

Article

PGA-LLM: A Probability-Guided Alignment Large Language Model Framework for Fault Diagnosis

Tao Wang ¹, Yanqiang Di ^{1,*}, Shaochong Feng ¹, Qiongyao Liu ¹, Haohao Cui ¹ and Qing Liu ^{1,2}

¹ Shijiazhuang Campus, Army Engineering University of PLA, Shijiazhuang 050051, China; wangtaoasis@aeu.edu.cn (T.W.); liu_qy1995@163.com (Q.L.); cuihaosim@aeu.edu.cn (H.C.); qing.liu@hebust.edu.cn (Q.L.)

² School of Economics and Management, Hebei University of Science and Technology, Shijiazhuang 051432, China

* Correspondence: yq.simu@aeu.edu.cn

Abstract

Fault diagnosis for complex industrial equipment plays a crucial role in safeguarding production safety and advancing the capabilities of intelligent operation and maintenance. Current deep learning approaches have demonstrated promising accuracy in fault classification tasks; however, their signal representations alone cannot provide a transparent interface for embedded large language models. To tackle the aforementioned challenges, we propose PGA-LLM, a novel fault diagnosis framework for industrial equipment that leverages large language models via probability-guided alignment. First, a variational autoencoder (VAE)-based signal encoder embedded with reconstruction constraints is established. Joint reconstruction and classification objectives balance discriminative representation learning and signal reconstruction. Second, the probability-guided alignment (PGA) module combines fault-class probability guidance with a residual feature path; a learned gate fuses both paths before continuous soft-prompt projection. Furthermore, a progressive three-stage training scheme is adopted, encompassing encoder pre-training, interface optimization, and low-rank adaptation (LoRA) of Qwen2.5-1.5B. Extensive experiments are carried out on four standard datasets, CWRU, Gear, Mixed, and MBHM, and the Stage 2 signal-side output achieves classification accuracies of 97.1%, 99.0%, 93.4%, and 96.3%, respectively. The report-generation branch provides a schema-constrained signal-to-language interface for maintenance-oriented reporting.

Keywords: fault diagnosis; large language model; vibration signal; probability-guided alignment; variational autoencoder; intelligent maintenance



Academic Editor: Juan Gabriel Avina-Cervantes

Received: 15 June 2026

Revised: 1 August 2026

Accepted: 3 August 2026

Published: 6 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

As complex industrial equipment is widely utilized in core fields such as manufacturing, energy and transportation, its operating condition directly determines equipment safety, production sustainability, and maintenance expenditure. Rolling bearings and gearboxes act as pivotal components of such equipment and are susceptible to multiple faults such as wear failure, fatigue cracks, and gear tooth breakage during long-term operational service [1,2]. Accordingly, reliable intelligent fault diagnosis using vibration signals has always been a key research topic in the fields of industrial condition monitoring and intelligent operation and maintenance. In recent years, convolutional neural networks and other deep temporal representation methods have greatly boosted fault identification performance [3,4]. Recent studies have applied prototype-attention domain adaptation to

explainable bearing fault diagnosis and fine-tuning strategies to mitigate distribution shifts in rotary-machine fault detection, highlighting the importance of interpretable and transferable representations under varying operating conditions [5,6]. However, most of these methods take classification accuracy as the core optimization goal, with model outputs merely limited to categorical labels or probability results. More importantly, constrained by classification loss dominance, the representations extracted by encoders tend to prioritize category boundary differentiation, while waveform structure, local impact characteristics, and background morphological information contained in original signals are likely to be weakened during feature compression. For downstream tasks that require signals to be mapped to large language models and the generation of diagnostic descriptions, such representations fail to provide an ideal semantic foundation.

The rapid development of large language models (LLMs) opens up a new research paradigm for fault diagnosis. Current studies on time series have attempted to map non-text sequential data into pre-trained language models, while research on time series foundation models is also rapidly developing; studies focusing on the design of interfaces between temporal foundation models and large language models are progressing from initial feasibility verification toward systematic modeling. Within the industrial fault diagnosis domain, existing mainstream research methods can be divided into three categories. The first category involves converting manual features such as statistical indicators and frequency-domain parameters into textual form, which are then input into LLMs for fault classification and diagnostic reasoning. The second category involves converting vibration signals into time-frequency graphs and achieving fault diagnosis using vision–language and multimodal models. Finally, the third category involves discretizing continuous vibration signals into token sequences to adapt to the inherent input modes of language models. These studies have validated the applicability of LLMs in fault diagnosis, yet they also expose prominent common defects: for example, text conversion methods can easily lose fine-grained structural signal information; image-based methods rely on additional time-frequency transformation and visual encoding branches; and discrete tokenization methods need to balance quantization error, sequence length, and training cost. It is clear that the core challenge is not simply importing vibration signals into LLMs, but constructing a logically coherent mapping path between signal representation, semantic intermediaries, and language model input.

In view of the above research limitations, in this paper, we propose PGA-LLM, a large language model fault diagnosis method based on probability-guided alignment. Targeting signal understanding, semantic mapping, and diagnostic generation tasks for complex industrial equipment, this method establishes an integrated modeling framework encompassing vibration signal input through to language-based diagnostic output. Specifically, an encoder with reconstruction constraints is first adopted to learn signal representations under joint reconstruction and classification objectives. The probability-guided alignment module is then utilized to project continuous vibration features into the embedding space of a large language model. Finally, Qwen2.5-1.5B combined with LoRA fine-tuning is employed for diagnostic text generation. The resulting framework combines signal-side fault prediction with a probability-guided interface for LLM-conditioned report generation.

The main contