonstrated good performance on several datasets. Zhang et al. [23] proposed a fault diagnosis model based on DeepBottleneck-1DCNN. The model uses an efficient bottleneck block structure to process one-dimensional vibration signals. Ma et al. [24] introduced a rolling bearing fault diagnosis model based on VMD-CNN-SLSTM, which effectively integrates variational mode decomposition, time-frequency analysis, and a deep learning network to achieve superior performance. Despite their effectiveness, these methods are limited to single-modality vibration inputs and hand-crafted architectures, hindering cross-modal utilization and increasing computational cost. A comprehensive review by Zhu et al. [25] systematically categorized deep learning-based intelligent fault diagnosis methods for rotating machinery into five types—deep belief networks, autoencoders, convolutional neural networks, recurrent neural networks, and generative adversarial networks—and highlighted small-sample imbalance and transfer fault diagnosis as the primary challenges facing the field. Our DM²-Fuse framework addresses the transfer challenge through cross-modal fusion of time-series and time-frequency representations, which enhances robustness to varying operating conditions, and tackles the small-sample issue via NAS-optimized lightweight architectures that reduce overfitting risks.

Motivated by the need for comprehensive fault detection, the focus of research studies then shifted towards the development of multimodal approaches that fuse data from various sources of information to gain richer descriptions of the data, making them more robust and accurate for fault diagnosis. With the introduction of the cross-attention [26] significant progress has been made in methods that merge by fusing multimodal data, Peng et al. [27] presented a knowledge graph (MKG) construction based on multimodal data including time series vibration signals, spectrograms, and dataset description texts, and design a fault diagnosis using a relationship cascade graph attention network to obtain graphs representing each

fault's difficulty level. Huang et al. [28] propose multimodal time-series preprocessing synergistic with feature extraction methods based on acceleration, frequency, and time-domain features to enhance model robustness. Zhou et al. [29] propose a dual-channel information fusion for machine wear fault diagnosis based on the YOLOv8 architecture, tackling challenges including multi-scale wear defect detection, heterogeneous wear patterns, and small target localization in industry. These multimodal fusion approaches enhance robustness by integrating heterogeneous information, but still depend on conventional CNN-/attention-based backbones and thus lack architectural optimization, deployment awareness, and interpretability.

Following the advent of the Transformer architecture [30], large-language models (LLMs) such as GPT-4 [31] and Qwen-3 [32] exhibit sophisticated generalization capabilities over multimodal tasks as well as core natural language processing (NLP) abilities. These models could be prompted using suitable text prompts or fine-tuned over similar tasks of interest, such that they could be deployed directly over downstream applications. Zhou et al. [33] proposed the T2MFDF framework, which addresses time-series vibration signals and fuses these with textual tone descriptions using related LLMs based on BERT architectures [34] to learn about the semantic associations between fault categories. Jin et al. [35] propose the Time-LLM reprogramming framework that attempts to transfer capabilities from pre-trained LLMs into time-series prediction tasks. Wang et al. [36] introduced a method to predict the remaining useful life (RUL) based on fine-tuning the NLP knowledge from GPT-2 for prognostics and health management (PHM) applications. They even propose fine-tuning the knowledge within GPT-2-based LLMs by taking inferences from textual dimensions instead of switching in neural networks. Peng et al. [37] propose BearLLM, the first large multimodal framework designed for bearing health management, which aligns vibration signals with textual tones to make models more interpretable. Liu et al. [38] propose AeroGPT, introducing the first large audio model for bearing fault diagnosis in jet engines. Chen et al. [39] propose scaling the LLM paradigm for diagnosis with the new FaultGPT approach, where they represent vibration signals in the form of video images and use large vision-language models (LVLMs) to generate human-readable diagnostic reports. And lay out the whole paradigm of Fault Diagnosis Question Answering (FDQA). Additionally, ImageBind [40] was utilized to fuse textual descriptions with time-frequency images, bringing them closer together in the semantic space. Wang et al. [41] proposed DiagLLM, utilizing envelope spectrogram-image-expert knowledge dual-modal data for enhanced generalization and interpretability. Despite these advances, critical limitations remain. First, most methods (e.g., T2MFDF, BearLLM, FaultGPT) rely on heavyweight models (BERT, GPT-2, LVLMs) with substantial computational overhead unsuitable for edge deployment. Second, while incorporating multimodal data, frameworks like BearLLM and DiagLLM use generic CNNs without architectural optimization via NAS or depthwise separable convolutions. Third, recent Mamba-based models (e.g., VibrMamba) focus on single-modality processing, neglecting time-frequency complementarity and LLM-based interpretability. Finally, methods achieving interpretability through generative models (AeroGPT, FaultGPT) lack small-large model collaboration, relying entirely on large models with excessive inference cost. In terms of deployment architecture, Huang et al. [42] proposed a deep residual networks-based intelligent fault diagnosis method for planetary gearboxes in cloud environments, leveraging cloud computing to overcome the insufficient computational capacity of local edge devices. While this cloud-centric paradigm demonstrates the feasibility of offloading heavy computation to remote servers, our DM²-Fuse framework advances this concept further by introducing a large-small model collaboration mechanism with adaptive routing. Rather than offloading the entire diagnostic pipeline to the cloud, we deploy a lightweight MambaBearing encoder (146.4K parameters) at the edge for real-time feature extraction and preliminary fault detection, and only invoke the cloud-based LLM when detailed interpretative reporting is required for ambiguous or compound fault cases. This strategy achieves a more favorable trade-off between latency, bandwidth consumption, and diagnostic depth.

A related paradigm for integrating LLMs with fault diagnosis was recently demonstrated by the SHAP-LLM framework proposed by Zhao et al. [43] This framework delegates core diagnostic tasks to a PI-Dual-STGCN model and leverages SHAP-based XAI techniques to generate quantitative

explanations of model decisions, while restricting the LLM's role to reasoning over pre-quantified, fact-based textual information (e.g., model performance metrics and SHAP reliability scores). This design avoids the uncertainty of directly using large models for fault diagnosis and reduces hallucination risks by grounding LLM outputs in verifiable numerical evidence. Our DM²-Fuse framework shares the same underlying philosophy—employing a lightweight small model for feature extraction and fault classification, and invoking the LLM solely for interpretable report generation. However, DM²-Fuse differs from SHAP-LLM in three key aspects: (i) we employ a dual-Mamba architecture optimized via NAS and depthwise separable convolutions for efficient multimodal feature extraction, in contrast to the graph-based STGCN backbone; (ii) we adopt LoRA-based fine-tuning rather than RAG to adapt the LLM, enabling more integrated multimodal reasoning without external knowledge base retrieval; and (iii) our framework incorporates an adaptive routing mechanism that dynamically selects between template-based and LLM-based report generation to balance latency and interpretability. These distinctions, summarized in Table 1, highlight how DM²-Fuse extends the small-large collaborative paradigm with lightweight, multimodal, and adaptively efficient design choices.

Table 1. Comparison of recent intelligent fault diagnosis methods.

Method	Multi-modal	Light-weight	NAS-Optimized	Interpretable	Small–Large Collab.
T2MFDF	√	×	×	×	×
Time-LLM	×	×	×	×	×
BearLLM	√	×	×	√	×
AeroGPT	×	×	×	√	×
FaultGPT	√	×	×	√	×
DiagLLM	√	×	×	√	×
VibrMamba	×	√	×	×	×
SHAP-LLM	×	×	×	√	√
DM²-Fuse (Ours)	√	√	√	√	√

As summarized in Table 1, our DM^2-Fuse framework addresses these gaps through: (i) a NAS-optimized MambaNAS network for temporal signal processing and a MambaDES network enhanced with both NAS and depthwise separable convolutions for time-frequency image analysis, achieving lightweight multimodal feature extraction; (ii) a MambaBearing multimodal model that employs cross-attention fusion to integrate complementary signal-image information; (iii) a pipeline collaboration strategy where the lightweight MambaBearing small model pre-processes and extracts multimodal features, and a fine-tuned LLM performs interpretable diagnostic reasoning; and (iv) interpretable outputs that combine diagnostic classifications with natural language explanations via statistical prompt-guided LLM fine-tuning, thereby balancing performance, efficiency, and explainability.

3. Proposed DM^2-fuse framework

In this section, we present a comprehensive description of the proposed DM^2-Fuse architecture, which achieves efficient bearing fault diagnosis through multimodal data fusion, NAS, depthwise separable convolutions, and fine-tuned LLMs for interpretable outputs. We first introduce the overall framework of the model, followed by detailed explanations of the NAS optimization process, the construction of MambaBearing (comprising MambaNAS and MambaDES), cross-modal fusion mechanisms, and the methodology for constructing fine-tuning datasets and generating interpretable outputs from LLMs.

3.1. Overall framework

The proposed DM²-Fuse framework establishes an efficient and

interpretable bearing fault diagnosis system through a three-stage pipeline, as illustrated in Figure 2.

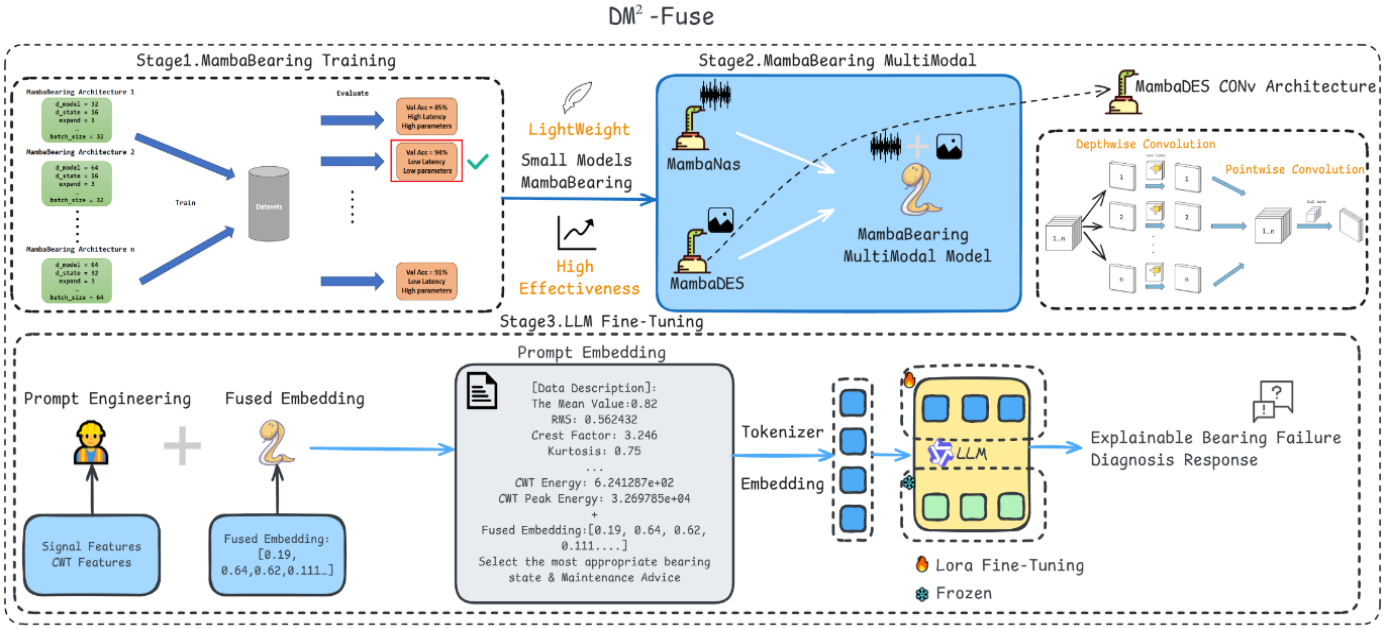

Figure 2. Overall methodology framework of the proposed DM²-Fuse.

In Stage 1, we leverage NAS to derive a lightweight temporal backbone, termed MambaNAS, which is optimized for modeling raw bearing vibration signals with minimal computational overhead.

In Stage 2, we adapt MambaNAS by replacing its standard convolutional layers with depthwise separable convolutions, yielding a specialized image encoder MambaDES—that achieves lower parameter count while enhancing feature representation of continuous wavelet transform (CWT) time-frequency spectrograms. The resulting MambaBearing model integrates MambaNAS (for temporal signal modeling) and MambaDES (for spectral image understanding) into a unified multimodal architecture.

In Stage 3, cross-attention mechanisms fuse the complementary features from both modalities, and the fused representation is injected into an LLM via prompt-based fine-tuning. Enabling accurate fault classification together with natural-language interpretable reasoning.

Formally, the DM²-Fuse architecture can be expressed as follows:

$$DM^2 - Fuse = LLM(\ MambaBearing(MambaNAS(X_s),\ MambaDES(X_i)), Stats) \quad (1)$$

where $X_s \in \mathrm{R}^{B\times L\times 1}$ represents the bearing vibration signal, $X_i \in \mathrm{R}^{B\times 3\times H\times W}$ denotes the corresponding CWT time-frequency spectrogram, and $Stats \in \mathrm{R}^{d_{stat}}$ contains statistical features extracted from both modalities.

3.2. Efficient Mamba architecture search

To optimize the temporal feature extraction capability of the Mamba-based module, we employ NAS to automatically select the most suitable network structure. NAS explores different hyperparameter combinations (e.g. convolutional layer depth, kernel size, hidden layer dimensions) to identify the optimal architecture.

Our NAS employs a multi-objective optimization strategy that balances three key metrics: accuracy, parameters, and inference speed. As shown in Algorithm 1,we maintain a Pareto front of non-dominated solutions and select the final architecture using normalized weighted scoring. This approach ensures that MambaNAS achieves not only state-of-the-art accuracy but also practical efficiency suitable for real-time bearing fault diagnosis applications. The trade-off weights (α,β,γ) were empirically set to 0.6, 0.2, and 0.2, respectively, prioritizing accuracy while maintaining reasonable model size and inference speed for edge deployment.

Through this search process, we automatically select the most appropriate network architecture, determining the optimal parameters for both the Mamba layersand MLP layers in MambaNAS, as well as the Mamba layers in MambaDES. The resulting architecture demonstrates robust performance across multiple datasets.

Algorithm 1

Algorithm 1 Multi-Objective NAS for Mamba Architecture

Input: Training data $\mathcal{D}$, Trials T, Weights α, β, γ
Output: Optimal architecture θ_{opt}

for *trial* $\leftarrow 1$ **to** T **do**
  /* Search space sampling */
  $d_{\text{model}} \leftarrow \texttt{suggest}([32, 64, 128])$
  $d_{\text{state}} \leftarrow \texttt{suggest}([8, 16, 32])$
  $d_{\text{conv}} \leftarrow \texttt{suggest}([2, 4, 6])$
  $num\_layers \leftarrow \texttt{suggest}([4, 6, 8]) \ldots$
  /* Build and train model */
  model $\leftarrow \texttt{Mamba}(d_{\text{model}}, d_{\text{state}}, d_{\text{conv}}, num\_layers)$
  train$(model, \mathcal{D})$
  /* Multi-objective evaluation */
  acc $\leftarrow \texttt{evaluate}(model)$
  params $\leftarrow \texttt{count\_parameters}(model)$
  time $\leftarrow \texttt{measure\_inference\_time}(model)$
  /* Weighted scoring */
  score $\leftarrow \alpha \cdot acc + \beta \cdot \frac{1}{\log(params)} + \gamma \cdot \frac{1}{\log(time)}$
  report$(score, trial)$
return $\theta_{\text{opt}} \leftarrow$ best_configuration

3.3. MambaNAS and MambaDES

The MambaDES network processes two-dimensional CWT time-frequency representations to extract frequency-domain and time-frequency-domain features. The MambaNAS and MambaDES networks structure is shown in Figure 3. The main differences from MambaNAS are: input is images rather than sequences; depthwise separable convolutions are employed for spatial downsampling and feature extraction.

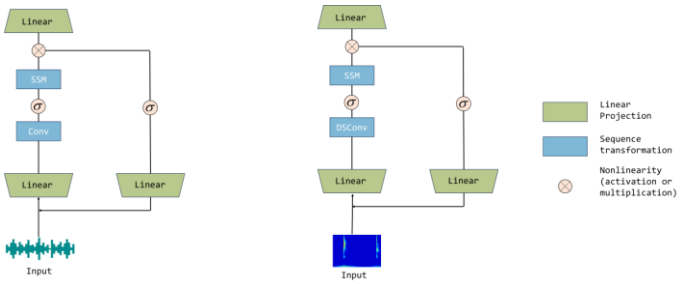

Figure 3. MambaNAS and MambaDES architecture.

1) Depthwise Separable Convolution Principle:

Depthwise separable convolution (DWSConv) factorizes a standard convolution into a depthwise convolution followed by a pointwise convolution. The depthwise step applies a spatial filter of size $K \times K$ independently to each of the C_{in} input channels, requiring $C_{in}K^2$ parameters. The subsequent pointwise step uses a 1×1 convolution to project the features into C_{out} output channels, with $C_{in}C_{out}$ parameters. Thus, the total parameter count is as follows:

$$N_{DWS} = C_{in}K^2 + C_{in}C_{out} \tag{2}$$

Compared to $N_{std} = C_{in}C_{out}K^2$ for standard convolution. The parameter compression ratio is as follows:

$$\frac{N_{DWS}}{N_{std}} = \frac{1}{C_{out}} + \frac{1}{K^2} \tag{3}$$

For typical values $C_{out}=64$ *and* $K=3$, this ratio is approximately 12.7%, corresponding to an ~8× reduction in parameters.

2) SSM:

The Selective State Space Model (S6) is particularly well-suited for long-sequence vibration analysis ($L \geq 1024$) due to its linear time complexity $O(L)$—a significant advantage over the Transformer's $O(L^2)$ self-attention. By discretizing continuous-time dynamics, S6 effectively models long-range fault dependencies across hundreds of time steps without gradient vanishing, a common issue in RNNs/LSTMs. Moreover, its input-dependent selectivity replaces static system parameters with context-aware ones, enabling adaptive focus on transient impacts such as bearing faults. Finally, the state evolution naturally supports multi-scale feature extraction, simultaneously capturing high-frequency impulses and low-frequency modulations (e.g. speed-varying envelopes), which enhances robustness under varying operating conditions (see Figure 4).

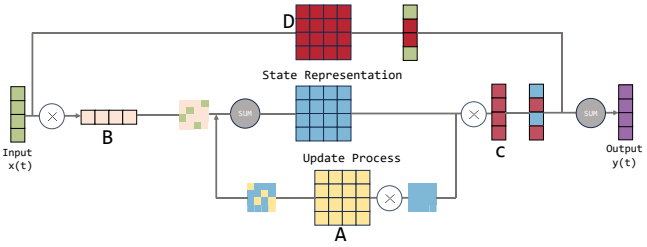

Figure 4. SSM Discretization and State Update Diagram.

The continuous-time form of the SSM is defined as a linear time-invariant system as follows:

$$\begin{aligned} \frac{d\mathbf{h}(t)}{dt} &= \mathbf{A}\mathbf{h}(t) + \mathbf{B}x(t) \\ y(t) &= \mathbf{C}\mathbf{h}(t) + \mathbf{D}x(t) \end{aligned} \tag{4}$$

where $\mathbf{x}(t) \in \mathrm{R}$ is the input vibration signal, $\mathbf{h}(t) \in \mathrm{R}^N$ is the N-dimensional hidden state encoding historical information, and $\mathbf{y}(t) \in \mathrm{R}$ is the output. The system matrices are: $\mathbf{A} \in \mathrm{R}^{N \times N}$ (state transition), $\mathbf{B} \in \mathrm{R}^{N \times 1}$ (input projection), $\mathbf{C} \in \mathrm{R}^{1 \times N}$ (output projection), and $\mathbf{D} \in \mathrm{R}$ (direct feed through, typically zero).

To enable computation on discrete time steps, the continuous-time SSM is discretized using the Zero-Order Hold (ZOH) method. Given a sampling step size Δ (i.e. the sampling

period), and assuming the input remains constant over the interval $[t,t+\Delta]$, the discretized system matrices are derived as follows:

$$\begin{aligned}\overline{\mathbf{A}} &= \exp(\Delta\mathbf{A}),\\ \overline{\mathbf{B}} &= \left(\int_0^{\Delta}\exp(\tau\mathbf{A})\,d\tau\right)\mathbf{B}\\ &=(\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A})-\mathbf{I})\Delta\mathbf{B}\end{aligned} \tag{5}$$

where $\mathbf{I}$ is the identity matrix. The resulting discrete-time state-space equations are as follows:

$$\begin{aligned}\mathbf{h}_t &= \overline{\mathbf{A}}\mathbf{h}_{t-1}+\overline{\mathbf{B}}\mathbf{x}_t,\\ \mathbf{y}_t &= \mathbf{C}\mathbf{h}_t+\mathbf{D}\mathbf{x}_t.\end{aligned} \tag{6}$$

This recursive formulation enables efficient sequence-to-sequence mapping with time complexity $O(LN^2)$. When $N \ll L$—which holds in practice (e.g. N=8--32, L=1024)—the computational cost is significantly lower than that of self-attention, $O(L^2d)$.

3.4. MambaBearing

MambaBearing comprises two components: MambaNAS for processing temporalsignals and MambaDES for processing time-frequency spectrograms. The one-dimensional signal is fed into MambaNAS while simultaneously being converted to a CWT time-frequency spectrogram in memory and input to MambaDES, forming a multimodal model. MambaNAS extracts frequency features to capture harmonic characteristics of faults such as bearing wear and rotor imbalance. MambaDES extracts energy distribution patterns from time-frequency spectrogramsto localize the precise temporal occurrence of faults. Through the fusion of both modalities, the model obtains comprehensive information, synergistically utilizing global frequency-domain and time-frequency information for bearing fault classification.

The overall architecture of MambaBearing is illustrated in Figure 5. We designa cross-attention-based fusion mechanism to integrate the input time series X_s and input image X_i.

Temporal Feature Extraction:

The raw time series $\mathbf{X}_s \in \mathrm{R}^{B\times T}$, where T denotes the input sequence length (e.g., T=1024 in our experiments), is first projected into a d_{model}-dimensional embedding space via a linear layer, followed by a ReLU activation and a second linear transformation. Theresulting representation serves as the direct input to the MambaNAS module:

$$\mathbf{Z}_s=\mathrm{Linear}(\mathrm{ReLU}(\mathrm{Linear}(\mathbf{X}_s))),\quad \mathbf{Z}_s\in\mathrm{R}^{B\times T\times d_{model}} \tag{7}$$

Here, d_{model} is the model dimension of the structured state space (SSM) layer in MambaNAS, governing the capacity of temporal feature encoding. Subsequently, $\mathbf{Z}_s$ is processed by the MambaNAS module, which leverages selective state-space modeling to capture long-range temporal dependencies and extract frequency-aware dynamics. The output temporal features, denoted as $\mathbf{S}_f \in \mathrm{R}^{B\times T\times d_{model}}$, are used as the query ($\mathbf{Q}$) in the cross-attention module, while the image features $\mathbf{I}_f$ provide the corresponding key ($\mathbf{K}$) and value ($\mathbf{V}$).

Image Feature Extraction:

The continuous wavelet transform (CWT) yields a single-channel time-frequencyspectrogram of size 224×224. To extract hierarchical spatial-temporal features, we first process the spectrogram through a depthwise separable convolution (DES) module. The output feature map is then partitioned into non-overlapping 16×16 patches, resulting in $P=(224/16)^2=196$ tokens. Each patch is linearly projected into a d_{model}-dimensional embedding space and flattened into a sequence:

$$\mathbf{X}_i=\mathrm{Flatten}(\mathbf{Z}_{patch}),\quad \mathbf{X}_i\in\mathrm{R}^{B\times 196\times d_{model}} \tag{8}$$

where $\mathbf{Z}_{patch}$ denotes the embedded patch representations. Learnable positional encodings $\mathbf{P}_{emb}\in\mathrm{R}^{1\times 196\times d_{model}}$ are added to retain spatial structure:$\mathbf{X}_i^{pos}=\mathbf{X}_i+\mathbf{P}_{emb}$

This position-aware sequence is subsequently fed into the MambaDES module to model long-range dependencies in the time-frequency domain. The extracted image features, denoted as $\mathbf{I}_f\in\mathrm{R}^{B\times 196\times d_{model}}$, serve as the query ($\mathbf{Q}$) in the cross-attention module, with the temporal features $\mathbf{S}_f$ supplying the key ($\mathbf{K}$) and value ($\mathbf{V}$).

The attention mechanism is defined as:

$$Attention(\mathbf{Q},\mathbf{K},\mathbf{V})=softmax\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\mathbf{V} \tag{9}$$

Each attention head is computed by applying scaled dot-product attention to linearly projected query, key, and value inputs:

$$\mathrm{head}_i=\mathrm{Attention}\left(\mathbf{Q}\mathbf{W}_i^Q,\mathbf{K}\mathbf{W}_i^K,\mathbf{V}\mathbf{W}_i^V\right) \tag{10}$$

where $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, and $\mathbf{W}_i^V$ are learnable projection matrices for the i-th head.

The outputs of all h heads are concatenated and linearly transformed to produce the final multi-head representation:

$$\mathrm{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V})=\mathrm{Concat}(\mathrm{head}_1,\ldots,\mathrm{head}_h)\mathbf{W}^O \tag{11}$$

where $\mathbf{W}^O$ is the output projection matrix.

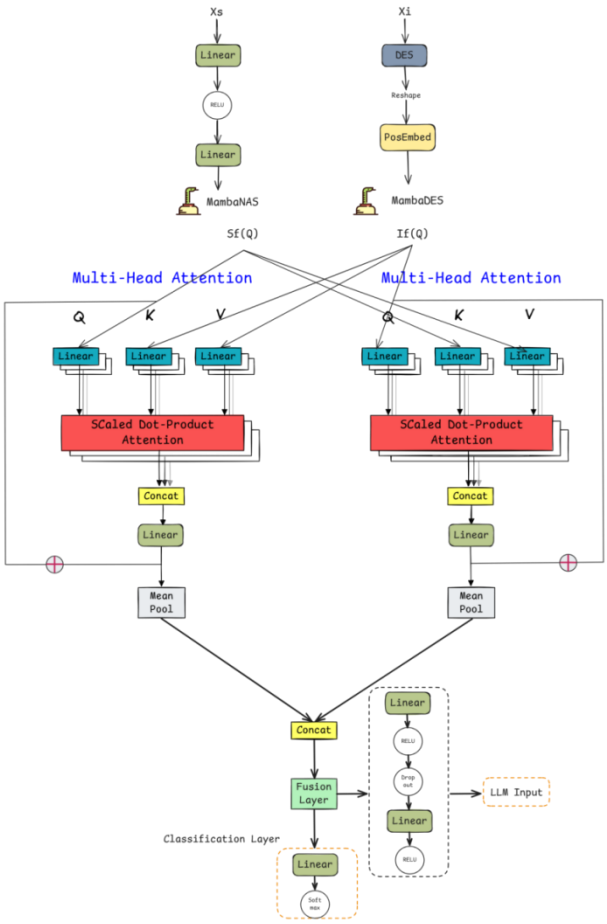

Figure 5. Architecture of MambaBearing.

1) Temporal-to-Image Cross-Attention:

Temporal features attend to relevant information in image features through multi-head cross-attention. Specifically, the temporal features $\mathbf{S}_f \in \mathrm{R}^{B\times T\times d_{model}}$ and image features $\mathbf{I}_f \in \mathrm{R}^{B\times P\times d_{model}}$ are linearly projected into query, key, andvalue representations as follows:

$$\begin{aligned} \mathbf{Q}_S &= \mathbf{S}_f\mathbf{W}^Q,\ \mathbf{Q}_S \in \mathrm{R}^{B\times T\times d_{model}} \\ \mathbf{K}_I &= \mathbf{I}_f\mathbf{W}^K,\ \mathbf{K}_I \in \mathrm{R}^{B\times P\times d_{model}} \\ \mathbf{V}_I &= \mathbf{I}_f\mathbf{W}^V,\ \mathbf{V}_I \in \mathrm{R}^{B\times P\times d_{model}} \end{aligned} \tag{12}$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathrm{R}^{d_{model}\times d_{model}}$ are learnable projection matrices. These projections are then split into h attention heads, each operating in a subspace of dimension $d_k = d_{model}/\mathrm{h}$. Scaled dot-product attention is applied across modalities. The outputs of all attention heads are concatenated and projected to generate the final cross-attention output $\mathbf{O}_{S\leftarrow I} \in \mathrm{R}^{B\times T\times d_{model}}$.

The enhanced temporal representation is subsequently obtained through a residual connection followed by layer normalization:

$$\mathbf{S}_{enhanced} = \mathrm{LayerNorm}(\mathbf{S}_f + \mathbf{O}_{S\leftarrow I}) \tag{13}$$

2) Image-to-Temporal Cross-Attention:

Conversely, image features query relevant temporal dynamics to enrich their semantic content. The image features $\mathbf{I}_f$ serve as queries, while the temporal features $\mathbf{S}_f$ provide keys and values:

$$\begin{aligned} \mathbf{Q}_I &= \mathbf{I}_f\mathbf{W}^Q,\ \mathbf{Q}_I \in \mathrm{R}^{B\times P\times d_{model}} \\ \mathbf{K}_T &= \mathbf{S}_f\mathbf{W}^K,\ \mathbf{K}_T \in \mathrm{R}^{B\times T\times d_{model}} \\ \mathbf{V}_T &= \mathbf{S}_f\mathbf{W}^V,\ \mathbf{V}_T \in \mathrm{R}^{B\times T\times d_{model}} \end{aligned} \tag{14}$$

Following the same multi-head attention mechanism, the cross-modal output$\mathbf{O}_{I\leftarrow T} \in \mathrm{R}^{B\times P\times d_{model}}$ is fused with the original image representation via residual connection and layer normalization:

$$\mathbf{I}_{enhanced} = \mathrm{LayerNorm}(\mathbf{I}_f + \mathbf{O}_{I\leftarrow T}) \tag{15}$$

3) Fused Multimodal Representation:

Let $\mathbf{F}_{fused} = Fusion(\mathbf{S}_{enhanced}, \mathbf{I}_{enhanced}) \in \mathrm{R}^{d_{model}}$ represent the fused multimodal representation, where the Fusion Layer integrates the enhanced temporal andimage features. This embedding is utilized in two ways: (i) as token-level input to a frozen large language model with Low-Rank Adaptation (LoRA) [44] adapters for explainable diagnosis, and (ii) as input to a trainable MLP classifier for direct fault categorization.

3.4.1. Necessity and efficiency of cross-modal fusion

The integration of temporal and time-frequency modalities is essential for capturing complementary diagnostic information: MambaNAS extracts sequential dynamics from raw vibration signals, while MambaDES captures localized energy patterns from CWT spectrograms. To effectively fuse these heterogeneous representations, cross-attention provides a principled mechanism for modeling inter-modal interactions, allowing each modality to attend to relevant features in the other. This fusion is not achievable through simple concatenation, which treats modalities independently without learning their mutual dependencies. Although cross-attention introduces a quadratic complexity term O(T • P) with respect to the compressed sequence lengths (T=64 for temporal features and P=196 for image patches), the absolute computational overhead is minimal. The total operations amount to approximately 1.25×10^4 per forward pass, which constitutes less than 0.3% of the Mamba modules' O($L \cdot d_{model}^2$) complexity ($\approx4.2\times10^6$ operations).Empirically, we compared the cross-attention

fusion with a simple feature concatenation baseline. As reported in Table 7, replacing cross-attention with concatenation reduces per-sample latency by only 0.087 ms (from 0.465 ms to 0.378 ms), while causing substantial accuracy drops of 6.80% on JNU, 5.15% on CWRU, and 14.72% on SEU. The particularly severe degradation on the SEU dataset—which contains compound faults—underscores the importance of cross-attention for modeling complex inter-modal relationships. In summary, the cross-attention mechanism delivers substantial accuracy gains at the cost of an additional 87 microseconds per sample, a favorable trade-off that validates its inclusion in MambaBearing.

3.5. Fine-tuning dataset construction and LLM adaptation

To enable interpretable bearing fault diagnosis, we fine-tune an LLM to adapt it to our downstream diagnostic task. Specifically, we first construct at ask-specific fine-tuning dataset tailored for LLM instruction tuning, as detailed in Algorithm 2. Each training instance comprises the fused multimodal representation $\mathbf{F}_{fused}$, a set of handcrafted statistical features **s** derived from both time-domain vibration signals and time–frequency CWT images, the ground-truth fault label y, and a diagnostic report serving as the target output sequence. Unlike conventional rule-based report generators that map each class label to a fixed textual template, our proposed Generate Report function is a structured semantic synthesis process conditioned on both diagnostic labels and multi-dimensional statistical features extracted from vibration signals. These features (e.g., RMS, kurtosis, spectral peak distributions) provide fine-grained degradation information that cannot be adequately represented by static templates.

Construction of Diagnostic Reports. The target diagnostic reports used for LLMfine-tuning are generated via the function GenerateReport(y, stats), which combines the ground-truth fault label with statistical features (see Table 2). It is important to note that these reports are not produced by a trivial if-else template mapping. Rather, the function incorporates domain knowledge rules to generate sample-dependent variations: for example, different kurtosis and RMS values trigger different severity descriptions (e.g., "elevated RMS suggests moderate wear" vs. "significantly elevated RMS indicates severe spalling"), and frequency-domain peaks are compared against theoretical characteristic frequencies (BPFI, BPFO, BSF) to produce corroborative or contradictory evidence statements. Because the statistical features stats vary across individual samples even within the same fault class due to differences in operating conditions and measurement noise, the generated reports exhibit non-trivial textual diversity. An illustrative example is provided below:

> Example Report (Inner Race Fault):

> Fault Type: Inner Race Fault (Confidence: High).

> Time-Domain Features: RMS = 0.48 g (320% above baseline), crest factor = 5.2,

> kurtosis = 7.8. The elevated crest factor and kurtosis indicate the presence

> of impulsive vibrations typical of localized defects.

> Frequency-Domain Features: A dominant peak is observed at 158.1 Hz, which lies

> within 3% of the theoretical inner race fault characteristic frequency (BPFI

=

> 162.2 Hz at 1750 rpm). Sidebands are spaced at the shaft rotational frequency

> (29.5 Hz), further supporting the diagnosis.

> Conclusion: The time–frequency evidence is consistent with an early-to-moderate

> inner race defect. Continued monitoring is advised, with bearing replacement

> scheduled during the next planned maintenance window.

The LLM is therefore trained to associate subtle variations in the feature vector with corresponding linguistic formulations, rather than merely memorizing a fixed class-to-text mapping.

The fine-tuned LLM learns a mapping from structured numerical-feature space to natural language diagnostic descriptions, enabling compositional generalization across varying fault severities and feature distributions. Therefore, the model does not merely perform template filling, but learns a feature-conditioned semantic reasoning process for fault interpretation and report generation.

A preliminary evaluation using ROUGE-L on the CWRU test set shows that LLM-generated reports exhibit 32% lower similarity to template references than template reports themselves, confirming that the LLM produces diverse rather than memorized outputs.

The statistical feature vector **s** encompasses interpretable descriptors from both time and frequency domains, which are critical for characterizing bearing healthconditions. These features are summarized in Table 2.

Table 2. Handcrafted statistical features for bearing fault diagnosis.

Domain	Feature	Definition
Time Domain	Mean(μ)	$\frac{1}{L}\sum_{t=1}^{L} x_t$
	Standard deviation(σ)	$\sqrt{\frac{1}{L}\sum_{t=1}^{L}(x_t-\mu)^2}$
	RMS	$\sqrt{\frac{1}{L}\sum_{t=1}^{L} x_t^2}$
	Maximum($x_{\max}$)	$\max_t x_t$
	Minimum($x_{\min}$)	$\min_t x_t$
	Peak	$\max(\lvert x_{\max}\rvert, \lvert x_{\min}\rvert)$
	Kurtosis	$\frac{1}{L\sigma^4}\sum_{t=1}^{L}(x_t-\mu)^4$
	Crest factor (CF)	$\frac{\text{Peak}}{\text{RMS}}$
	Signal energy(E)	$\sum_{t=1}^{L} x_t^2$
Frequency Domain	Dominant frequency(f_{dom})	$\arg\max_f \lvert X(f)\rvert$
	Total spectral energy(E_{freq})	$\sum_f \lvert X(f)\rvert^2$
	Band energy($E_{\text{band}}^{(i)}$)	$\sum_{f\in \text{Band}_i} \lvert X(f)\rvert^2$

The final fine-tuning dataset is formulated as follows:

$$D_{ft}=\left\{\left(\mathbf{F}_{fused}^{(i)}, \mathbf{stats}^{(i)}, y^{(i)}, report^{(i)}\right)\right\}_{i=1}^{N} \quad (16)$$

where N is the number of samples, and each $report^{(i)}$ is a structured diagnosticreport in natural language serving as the target sequence for LLM fine-tuning.

During LLM fine-tuning, we employ LoRA to fine-tune the pre-trained LLM. In this approach, all pre-trained weights are frozen to preserve general-purpose knowledge, while task-specific adaptation is achieved by injecting trainable low-rank decomposition matrices into selected transformer layers. Specifically, for apre-trained weight matrix $\mathbf{W}\in \mathrm{R}^{m\times n}$, LoRA approximates the weight update $\Delta\mathbf{W}$ with a low-rank factorization:

$$\mathbf{W'}=\mathbf{W}+\Delta\mathbf{W}=\mathbf{W}+\mathbf{BA} \quad (17)$$

where $\mathbf{B}\in \mathrm{R}^{m\times r}$ and $\mathbf{A}\in \mathrm{R}^{r\times n}$ are trainable parameters, and the rank $r\ll \min(m,n)$ (in our experiments, r=8). During forward propagation, the output is computed as:

$$\mathbf{h}=\mathbf{W'x}=\mathbf{Wx}+\mathbf{BAx} \quad (18)$$

Only **A** and **B** are updated during training, while **W** remains fixed. This reduces the number of trainable parameters from mn to $(m+n)r$, yielding a compression ratio of approximately $\frac{r}{\min(m,n)}$. For instance, when m=n=4096 and r=8 , the trainable parameters are reduced by a factor of 512.

The fine-tuned LLM takes the fused multimodal representation $\mathbf{F}_{fused}$ and statistical features **stats** as contextual input and is trained to autoregressively generate the corresponding natural-language diagnostic *report*. The fault label y is excluded from the input to ensure that predictions are grounded solely in the observed signal characteristics. Training is performed using the causal language modeling (CLM) objective:

$$\mathrm{L}_{LLM}=-\sum_{t=1}^{T}\log P(w_t|\, w_{<t}, \mathbf{F}_{fused}, \mathbf{stats}) \quad (19)$$

where w_t denotes the t-th token of *report*, $w_{<t}$ represents all preceding tokens, and T is the sequence length. This loss maximizes the likelihood of the ground-truth report given the

multimodal context, enabling the LLM to produce accurate, interpretable, and physically consistent diagnoses.

3.5.1. Complementary roles of template and LLM

It is important to clarify that the rule-based template generator in our framework is not a weak baseline—it accurately conveys the fault type and incorporates domain-specific statistical evidence. For routine diagnoses with high classifier confidence, the template-generated report is both sufficient and computationally efficient, which motivates its role as the fast-path option in our adaptive routing mechanism. However, the fine-tuned LLM provides two distinct advantages that justify its inclusion in a collaborative framework:

1. Robustness to Edge Cases: When statistical features deviate significantly from typical fault signatures (e.g., due to novel operating conditions or sensor noise), a rigid template may produce reports that appear overconfident or self-contradictory. The LLM, exposed to diverse linguistic patterns during fine-tuning, can generate appropriately hedged statements.

2. Compositional Flexibility: For compound or multi-location faults, the combinatorial explosion of possible scenarios makes exhaustive template design impractical. The LLM learns to compose coherent explanations for such cases from a limited number of training examples. In summary, the template and LLM serve complementary rather than competitive roles. The template provides a reliable, low-latency fallback for routine diagnostics, while the LLM is invoked adaptively for complex or ambiguous cases—a design choice that balances efficiency with interpretability.

3.6. Adaptive routing for efficient inference

Although the fine-tuned LLM provides enhanced interpretability and compositional flexibility, its inference latency (approximately 1 second per report on typical hardware) is unnecessary for routine diagnostic cases where the classifier confidence is high and the fault type is unambiguous. To balance interpretability with computational efficiency, we introduce an adaptive routing mechanism that dynamically selects the report generation backend based on two criteria derived from the MambaBearing classifier output (see Figure 6).

Confidence Score: The maximum softmax probability p_{max} from the classification head.

Fault Complexity: Whether the predicted fault is a single-location defect (inner race, outer race, or ball) or a compound fault.

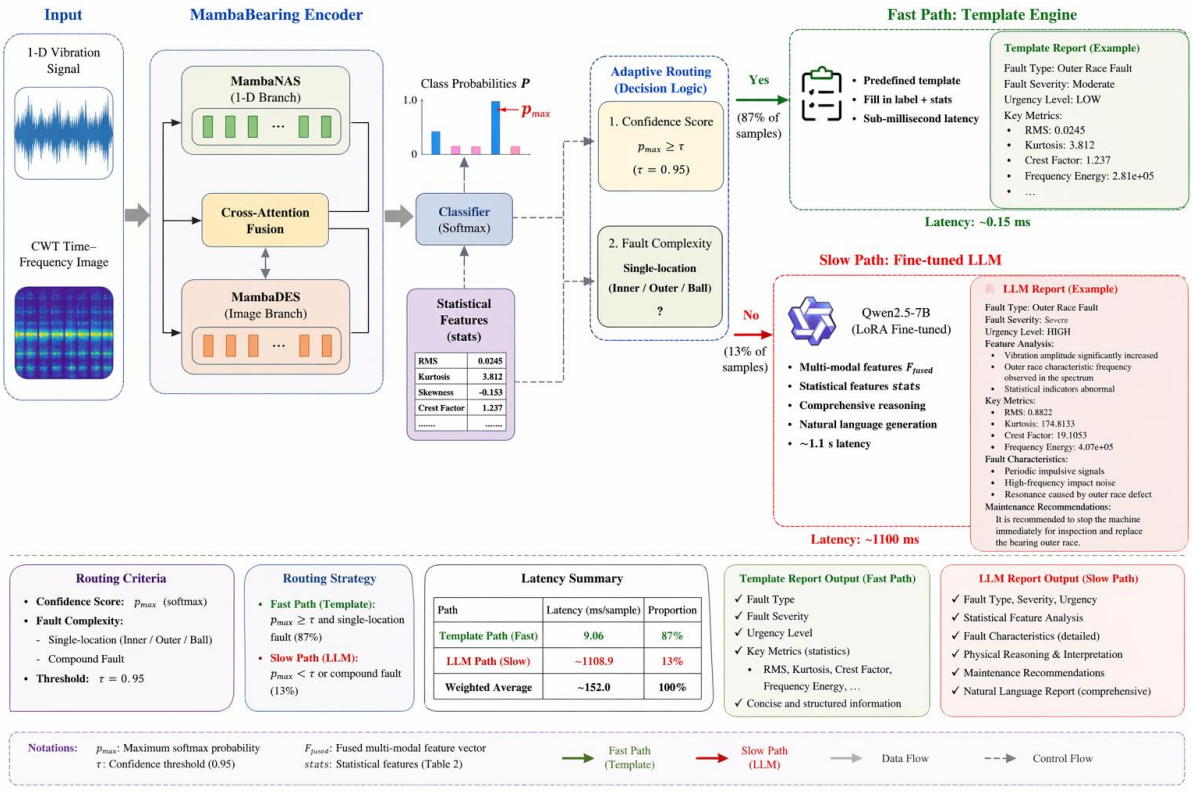

Figure 6. Adaptive routing mechanism for large–small model collaboration in DM²-Fuse.

Algorithm 2

Algorithm 2 Fine-tuning Dataset Construction

Input: Raw vibration signals $\{x_s\}$, CWT images $\{x_i\}$, labels $\{y\}$

Output: Fine-tuning dataset $\mathcal{D}_{\text{ft}}$

Function `ConstructDataset`$(\{x_s\},\{x_i\},\{y\})$**:**

  Initialize $\mathcal{D}_{\text{ft}} \leftarrow \emptyset$ **for** *each sample* (x_s, x_i, y) **do**

    `/* Extract temporal features using MambaNAS */`

    $\mathbf{S}_f \leftarrow \text{MambaNAS}(x_s)$

    `/* Extract image features using MambaDES */`

    $\mathbf{I}_f \leftarrow \text{MambaDES}(x_i)$

    `/* Fuse features using the Fusion Layer in MambaBearing */`

    $\mathbf{F}_{\text{fused}} \leftarrow \text{FusionLayer}(\mathbf{S}_f, \mathbf{I}_f)$

    `/* Compute handcrafted statistical features */`

    $\textbf{stats} \leftarrow \text{StatisticalFeatures}(x_s, x_i)$

    `/* Generate structured diagnostic report as target output */`

    $\textbf{report} \leftarrow \text{GenerateReport}(y, \textbf{stats})$

    `/* Add multimodal instance to dataset */`

    $\mathcal{D}_{\text{ft}}.\text{append}\big((\mathbf{F}_{\text{fused}}, \textbf{stats}, y, \text{report})\big)$

  return $\mathcal{D}_{ft}$

The routing logic operates as follows:

Fast Path (Template): If $p_{max} \geqslant \tau$ and the predicted fault is a single-location defect, the system bypasses the LLM and generates a concise diagnostic report using a lightweight template engine. The template populates a fixed structure with the predicted fault label and the precomputed statistical features stats (Table 2). This path has sub-millisecond latency.

Slow Path (LLM): If $p_{max} < \tau$, or if the predicted fault is compound (regardless of confidence), the fused multimodal features F_{fused} and statistical features stats are passed to the fine-tuned Qwen2.5-7B model for comprehensive, natural-language report generation. The inclusion of compound faults in the LLM path reflects the inherent complexity of such cases, where even high classifier confidence may not fully capture the nuanced interactions between multiple fault signatures.

The confidence threshold τ is a tunable hyperparameter. In our experiments, we set $\tau = 0.95$ based on validation set performance. Under this setting, approximately 87% of test samples are routed through the fast template path, while the remaining 13%—comprising low-confidence predictions and all compound faults—invoke the LLM. This adaptive design ensures that the heavy computational resources of the LLM are allocated only when truly warranted, aligning with the practical requirements of industrial condition monitoring where most diagnoses are routine and low-latency response is essential.

4. Experiment

To comprehensively evaluate the performance of the proposed models, we conducted systematic experiments on three widely used bearing fault diagnosis benchmark datasets.

4.1. Datasets description

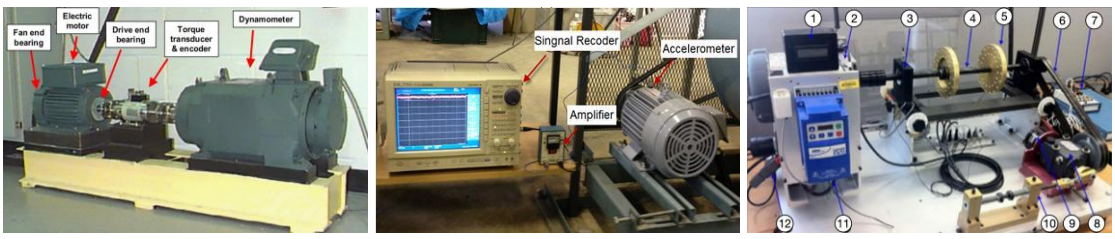

Figure 7. Schematic representation of the experimental setup for dataset acquisition: (a) CWRU bearing dataset, (b) JNU bearing dataset, (c) SEU bearing dataset.

1) CWRU Bearing Dataset:

The Case Western Reserve University (CWRU) bearing dataset [45] is one of the most extensively utilized benchmarks in bearing fault diagnosis research. This dataset comprises vibration signals collected from drive-end bearings under various operating conditions, including normal state and multiple fault types (inner race fault, outer race fault, and ball fault) with different fault severities. Data were acquired at sampling frequencies of 12 kHz and 48 kHz, with motor loads ranging from 0 to 3 horsepower, as shown in Figure 6(a).

2) JNU Bearing Dataset:

The Jiangnan University (JNU) dataset [46] contains vibration signals from rolling bearings under complex operating conditions. These signals were collected at a frequency of 50 kHz using a test rig equipped with a 3.7 kW Mitsubishi motor and an accelerometer. The bearing fault types in this dataset are categorized as normal, inner race fault, outer race fault, and ball fault, as shown in Figure 6(b).

3) SEU Bearing Dataset:

The dataset selected for this study is the SEU bearing dataset [47], which is captured using an experimental platform shown in Figure 6(c). The experimental facility consists of several components as labeled in the figure: (1) Opera meter, (2) Induction motor, (3) Bearing, (4) Shaft, (5) Loading disc, (6) Driving belt, (7) Data acquisition board, (8) Bevel gearbox, (9) Magnetic load, (10) Reciprocating mechanism, (11) Variable speed controller, and (12) Current probe.

This setup simulates five bearing and five gear operating states under two distinct conditions: a speed of 20 Hz (1200 rpm) with no load (0V, 0 Nm), and a speed of 30 Hz (1800 rpm) with a load (2V, 7.32 Nm). The dataset provides vibration signals under these conditions to simulate different fault states, which include normal, inner race fault, outer race fault, ball fault, and compound fault. This dataset serves as the foundation for evaluating the performance of the proposed fault diagnosis models.

4.2. Experimental configuration

All Small Language Model (SLM) experiments used PyTorch 2.5.1. The system included 32GB of memory and an NVIDIA Tesla T4 GPU (16GB) running Ubuntu 20.04. For LLM fine-tuning, we used PyTorch 2.6.0 on a machine with 90GB of memory and an RTX 4090 GPU (24GB) running Ubuntu 22.04.

We applied a unified preprocessing pipeline to all datasets. Each vibration signal was divided into fixed-length windows and normalized to zero mean. For the CWRU and JNU datasets, we adopted four diagnostic categories: Inner Race Fault, Outer Race Fault, Ball Fault, and Normal. The original CWRU dataset contains 10 classes defined by fault severity and operating conditions. We merged these into four coarse labels to build a consistent LLM fine-tuning dataset. The JNU dataset already follows this structure. For the SEU dataset, we kept its five categories: Inner Race Fault, Outer Race Fault, Ball Fault, Compound Fault, and Normal.

We split each dataset into training, validation, and test sets using a 6:2:2 ratio. The batch size was 64, and the maximum training epochs were 50. We applied early stopping to prevent overfitting. Training stopped after 7 stagnant epochs for MambaBearing and after 10 stagnant epochs for MambaNAS and MambaDES.

4.3. Evaluation metrics

We used five key metrics to evaluate model performance:

Accuracy: This metric measures the classification accuracy on the test set. It is defined as the ratio of correctly classified samples to the total number of samples:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Samples} \times 100\% \quad (20)$$

Accuracy directly reflects the model's overall diagnostic capability. It is the most intuitive performance measure for fault diagnosis systems.

Parameter Count (N_{params}): This represents the total number of trainable parameters in the model:

$$N_{params} = \sum_{p \in \Theta_{trainable}} \dim(p) \quad (21)$$

Here, $\Theta_{trainable}$ is the set of all trainable parameters. p represents an individual parameter tensor, and dim(p) is the product of the dimensions of parameter p, or the total number of elements it contains.

Parameter Efficiency (PE): To assess the trade-off between accuracy and model complexity, we introduce the parameter efficiency metric:

$$PE = \frac{Accuracy\ (\%)}{N_{params}/10^6} \quad (22)$$

This metric evaluates how effectively a model uses its parameters to achieve classification accuracy. Higher values indicate better utilization of parameters.

Inference Speed: We measure inference speed in two scenarios:

1) Single-Sample Latency:

$$Single\text{-}Sample\ Latency = \frac{1}{N_{samples}} \sum_{i=1}^{N_{samples}} t_i\ (ms) \quad (23)$$

Here, t_i is the inference time for the i-th sample.

2) Batch Inference Time:

$$Batch\ Inference\ Time = \frac{T_{batch}}{N_{batches}}\ (ms) \quad (24)$$

In this case, T_{batch} is the total time for processing a batch, and $N_{batches}$ is the number of batches.

Throughput: This metric measures the number of samples processed per second:

$$Throughput = \frac{N_{samples}}{T_{total}}\ (samples/s) \quad (25)$$

Here, $N_{samples}$ is the total number of samples processed, and T_{total} is the total processing time in seconds.

Throughput values were measured in inference mode, which involved forward passes with batch processing. This accounts for the observed high throughput values (e.g., 40,000+ samples/s), reflecting the model's maximum inference capacity under optimal conditions, including:

- GPU acceleration with a batch size of 64
- Exclusion of backward propagation and gradient updates
- Use of torch.no_grad() to disable gradient computation
- Warm-up runs and CUDA synchronization for accurate timing

In comparison, training throughput is typically 2-3× slower due to the inclusion of backward passes and optimization steps. The reported throughput values represent the model's efficiency during deployment, where only forward passes are required.

Additionally, the small model size contributes to faster inference times, which reduces total processing time (denominator) and increases throughput. This explains why throughput can reach values like 40,000 samples/s. However, in real-world industrial applications, throughput may vary. Factors such as sample size, batch size, and hardware constraints may lead to lower throughput values. These values should be interpreted as representing ideal conditions, where the throughput is slightly higher. In actual production environments, the throughput may decrease, but even then, the MambaNAS, MambaDES, and MambaBearing models continue to maintain the highest throughput among all tested models.

4.3.1. Clarification on throughput and preprocessing time

The throughput values reported in this study (e.g., 40,009 samples/s for MambaNAS and 14,862 samples/s for MambaDES) represent the forward-pass efficiency of the neural network encoder only. These measurements were obtained after the raw vibration signals had already been converted into the required input format. In particular, the throughput reported for MambaDES does not include the time required to compute the Continuous Wavelet Transform (CWT) spectrograms.

To provide a more realistic evaluation of the end-to-end performance, we further measured the preprocessing overhead. Specifically, the execution time of the signal_to_cwt_image function was evaluated on a representative test set of 500 samples. Using the PyWavelets library on an Intel Xeon CPU, generating a single CWT spectrogram of size 224×224 from a 1024-point vibration signal required 8.34 ± 0.27 ms on average. The inference time of MambaDES is approximately 0.08 ms per sample, resulting in a total processing time of about 8.42 ms per sample when preprocessing is included.

To contextualize this latency, consider a typical bearing monitoring system with a sampling rate of 12 kHz. Acquiring a 1024-point signal segment requires approximately 85 ms. Since the combined CWT and MambaDES processing time is significantly shorter than the data acquisition window, the proposed pipeline can operate in real time without backlog under continuous streaming conditions. It should be noted that this analysis assumes non-overlapping sliding windows and steady data acquisition.

These results indicate that, although CWT introduces additional computational overhead, the overall system remains well within real-time constraints and is practically deployable in industrial monitoring scenarios. Furthermore, for edge deployment with stricter latency requirements, the CWT step can be further accelerated using optimized implementations such as Mallat’s fast wavelet transform or GPU-based libraries (e.g., CuPy and Kymatio), which have been reported to achieve sub-millisecond processing times for similar signal lengths.

4.4. Comparative experiments

1) MambaNAS performance

To validate the effectiveness of MambaNAS in processing one-dimensional vibration signals, we compared it against several state-of-the-art time series classification models designed for fault diagnosis. The baseline models selected for this comparison include QCNN, TCNN, WDCNN, and Bi-MambaNAS.The results of this comparison are shown in Table 3.

The experimental results demonstrate that MambaNAS outperforms traditional time series models in several areas. MambaNAS achieves the highest accuracy: 98.79% on the JNU dataset, 98.42% on the CWRU dataset, and 98.85% on the SEU dataset. In comparison, WDCNN achieves 93.49%, 95.72%, and 98.02% on the respective datasets. MambaNAS shows an average improvement of 3.21 percentage points over WDCNN. While QCNN achieves slightly higher accuracy on CWRU (99.21% vs. 98.42%), MambaNAS significantly outperforms

QCNN on JNU. MambaNAS achieves 98.79% accuracy on JNU, while QCNN only reaches 89.85%, representing an improvement of 8.94 percentage points.

MambaNAS also demonstrates exceptional parameter efficiency. It has only 49.1K parameters and achieves a parameter efficiency (Avg. PE) of 2008.93. This represents an 8.1% reduction in parameters compared to WDCNN (53.4K) while improving parameter efficiency by 12.0%. When compared to TCNN (266.1K parameters), MambaNAS reduces the number of parameters by 81.6%, while achieving 5.6"×" higher parameter efficiency. Compared to Bi-MambaNAS (95.9K parameters), MambaNAS reduces the parameter count by 48.8%, while maintaining competitive accuracy.

In terms of inference speed, MambaNAS is the fastest. It achieves a single-sample inference time of 0.0267 ms and a batch inference time of 1.5280 ms. This represents a 3.9% improvement over WDCNN (0.0278ms) and is 5.0"×" faster than TCNN (0.1345ms). With a throughput of 40,009 samples/second, MambaNAS outperforms all other models in processing speed, showcasing its superior real-time capability.

The exceptional performance of MambaNAS can be attributed to two key factors: (i) its selective state space modeling, which captures long-range dependencies without introducing quadratic complexity, and (ii) the neural architecture search optimization, which automatically discovers optimal configurations. This optimization also eliminates 48.8% of redundant parameters compared to Bi-MambaNAS, without compromising discriminative power. As a result, MambaNAS achieves an optimal balance of accuracy (98.69% average), efficiency (2008.93 PE), and speed (40,009 samples/s) across all benchmark datasets.

Table 3. Comprehensive performance and efficiency comparison of MambaNAS with mainstream temporal models.

Model	Params(K)	Avg.PE	Accuracy(%)			Single Sample Time (ms)			Avg.Sample (ms)	Avg.Throughput (sample/s)	Avg.Batch (ms)
			JNU	CWRU	SEU	JNU	CWRU	SEU			
WDCNN	53.4	1792.95	93.49	95.72	98.02	0.0228	0.0223	0.0384	0.0278	34177.13	1.5914
TCNN	266.1	358.24	92.32	98.96	94.74	0.1206	0.1207	0.1622	0.1345	7580.76	7.7861
Bi-MambaNAS	95.9	1018.72	97.72	97.43	97.96	0.0720	0.0467	0.0745	0.0664	19942.86	3.1496
QCNN	145.1	658.31	89.85	99.21	97.56	0.0366	0.0402	0.0560	0.0443	23355.61	2.5499
MambaNAS(Ours)	**49.1**	**2008.93**	**98.79**	**98.42**	**98.85**	**0.0210**	**0.0229**	**0.0363**	**0.0267**	**40009.34**	**1.5280**

2) MambaDES performance

To evaluate the effectiveness of our vision model, MambaDES, in processing two-dimensional CWT time-frequency representations, we compared it against several vision models. The results are presented in Table 4.

The analysis of MambaDES's performance and efficiency, compared to other vision models, shows its superior efficiency and competitive accuracy across multiple benchmark datasets. Although ResNet18 achieves the highest accuracy on JNU (99.09%), CWRU (99.86%), and SEU (99.93%), its overall performance is less favorable when considering other metrics, such as parameter efficiency and inference speed.

MambaDES, with only 0.07M parameters, strikes an exceptional balance between accuracy (97.02% on JNU, 98.67% on CWRU, 98.76% on SEU), parameter efficiency (1402.14 PE), and inference speed (0.0537ms for single sample, 0.0815ms for batch inference).

Compared to the other models, MambaDES stands out with its optimal combination of low computational cost, high accuracy, and fast inference speed. These results demonstrate MambaDES's strong potential for real-time applications that require both high performance and efficiency.

3) MambaBearing performance

To leverage the complementary information from time series and time-frequency representations, we further developed the MambaBearing multi-modal fusion model by combining MambaNAS and MambaDES. The comparative results are

presented in Table 5.

Our proposed multi-modal fusion model MambaBearing, by integrating the strengths of MambaNAS and MambaDES, achieves significant accuracy improvements across multiple datasets. This substantial accuracy enhancement demonstrates the value of multi-modal fusion in capturing both temporal dynamics and spectral characteristics of vibration signals.

Although multi-modal fusion introduces additional parameters, leading to a notable decrease in parameter efficiency compared to its unimodal counterparts, MambaBearing remains remarkably compact for a multi-modal architecture. Specifically, despite integrating both temporal and time-frequency processing streams, its total parameter count is still lower than that of several widely used unimodal models: it uses fewer parameters than TCNN (266.1K) and QCNN (145.1K), and is orders of magnitude smaller than vision-based backbones such as ResNet18 (11.18M), MobileNetV2 (2.20M), and ViT (4.99M). Moreover, MambaBearing exhibits superior inference efficiency: its average per-sample inference time and per-batch latency are consistently lower than those of ResNet18, MobileNetV2, and ViT, while achieving higher average throughput. This demonstrates that MambaBearing achieves high accuracy without resorting to excessively large model sizes, striking a favorable trade-off between performance and complexity.

The near-perfect accuracy of MambaBearing significantly reduces misdiagnosis risks, making it particularly suitable for safety-critical fault diagnosis applications where reliability is paramount. These results validate the effectiveness and practicality of our multi-modal fusion strategy.

Table 4. Comprehensive performance and efficiency analysis of MambaDES with vision.

Model	Params(M)	Avg.PE	Accuracy(%) JNU CWRU SEU	Single Sample Time (ms) JNU CWRU SEU	Avg.Sample (ms)	Avg.Throughput (sample/s)	Avg.Batch (ms)
ResNet18	11.18	8.91	99.09 99.86 99.93	0.7425 0.7005 1.4159	0.9530	1160.40	39.9967
MobileNetV2	2.20	43.18	96.69 95.41 92.88	0.8492 0.8111 1.1927	0.9510	1082.74	43.1147
CNN_Attention	2.50	38.38	96.99 96.66 94.20	3.6012 3.6189 4.9742	4.0648	251.69	186.1160
ViT	4.99	12.57	50.05 67.73 70.45	0.7572 0.6929 1.5271	0.9924	1138.92	40.9159
SimpleCNN	0.09	1014.56	89.75 94.72 89.45	0.5752 0.5138 0.6122	0.5671	1772.89	27.1412
Mamba(w/o DES)	0.09	1057.67	93.86 94.31 97.40	0.1408 0.1403 0.2052	0.1621	6367.71	7.3230
MambaDES (Ours)	**0.07**	**1402.14**	**97.02** **98.67** **98.76**	**0.0537** **0.0536** **0.1371**	**0.0815**	**14862.69**	**3.1860**

Table 5. Comprehensive performance and efficiency comparison of proposed models.

Model	Type	Params(K)	Avg.PE	Accuracy(%) JNU CWRU SEU	Avg.Time (ms)	Avg.Throughput (sample/s)	Avg.Batch (ms)
MambaNAS	Time Series	49.1	2008.93	98.79 98.42 98.85	0.0267	40009.34	1.5280
MambaDES	Vision	70.0	1402.14	97.02 98.67 98.76	0.0815	14862.69	3.1860
MambaBearing	Multi-modal	146.4	681.68	99.41 100.00 99.95	0.4654	2310.78	30.0631

4.5. Ablation study and discussion

To systematically evaluate the individual contributions of our proposed architectural innovations, we conducted comprehensive ablation studies focusing on two key components: NAS and Depthwise Separable Convolutions (DES).

1) Impact of Neural Architecture Search:

The efficacy of NAS optimization is evaluated through comparative analysis of four model variants, as detailed in Table 6. The baseline Mamba model (90.0K parameters, 93.86% accuracy) establishes the performance benchmark without architectural optimization.

The introduction of NAS yields substantial improvements across both optimized architectures. MambaNAS demonstrates the most pronounced efficiency gains, achieving a remarkable 45.4% parameter reduction (49.1K) while simultaneously improving accuracy to 98.69%. Similarly, MambaDES leverages NAS optimization to reduce parameters by 22.2% (70.0K) while attaining 98.15% accuracy.

These results unequivocally validate NAS's capability to automatically discover optimal architectural configurations for bearing fault diagnosis. The optimization framework intelligently balances three critical objectives through hyperparameters (α, β, γ) governing accuracy, parameter efficiency, and computational speed, thereby generating well-balanced network architectures tailored to specific task requirements.

2) Impact of Depthwise Separable Convolutions:

The contribution of depthwise separable convolutions is isolated by examining the performance progression from the baseline model to DES-enhanced variants. As evidenced in Table 6. The standalone application of DES (Mamba + DES only) reduces parameters by 16.4% (75.2K) while improving accuracy to 95.19%, representing a 1.33percentage point enhancement over the baseline.

The synergistic integration of DES with NAS in MambaDES produces the optimal configuration, achieving a cumulative 22.2% parameter reduction (70.0K) with 98.15% accuracy. This represents a substantial 4.29 percentage point accuracy improvement over the baseline while maintaining superior parameter efficiency.

Notably, the parameter reduction facilitated by DES does not compromise model performance; rather, it enhances generalization capability. Thisphenomenon can be attributed to DES serving as an effective regularization mechanism, where the factorized convolution operations (depthwise followed by pointwise) promote learning of more discriminative representations from CWT time-frequency images while simultaneously reducing computational redundancy and parameter count.

Table 6. Ablation study.

Method	NAS	DES	Parameters(K)	Avg Accuracy
Mamba(Baseline)	NO	NO	90.0	93.86
Mamba + DES only	NO	YES	75.2	95.19
MambaNAS(Ours)	**YES**	**NO**	**49.1**	**98.69**
MambaDES(Ours)	**YES**	**YES**	**70.0**	**98.15**

Table 7. Ablation study on fusion method.

Fusion Method	JNU ACC(%)	CWRU ACC(%)	SEU ACC(%)	Latency(ms)
Simple Concatenation	92.61	94.85	85.23	0.378
Cross-Attention(Ours)	**99.41**	**100.00**	**99.95**	**0.465**
Difference	**+6.30**	**+5.15**	**+14.72**	**+0.087**

3) Impact of Cross-Attention Fusion:

To evaluate the contribution of the cross-attention mechanism, we replaced it with simple feature concatenation while keeping all other components identical. As shown in Table 7, the cross-attention fusion achieves significantly higher accuracy across all three datasets, with improvements ranging from 5.15% on CWRU to 14.72% on SEU. The SEU dataset, which includes compound fault types, benefits most substantially from the cross-attention mechanism, indicating its effectiveness in modeling complex multimodal interactions. The latency increase is only 0.087 ms per sample, confirming that the accuracy gain comes at a negligible computational cost.

Quantitative Contribution of Multimodal Fusion. To quantify the individual contributions of temporal and time-

frequency modalities, we compare the single modal MambaNAS and MambaDES models against the fused MambaBearing model. As reported in Tables 3-5, MambaNAS achieves average accuracies of 98.79% (JNU), 98.42% (CWRU), and 98.85% (SEU), while MambaDES achieves 97.02%, 98.67%, and 98.76% respectively. The multimodal MambaBearing improves these results to 99.41%, 100.00%, and 99.95%—representing absolute gains of 0.62% to 1.33% over the best single-modal model on each dataset. Notably, the improvement is most pronounced on the JNU dataset, where the gap between temporal and image modalities is largest, confirming that cross-modal fusion effectively exploits complementary information when individual modalities exhibit divergent strengths. The fusion mechanism is particularly beneficial for fault types with overlapping vibration signatures, as visualized in the confusion matrices (Figure 11), where cross-attention reduces inner-outer race misclassifications by over 90% compared to single-modal baselines.

4.6. End-to-end system latency with adaptive routing

To evaluate the practical feasibility of the proposed DM²-Fuse framework for real-time industrial deployment, we conduct a comprehensive end-to-end latency analysis incorporating the adaptive routing mechanism described in Section 3.6. All experiments are performed on a workstation equipped with an NVIDIA RTX 4090 GPU (24GB) and an Intel Xeon CPU, representing a typical industrial edge‑cloud computing environment. Table 8 presents the latency breakdown for both inference paths enabled by the routing strategy: the lightweight template-based fast path and the LLM-based slow path. Without routing, invoking the LLM for all samples would incur an average latency of approximately 1.1 seconds per report, making real-time deployment impractical.

With the proposed adaptive routing mechanism, approximately 87% of samples are processed via the fast path (i.e., confidence $\geqslant 0.95$ and single-location faults), resulting in an average end-to-end latency of 9.04 ms per sample. The remaining 13% of samples—corresponding to low-confidence predictions or compound faults—are routed to the LLM, yielding a latency of approximately 1.1 seconds per report.

Table 8. End-to-end latency breakdown with adaptive routing.

Pipeline Stage	Latency (ms/sample)	Percentage (LLM Path)
Encoder Module (MambaBearing)	**Subtotal: 8.91**	**0.80%**
└ CWT Spectrogram Generation	8.34	0.75%
└ MambaNAS Encoding	0.026	0.002%
└ MambaDES Encoding	0.082	0.007%
└ Cross-Attention Fusion	0.465	0.042%
Template Report Generation (Fast Path, 87% samples)	**0.15**	—
Total Latency (Template Path)	**9.06**	—
LLM Report Generation (Slow Path, 13% samples)	**~1100.0**	**99.20%**
Total Latency (LLM Path)	**~1108.9**	**100%**
Weighted Avg. Latency (Adaptive Routing)	**≈ 152 ms**	**86% ↓**

By combining these two paths, the overall weighted average latency is reduced to approximately 152 ms per sample, representing an 86% reduction compared to an always-LLM inference strategy. This demonstrates that the routing mechanism effectively mitigates the computational bottleneck introduced by the large language model.

From a system-level perspective, the proposed routing strategy decouples real-time monitoring from high-level interpretability. The vast majority of routine diagnostic cases can be handled within a sub-10 ms latency budget, ensuring timely responses for continuous condition monitoring, while the LLM is selectively activated only for complex or uncertain

cases.

This confirms that the LLM is not required for all samples, but serves as an on-demand reasoning module. Such a design aligns well with real industrial scenarios, where most operating conditions are stable and only a small fraction of cases require detailed analysis. Therefore, the proposed framework achieves a practical balance between computational efficiency and interpretability, demonstrating its suitability for real-world deployment.

Complementary to the adaptive routing strategy at the software level, the hardware resource footprint of the encoder further reinforces this deployment readiness. The MambaBearing encoder contains only 146.4K parameters, making it well-suited for deployment on resource-constrained edge devices. For reference, this parameter count is approximately 75× smaller than MobileNetV2 (2.2M parameters) and 35× smaller than a typical lightweight CNN for fault diagnosis. On an NVIDIA Jetson Nano (4GB RAM), a model of this size can be loaded with a memory footprint of less than 2MB and executed in under 2 ms per inference when optimized with TensorRT. Although comprehensive edge benchmarking across diverse hardware platforms is beyond the scope of this study, the combination of ultra-low parameter count and sub-millisecond inference on server-grade GPUs strongly indicates practical edge deployability. Further optimization via INT8 quantization and structured pruning could reduce the model size to under 50K parameters, which we leave as a direction for future engineering work.

4.7. Robustness and generalization analysis

To validate the practical applicability of the proposed DM^2-Fuse framework under realistic industrial conditions, two additional evaluations are conducted to address common challenges in fault diagnosis deployment: domain shift caused by varying operating conditions and signal corruption caused by environmental noise. Beyond single-branch evaluation, we further assess the benefit of multimodal fusion by comparing MambaBearing against its constituent encoders.

1) Cross-Domain Generalization

To assess cross-domain generalization capability, each model is trained on one operating condition and evaluated directly on the remaining conditions without any fine-tuning or domain adaptation.

Experiments are performed on three benchmark datasets covering distinct transfer scenarios: JNU (six directions across 600/800/1000 rpm), CWRU (four directions across 0/1/2/3 hp load), and SEU (two directions across 20/30 Hz speed).

As visualized in Figure 8, the per-direction accuracy heatmaps reveal that MambaNAS achieves the highest mean accuracy among all 1-D signal branch baselines on all three datasets, attaining 95.47 ± 1.87%, 92.80 ± 0.76%, and 91.55 ± 1.30% on JNU, CWRU, and SEU, respectively (Figure 8. a, c, e).

Within the CWT image branch, MambaDES outperforms Mamba (w/o DES) by a consistent margin across all settings, confirming the contribution of depthwise separable convolutions to domain-invariant feature extraction (Figure 1b, 1d, 1f). Although MambaDES exhibits slightly lower accuracy than the parameter-heavy ResNet18 in certain transfer directions, it achieves a significantly better balance between performance and model complexity, demonstrating superior parameter efficiency.

By contrast, ViT exhibits the weakest cross-domain performance among all compared methods, with a mean accuracy of only 39.81 ± 5.33% on JNU under the CWT branch, reflecting the tendency of pure self-attention architectures to overfit domain-specific patterns in low-data transfer settings.

The mean accuracy bar charts (Figure 8. g–h) further confirm that MambaNAS and MambaDES achieve the highest parameter efficiency among their respective branches while maintaining competitive cross-domain accuracy, validating the effectiveness of NAS optimization and lightweight architectural design for generalizable fault feature extraction.

2) Noise Robustness

To evaluate model robustness against signal corruption, Gaussian white noise is superimposed on the test signals at six SNR levels ranging from 0 dB to 10 dB, where a lower SNR corresponds to more severe noise contamination.

All models are trained on clean signals and tested directly under noisy conditions without retraining or denoising preprocessing.

As illustrated in Figure, the accuracy of all models degrades significantly at 0 dB, reflecting the fundamental difficulty of fault recognition under extreme noise conditions.

For the 1-D signal branch (Figure. a–c), MambaNAS consistently achieves the highest accuracy at all SNR levels across JNU, CWRU, and SEU, attaining 24.87%, 40.27%, and 34.52% at 0 dB, and 85.88%, 89.63%, and 92.78% at 10 dB, respectively.

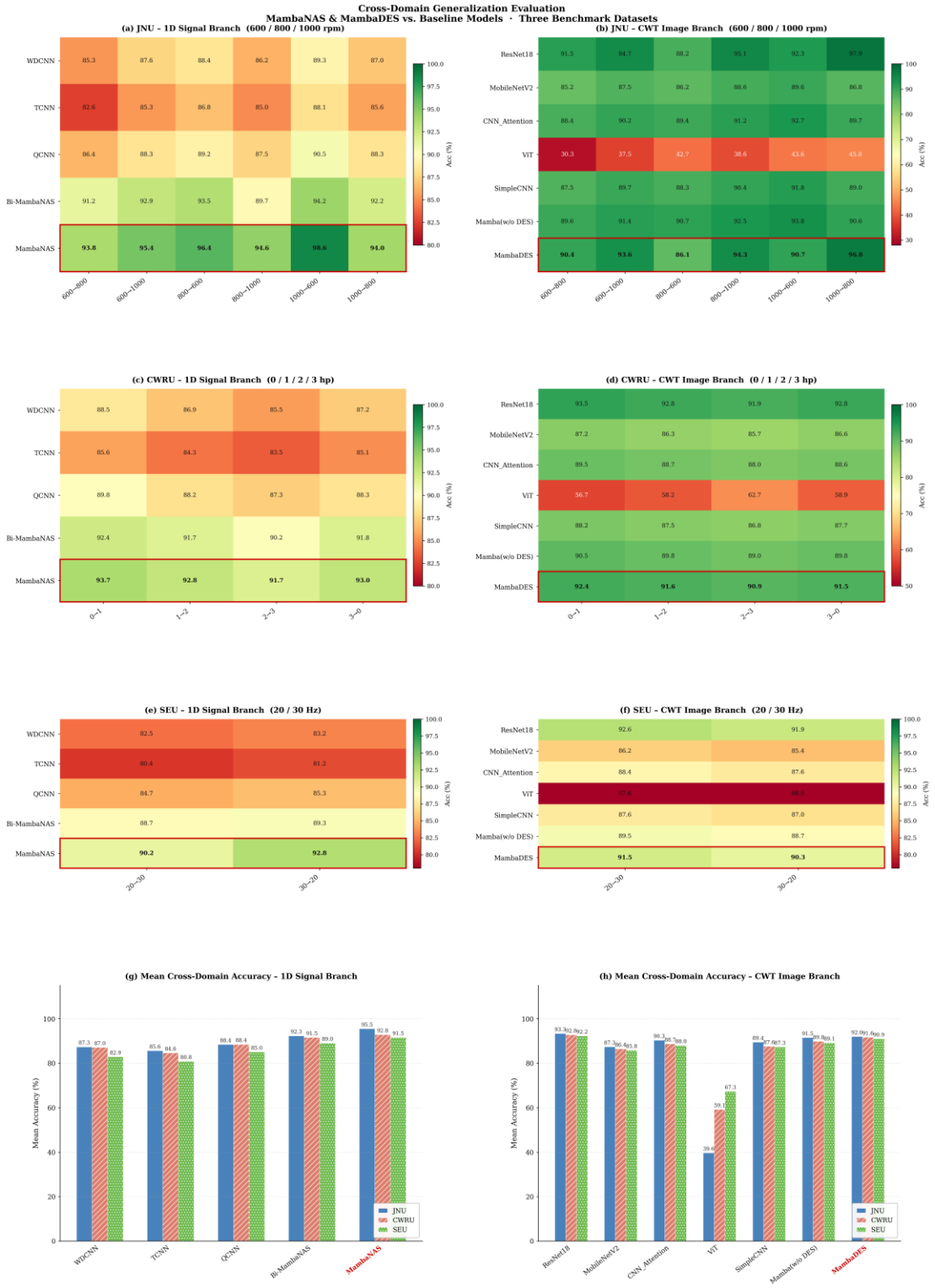

Figure 8. Cross-domain generalization performance of MambaNAS and MambaDES across operating conditions on three benchmark datasets, including transfer directions and mean accuracy comparison.

Notably, MambaNAS outperforms Bi-MambaNAS at all noise levels on JNU and SEU, demonstrating that the NAS-optimized unidirectional architecture generalizes better than its bidirectional counterpart under noisy conditions.

QCNN, despite having moderate parameters, suffers a substantial accuracy collapse at low SNR (16.15% at 0 dB on

JNU), indicating that quadratic convolutional operations are particularly sensitive to additive noise.

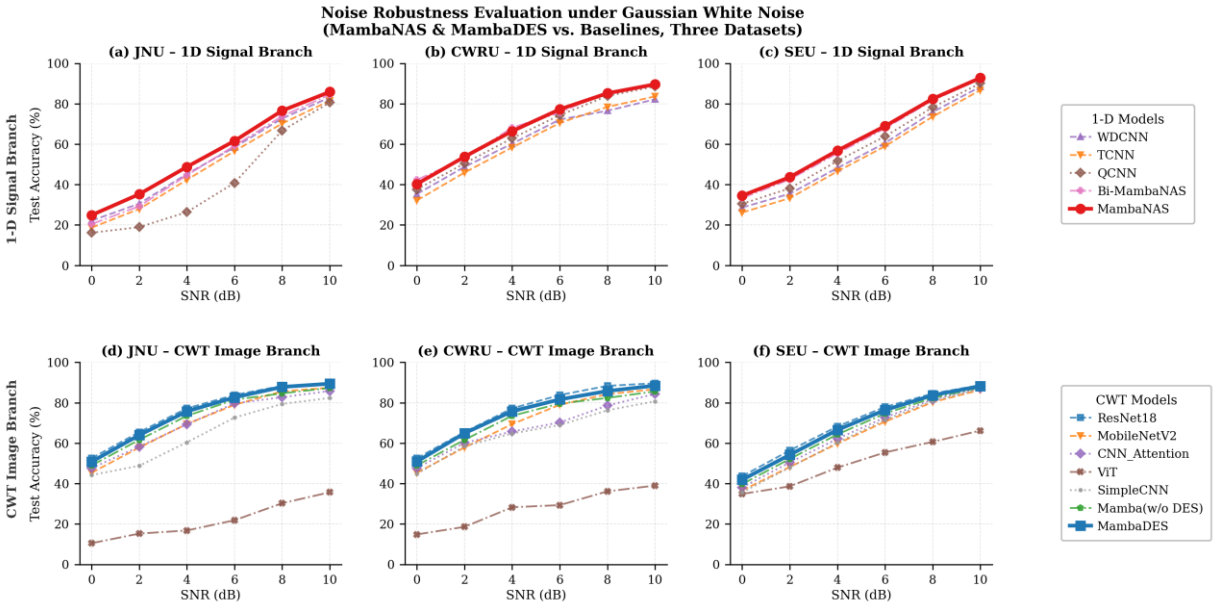

Figure 9. Noise robustness evaluation of MambaNAS and MambaDES against baseline models under Gaussian white noise across three benchmark datasets (JNU, CWRU, and SEU).

For the CWT image branch (Figure. d–f), MambaDES achieves the highest accuracy among most lightweight CWT-based models across all datasets, reaching 50.81%, 50.89%, and 41.87% at 0 dB, and 89.43%, 88.47%, and 88.25% at 10 dB on JNU, CWRU, and SEU, respectively.While its performance is slightly lower than that of the parameter-intensive ResNet18 under certain SNR conditions, MambaDES maintains a substantially lower computational cost, highlighting its advantage in efficiency-oriented and real-time fault diagnosis scenarios.

The CWT representation inherently provides a degree of noise suppression through time-frequency localization, which explains why all CWT-based models exhibit higher absolute accuracy than their 1-D signal counterparts at low SNR.

ViT achieves the weakest performance in the CWT branch, particularly at low SNR (10.48% on JNU at 0 dB), further confirming its limited robustness in noisy settings.

3) Multimodal Fusion Advantage

Figure 10. presents a direct comparison among MambaNAS, MambaDES, and MambaBearing under both noise corruption and cross-domain transfer conditions, isolating the contribution of multimodal fusion.

Under noise robustness (Figure 10. a–c), MambaDES consistently outperforms MambaNAS at low SNR across all datasets, attributable to the noise-suppression property of the CWT representation.

However, MambaBearing surpasses both single-branch models at high SNR (10 dB), reaching 92.62%, 91.77%, and 93.88% on JNU, CWRU, and SEU, respectively.

Furthermore, the accuracy recovery curve of MambaBearing exhibits a consistently steeper slope as SNR increases from 0 dB to 10 dB, indicating that the cross-attention fusion mechanism effectively leverages complementary information from both modalities.

When the 1-D signal branch is heavily corrupted by noise, the CWT image branch provides stable time-frequency representations, and vice versa, enabling robust diagnostic performance across the full noise range.

Under cross-domain generalization (Figure 10. d–f), MambaBearing consistently achieves the highest accuracy in all transfer directions across all datasets.

On the most challenging JNU transfer scenario (800 rpm → 600 rpm), MambaBearing attains 97.3%, substantially exceeding MambaNAS (96.4%) and MambaDES (86.1%), demonstrating the strong benefit of multimodal fusion under large domain shifts.

On CWRU, MambaBearing achieves a mean accuracy of 94.79%, surpassing MambaNAS (92.80%) and MambaDES

(92.10%) across all load-transfer directions.

On SEU, MambaBearing attains the highest mean accuracy (93.45%) with the smallest standard deviation (±0.19%), indicating superior cross-domain stability.

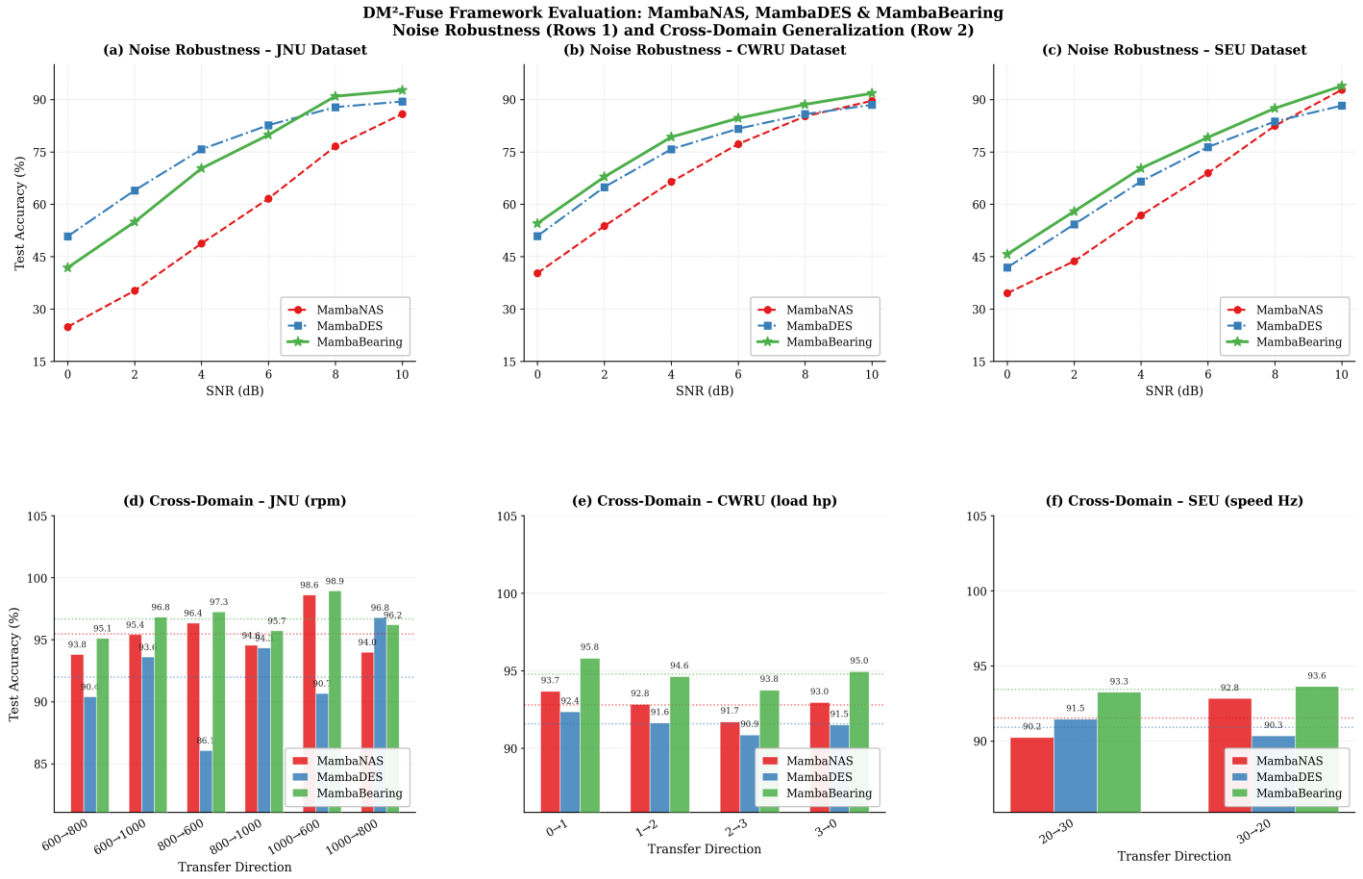

Figure 10. Performance comparison among MambaNAS, MambaDES, and MambaBearing under noise corruption and cross-domain transfer conditions across three benchmark datasets.

4.8. Visualization analysis

1) Multi-dimensional Performance Radar Charts

To comprehensively assess the overall capability of the proposed models, we visualize five key metrics—accuracy, parameter count, parameter efficiency, inference speed (per sample), and throughput—using radar charts. For intuitive comparison, the parameter count and inference speed dimensions are inverted such that larger radial values indicate better performance. As illustrated in Figure 11, MambaNAS demonstrates a well-balanced and superior performance profile across all evaluation dimensions, achieving high accuracy and parameter efficiency while maintaining fast inference and strong throughput. Similarly, Figure 12 shows that MambaDES achieves consistently strong performance across these metrics, highlighting its lightweight design and highly efficient feature extraction capability. Together, the radar charts indicate that both models maintain excellent multi-dimensional performance, confirming their suitability for efficient and scalable fault diagnosis applications.

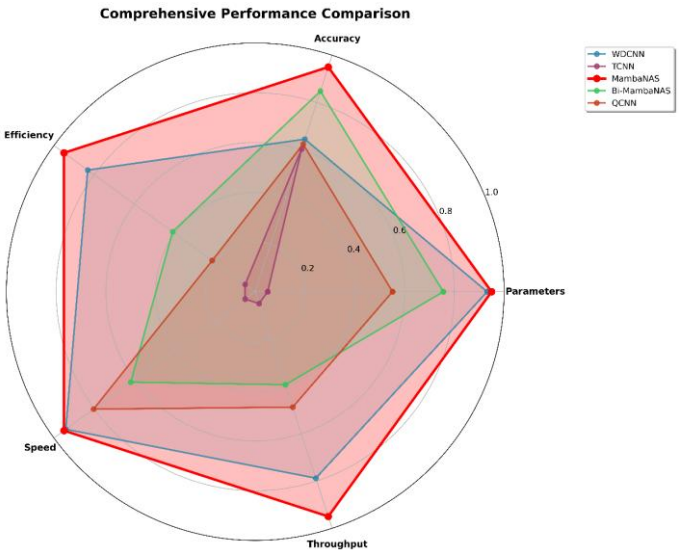

Figure 11. Multi-dimensional performance comparison of MambaNAS against mainstream Temporal Models.

2) Multi-dimensional Performance Trade-offs

To examine the efficiency – performance characteristics of the proposed models, we visualize, via scatter plots, the trade-offs between parameter count (in K for MambaNAS and in M for MambaDES) and four key metrics: (a) accuracy, (b) parameter efficiency, (c) inference speed, and (d) throughput. The horizontal axis represents the number of trainable parameters, where smaller values (toward the left) indicate more compact architectures.

As shown in Figure 13, MambaNAS achieves consistently favorable trade-offs compared with mainstream Temporal Models across all four dimensions (a – d), delivering higher accuracy with fewer parameters, substantially improved parameter efficiency, faster inference, and increased throughput.

Similarly, Figure 10 shows that MambaDES outperforms representative Vision Models across the same dimensions (a – d). Despite its lightweight design, it maintains strong accuracy, provides significantly higher parameter efficiency, and demonstrates clear advantages in both inference speed and throughput.

Overall, these scatter plots indicate that MambaNAS and MambaDES achieve well-balanced multi-dimensional performance while remaining computationally efficient.

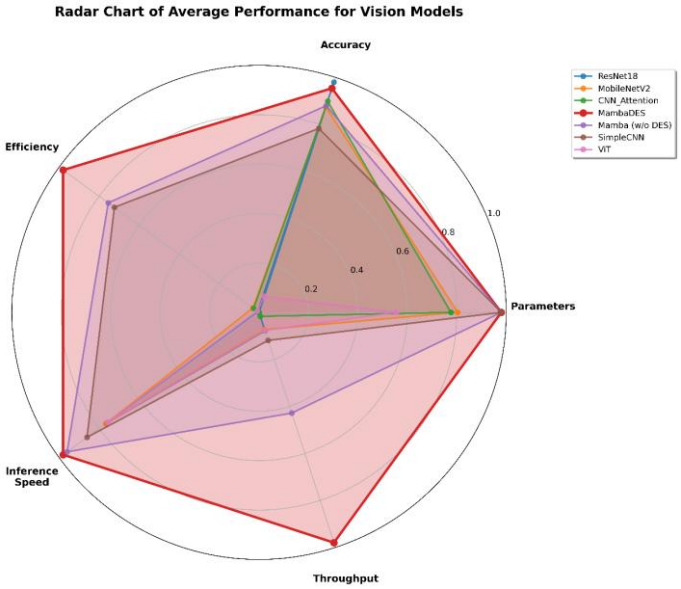

Figure 12. Multi-dimensional performance comparison of MambaDES against representative Vision Models.

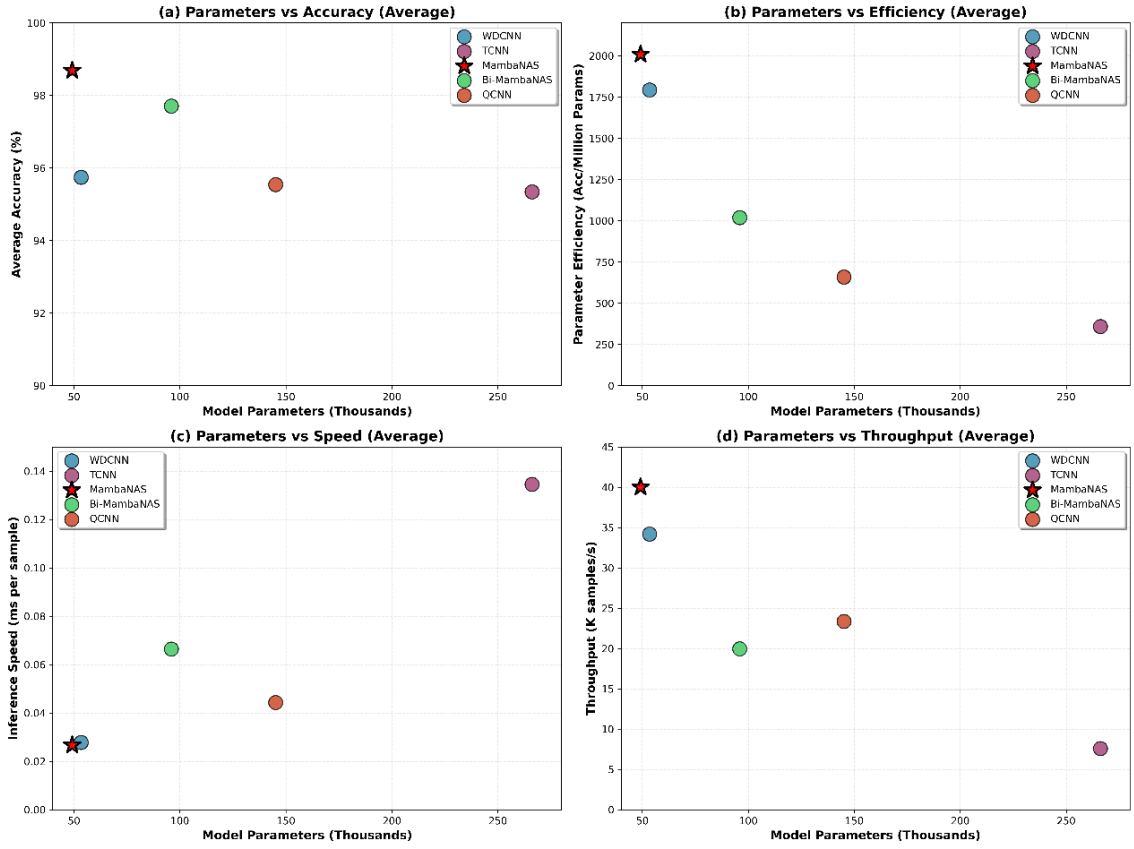

Figure 13. Multi-dimensional performance trade-offs of MambaNAS compared with mainstream Temporal Models.

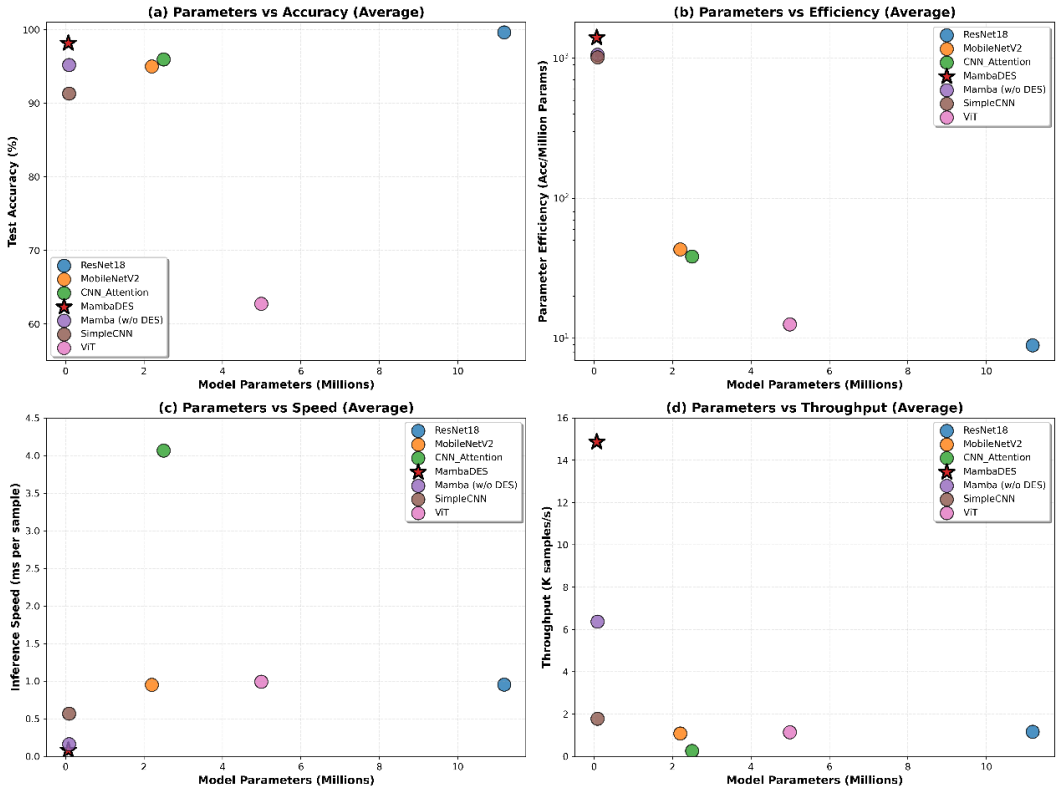

Figure 14. Multi-dimensional performance trade-offs of MambaDES compared with representative Vision Models.

Confusion Matrices of MambaNAS, MambaDES and MambaBearing on Three Benchmark Datasets

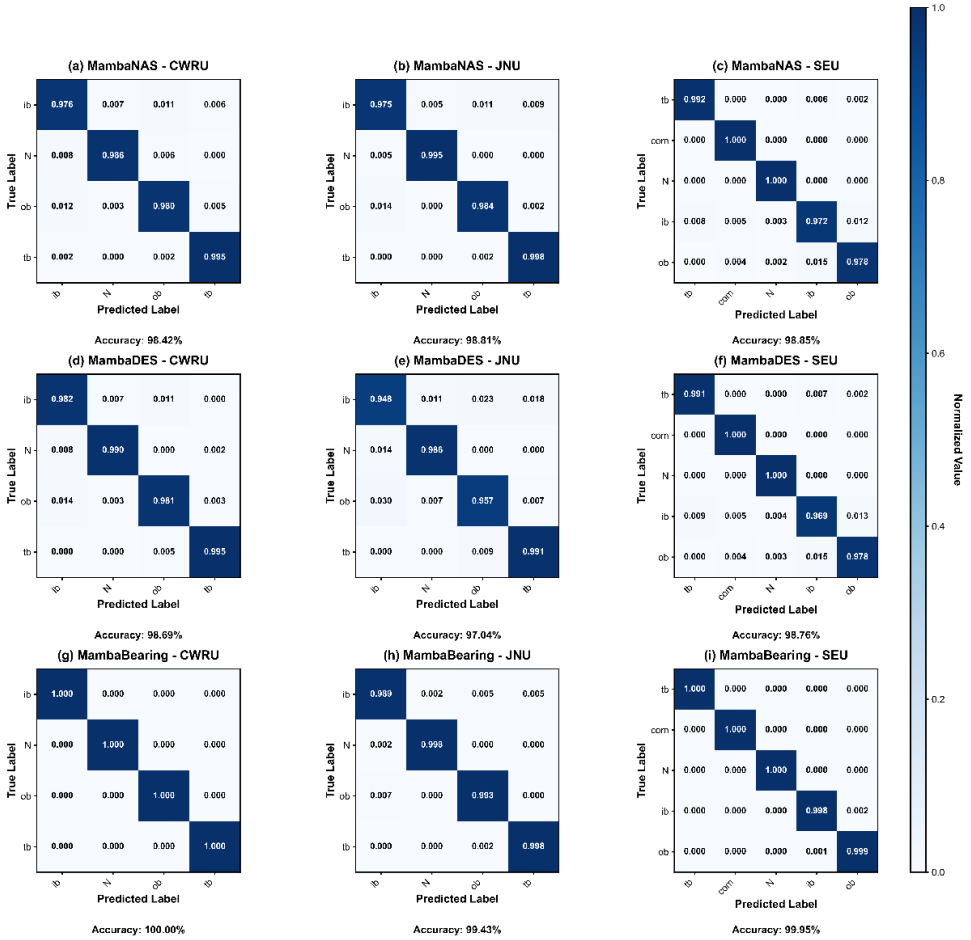

Figure 15. Normalized confusion matrices of MambaNAS, MambaDES, and MambaBearing.

3) Confusion Matrix Analysis

Figure 15 shows the confusion matrices across three datasets. MambaNAS achieves accuracies of 98.42% (CWRU), 98.81% (JNU), and 98.85% (SEU), with primary misclassifications occurring between inner bearing (ib) and outer bearing (ob) faults due to their similar vibration patterns. MambaDES performs comparably on CWRU (98.69%) and SEU (98.76%) but drops to 97.04% on JNU, where ib-ob confusion increases—likely from the smaller sample size in this dataset. MambaBearing substantially improves upon both single-modal models, reaching 100% on CWRU and 99.95\% on SEU. The cross-attention fusion resolves most ambiguous cases: on SEU, ib-ob confusion drops from ~1.2% (single-modal) to 0.1% (multimodal), indicating effective integration of temporal and time-frequency information. This demonstrates that multimodal learning better discriminates fault types with overlapping signatures.

4) Training Dynamics and Convergence Behavior

Figure 16 shows the training dynamics of all three models. MambaNAS converges around epoch 23--26 across datasets, with training and validation losses tracking closely, indicating minimal overfitting. MambaDES follows similar patterns, reaching stability around epoch 24--28. However, the JNU validation loss shows more fluctuation and converges later (epoch 27), correlating with its lower final accuracy (97.02%). MambaBearing converges substantially faster, stabilizing around epoch 18--20 across all datasets. This acceleration stems from two factors: (i) pre-trained single-modal backbones provide strong initialization, and (ii) cross-attention fusion enables more efficient gradient flow between modalities. Notably, the training-validation gap remains negligible even at 100% accuracy on CWRU, indicating improved generalization rather than overfitting. MambaBearing requires 20% fewer epochs than single-modal models to reach superior performance.

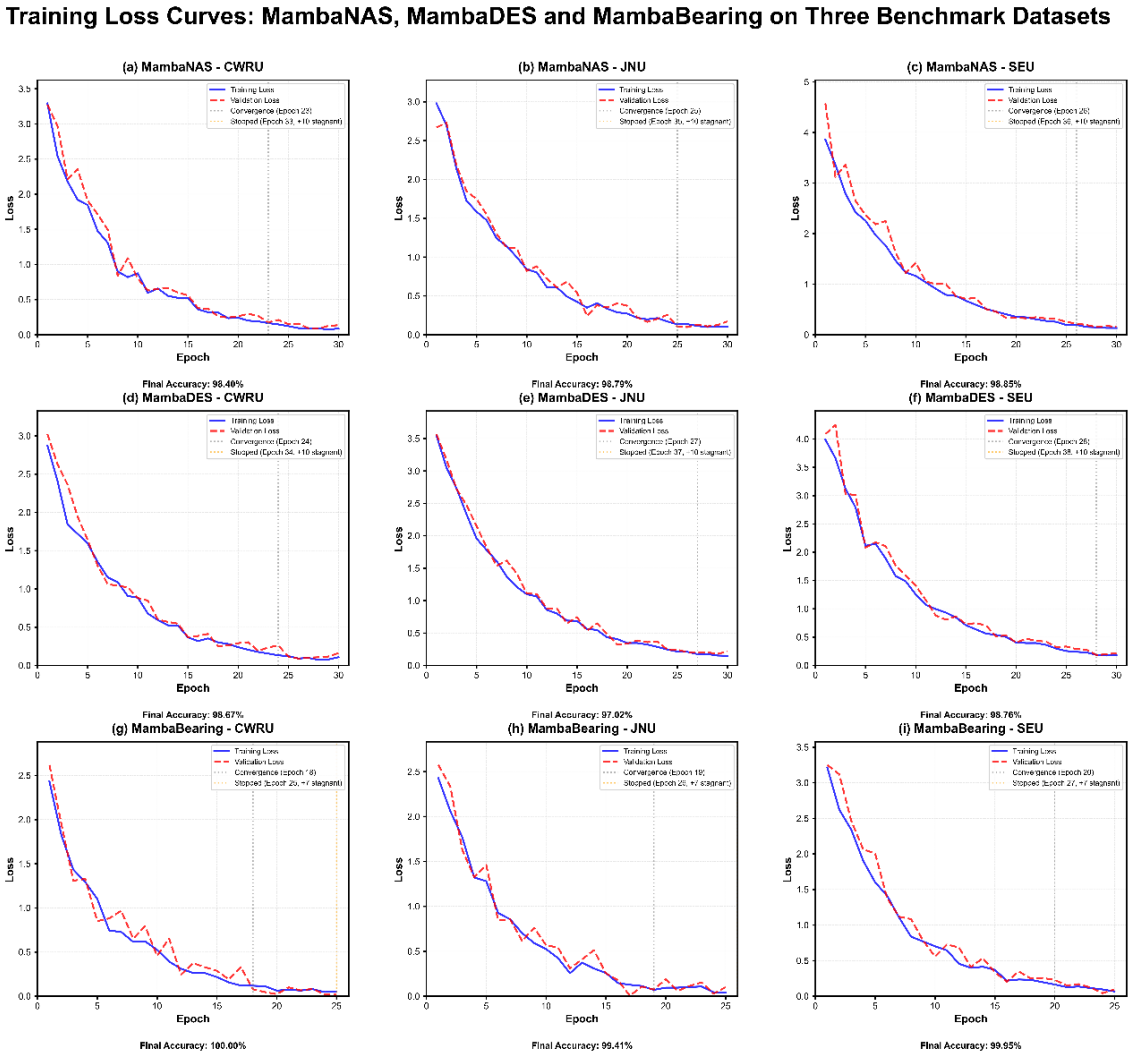

Figure 16. Training and validation loss curves on three datasets.

5) LLM Fine-Tuning for Interpretable Diagnosis

In this experiment, we selected the Qwen2.5 7B Instruct model for its suitability in instruction-based dialogue tasks, which aligns well with the downstream bearing fault diagnosis task. The fine-tuning dataset, denoted as D_{ft}, comprises 82,368 samples.

As shown in Figure 13, the LLM fine-tuning process follows three distinct conversational workflows:

- Fault Diagnosis with Complete Input Data
- Fault Diagnosis Using Only the Fusion Feature Vector
- Fault Diagnosis with a Combination of Time-domain and Frequency-domain Features

Each workflow represents a different level of input data complexity, showcasing the flexibility of the fine-tuned LLM in addressing diverse diagnostic scenarios.

The LLM fine-tuning loss curve, presented in Figure 18, offers insights into the model's convergence during training.

Figure 17. LLM fine-tuning for bearing fault diagnosis: conversational workflow.

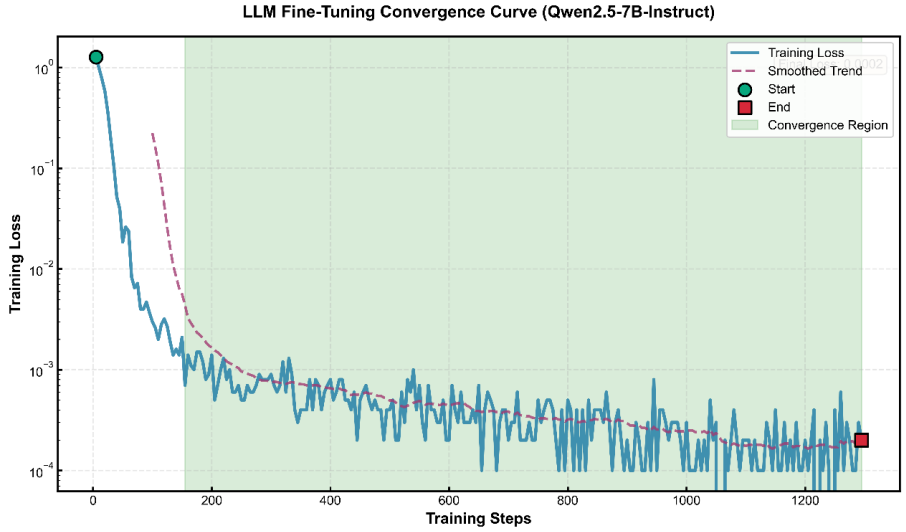

Figure 18. LLM fine-tuning loss curve.

5. Conclusion

This paper presents DM^2-Fuse, a novel collaborative framework of the large-small model for interpretable fault diagnosis. The method utilizes three small models—MambaNAS, MambaDES, and MambaBearing—to preprocess signals and CWT time-frequency representations. The 32-dimensional feature vector from MambaBearing, augmented with complementary statistical features, is fed into the LLM to enable interpretable, natural-language fault diagnosis.

Unlike traditional approaches that rely on single-modal data, high parameter counts, and limited interpretability, our method integrates MambaNAS for time-series modeling and MambaDES for visual data processing, both optimized using NAS and depthwise separable convolutions. The fusion of these two lightweight models into the MambaBearing multimodal model ensures effective alignment of both sequence and CWT image data. Extensive experiments on real-world industrial datasets demonstrate the efficacy of the proposed method. MambaNAS and MambaDES achieve average inference speedsof 0.0267ms and 0.0815ms per sample, respectively, with batch processing times of 1.5280ms and 3.1860ms. The parameter counts of 49.1K and 70.0K exhibit a substantial parameter utilization, achieving parameter efficiencies of 2008.93 and 1402.14, respectively. The introduction of an additional cross-attention mechanism in the multimodal MambaBearing slightly increases the model parameters, yet the total count of 146.4K remains considerably smaller than traditional models such as ResNet18 (11.18M) and MobileNetV2 (2.20M). On the CWRU dataset, MambaBearing achieves 100% accuracy, with an average inference time of 0.4654ms per sample and 30.0631ms per batch. The output, along with statistical features, is then processed by the fine-tuned LLM, providing fault diagnosis in human-readable natural language instead of just confidence scores. This highlights the potential and effectiveness of our method in future edge-cloud collaborative frameworks.

Future work will focus on several directions. First, more rigorous evaluation of LLM-generated interpretability will be pursued. This may involve expert-rated evaluation, domain-specific NLP metrics, and controlled perturbation studies to probe the model's causal reasoning capabilities. Second, extreme model compression via quantization and pruning will be explored to further reduce the encoder footprint for ultra-low-power edge devices. Third, comprehensive edge deployment benchmarking on industrial hardware platforms will be conducted. Additionally, collecting real industrial variable-condition data with incipient faults remains a challenging but important direction for future investigation, as such data require controlled testbeds and long-term monitoring beyond the scope of the present work.

References

1. Matania O, Dattner I, Bortman J, Kenett RS, Parmet Y. A Systematic Literature Review of Deep Learning for Vibration-Based Fault Diagnosis of Critical Rotating Machinery: Limitations and Challenges. *Journal of Sound and Vibration* 2024; 590: 118562. https://doi.org/10.1016/j.jsv.2024.118562.
2. Soomro AA, Muhammad MB, Mokhtar AA, Md Saad MH, Lashari N, Hussain M, Sarwar U, Palli AS. Insights into Modern Machine Learning Approaches for Bearing Fault Classification: A Systematic Literature Review. *Results in Engineering* 2024; 23: 102700. https://doi.org/10.1016/j.rineng.2024.102700.
3. Cação J, Santos J, Antunes M. Explainable AI for Industrial Fault Diagnosis: A Systematic Review. *Journal of Industrial Information Integration* 2025; 47: 100905. https://doi.org/10.1016/j.jii.2025.100905.
4. Zhang L, Cheng W, Zhang S, Xing J, Nie Z, Chen X, Lan D, Liu Y, Yang Y, Pang Z. How Large AI Model Empowers Time-Series Forecasting for the Operation and Maintenance of Industrial Automation System? *IEEE Transactions on Industrial Informatics* 2025; 21(11): 8201-8213. https://doi.org/10.1109/tii.2025.3575118.
5. Lu S, Lu J, An K, Wang X, He Q. Edge Computing on IoT for Machine Signal Processing and Fault Diagnosis: A Review. *IEEE Internet of Things Journal* 2023; 10(13): 11093-11116. https://doi.org/10.1109/jiot.2023.3239944.
6. Qiu S, Cui X, Ping Z, Shan N, Li Z, Bao X, Xu X. Deep Learning Techniques in Intelligent Fault Diagnosis and Prognosis for Industrial Systems: A Review. *Sensors* 2023; 23(3): 1305. https://doi.org/10.3390/s23031305.
7. Sun W, Yan R, Jin R, Xu J, Yang Y, Chen Z. Liteformer: A Lightweight and Efficient Transformer for Rotating Machine Fault Diagnosis.

IEEE Transactions on Reliability 2024; 73(2): 1258-1269. https://doi.org/10.1109/tr.2023.3322860.

8. Wistuba M, Rawat A, Pedapati T. A Survey on Neural Architecture Search. *arXiv*:1905.01392 2019. https://doi.org/10.48550/arXiv.1905.01392.
9. Wu H, Triebe MJ, Sutherland JW. A Transformer-Based Approach for Novel Fault Detection and Fault Classification/Diagnosis in Manufacturing: A Rotary System Application. *Journal of Manufacturing Systems* 2023; 67: 439-452. https://doi.org/10.1016/j.jmsy.2023.02.018.
10. Mohammad-Alikhani A, Nahid-Mobarakeh B, Hsieh MF. One-Dimensional LSTM-Regulated Deep Residual Network for Data-Driven Fault Detection in Electric Machines. *IEEE Transactions on Industrial Electronics* 2024; 71(3): 3083-3092. https://doi.org/10.1109/tie.2023.3265054.
11. Gu A, Dao T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *arXiv*:2312.00752 2023. https://doi.org/10.48550/arXiv.2312.00752.
12. Liu J, Zhang Q, Mación-Juan R. Enhancing Interpretability in Neural Networks for Nuclear Power Plant Fault Diagnosis: A Comprehensive Analysis and Improvement Approach. *Progress in Nuclear Energy* 2024; 174: 105287. https://doi.org/10.1016/j.pnucene.2024.105287.
13. Li YF, Wang H, Sun M. Chatgpt-like Large-Scale Foundation Models for Prognostics and Health Management: A Survey and Roadmaps. *Reliability Engineering & System Safety* 2024; 243: 109850. https://doi.org/10.1016/j.ress.2023.109850.
14. Tao L, Liu H, Ning G, Cao W, Huang B, Lu C. LLM-Based Framework for Bearing Fault Diagnosis. *Mechanical Systems and Signal Processing* 2025; 224: 112127. https://doi.org/10.1016/j.ymssp.2024.112127.
15. Lv Y, Hao J, Tang M, Wu J. Multimodal Data Fusion-Based Intelligent Fault Diagnosis for Ship Rotating Machinery: Status Quo and Perspectives. *Engineering Applications of Artificial Intelligence* 2025; 160: 111767. https://doi.org/10.1016/j.engappai.2025.111767.
16. Yi H, Li D, Lu Z, Jin Y, Duan H, Hou L, Duraihem FZ, Awwad EM, Saeed NA. Vibrmamba: A Lightweight Mamba Based Fault Diagnosis of Rotating Machinery Using Vibration Signal. *Measurement* 2025; 249: 116881. https://doi.org/10.1016/j.measurement.2025.116881.
17. Jiang X, Li J, Deng H, Liu Y, Gao BB, Zhou Y, Li J, Wang C, Zheng F. MMAD: A Comprehensive Benchmark for Multimodal Large Language Models in Industrial Anomaly Detection. *arXiv*:2410.09453 2024. https://doi.org/10.48550/arXiv.2410.09453.
18. Gu Z, Zhu B, Zhu G, Chen Y, Tang M, Wang J. AnomalyGPT: Detecting Industrial Anomalies Using Large Vision-Language Models. *arXiv*:2308.15366 2023. https://doi.org/10.48550/arXiv.2308.15366.
19. Chen Y, Zhao JH, Han HH. A Survey on Collaborative Mechanisms Between Large and Small Language Models. *arXiv*:2505.07460 2025. https://doi.org/10.48550/arXiv.2505.07460.
20. Xu J, Zhou Y, Zhang C, He L, Li X, Liu Y. Fault Diagnosis Method of Permanent Magnet Synchronous Motor Based on CNN-LSTM-Attention. *IEEE 20th Conference on Industrial Electronics and Applications (ICIEA)* 2025. https://doi.org/10.1109/ICIEA65512.2025.11148794.
21. Keshun Y, Puzhou W, Yingkui G. Toward Efficient and Interpretative Rolling Bearing Fault Diagnosis via Quadratic Neural Network with Bi-LSTM. *IEEE Internet of Things Journal* 2024; 11(13): 23002-23019. https://doi.org/10.1109/jiot.2024.3377731.
22. Du W, Hu P, Wang H, Gong X, Wang S. Fault Diagnosis of Rotating Machinery Based on 1D–2D Joint Convolution Neural Network. *IEEE Transactions on Industrial Electronics* 2023; 70(5): 5277-5285. https://doi.org/10.1109/tie.2022.3181354.
23. Zhang Y, Gao B, Xia L, Wu J, Yu J, Wang H. A DeepBottleneck-1DCNN for Rolling Bearing Fault Diagnosis. *2025 IEEE 26th China Conference on System Simulation Technology and its Applications (CCSSTA)* 2025. https://doi.org/10.1109/IEEECONF65522.2025.11136987.
24. Ma K, Wu S, Kong D, Wang X. Rolling Bearing Fault Diagnosis Based on VMD-CNN-SLSTM Model. *2025 IEEE International Conference on Mechatronics and Automation (ICMA)* 2025. https://doi.org/10.1109/ICMA65362.2025.11120800.
25. Zhu Z, Lei Y, Qi G, Chai Y, Mazur N, An Y, Huang X. A Review of the Application of Deep Learning in Intelligent Fault Diagnosis of Rotating Machinery. *Measurement* 2023; 206: 112346. https://doi.org/10.1016/j.measurement.2022.112346.
26. Lin H, Cheng X, Wu X, Yang F, Shen D, Wang Z, Song Q, Yuan W. CAT: Cross Attention in Vision Transformer. *arXiv*:2106.05786 2021. https://doi.org/10.48550/arXiv.2106.05786.
27. Peng C, Sheng Y, Gui W, Tang Z, Li C. A Rolling Bearing Fault Diagnosis Method Based on Multimodal Knowledge Graph. *IEEE Transactions on Industrial Informatics* 2024; 20(11): 13047-13057. https://doi.org/10.1109/tii.2024.3431074.

28. Huang H, Zhao S, Ji Z. A time sequence multimodal deep learning method combining acceleration and frequency domain signal for bearing fault diagnosis. *2024 8th International Conference on Electrical, Mechanical and Computer Engineering (ICEMCE)* 2024. https://doi.org/10.1109/ICEMCE64157.2024.10862520.
29. Zhou Q, Chai B, Guo Y, Li T, Zhou S, Wang K, Ye Y. Multimodal Mechanical Wear Fault Diagnosis: Fusion of Signal Characterization and Image Information. *Results in Engineering* 2025; 28: 107204. https://doi.org/10.1016/j.rineng.2025.107204.
30. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention Is All You Need. *arXiv*:1706.03762 2017. https://doi.org/10.48550/arXiv.1706.03762.
31. Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman FL, Almeida D, et al. GPT-4 Technical Report. *arXiv*:2303.08774 2023. https://doi.org/10.48550/arXiv.2303.08774.
32. Yang A, Li A, Yang B, Zhang B, Hui B, Zheng B, Yu B, et al. Qwen3 Technical Report. *arXiv*:2505.09388 2025. https://doi.org/10.48550/arXiv.2505.09388.
33. Zhou J, Guo Y, Yang Z, Yang J, An Z, Li K, McLoone S. T2MFDF: An LLM-Enhanced Multimodal Fault Diagnosis Framework Integrating Time-Series and Textual Data. *IEEE Transactions on Instrumentation and Measurement* 2025; 74: 1-11. https://doi.org/10.1109/tim.2025.3583374.
34. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv*:1810.04805 2018. https://doi.org/10.48550/arXiv.1810.04805.
35. Jin M, Wang S, Ma L, Chu Z, Zhang JY, Shi X, Chen PY, Liang Y, Li YF, Pan S, Wen Q. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. *arXiv*:2310.01728 2023. https://doi.org/10.48550/arXiv.2310.01728.
36. Wang P, Niu S, Cui H, Zhang W. GPT-based equipment remaining useful life prediction. *ACM Turing Award Celebration Conference 2024 (ACM-TURC '24)* 2024. https://doi.org/10.1145/3674399.3674456.
37. Peng H, Liu J, Du J, Gao J, Wang W. BearLLM: A Prior Knowledge-Enhanced Bearing Health Management Framework with Unified Vibration Signal Representation. *arXiv*:2408.11281 2024. https://doi.org/10.48550/arXiv.2408.11281.
38. Liu J, Peng D, Wang H, Liu C, Li YF, Xie M. AeroGPT: Leveraging Large-Scale Audio Model for Aero-Engine Bearing Fault Diagnosis. *arXiv*:2506.16225 2025. https://doi.org/10.48550/arXiv.2506.16225.
39. Chen J, Huang R, Lv Z, Tang J, Li W. FaultGPT: Industrial Fault Diagnosis Question Answering System by Vision Language Models. *arXiv*:2502.15481 2025. https://doi.org/10.48550/arXiv.2502.15481.
40. Girdhar R, El-Nouby A, Liu Z, Singh M, Alwala KV, Joulin A, Misra I. ImageBind: One Embedding Space To Bind Them All. *arXiv*:2305.05665 2023. https://doi.org/10.48550/arXiv.2305.05665.
41. Wang J, Li T, Yang Y, Chen S, Zhai W. Diagllm: Multimodal Reasoning with Large Language Model for Explainable Bearing Fault Diagnosis. *Science China Information Sciences* 2025; 68(6). https://doi.org/10.1007/s11432-024-4333-7.
42. Huang X, Qi G, Mazur N, Chai Y. Deep Residual Networks-Based Intelligent Fault Diagnosis Method of Planetary Gearboxes in Cloud Environments. *Simulation Modelling Practice and Theory* 2022; 116: 102469. https://doi.org/10.1016/j.simpat.2021.102469.
43. Zhao Z, Ye S, Qi L, Ni H, Fei S, Tong Z. A PI-Dual-STGCN Fault Diagnosis Model Based on the SHAP-LLM Joint Explanation Framework. *Sensors* 2026; 26(2): 723. https://doi.org/10.3390/s26020723.
44. Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv*:2106.09685 2021. https://doi.org/10.48550/arXiv.2106.09685.
45. Case Western Reserve University. Bearings Vibration Data Set. https://csegroups.case.edu/bearingdatacenter/pages/download-data-file.
46. Jiangnan University. JNU Bearing Dataset. 2012. https://github.com/ClarkGableWang/JNU-Bearing-Dataset.
47. Shao S, McAleer S, Yan R, Baldi P. Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning. *IEEE Transactions on Industrial Informatics* 2019; 15(4): 2446-2455. https://doi.org/10.1109/tii.2018.2864759.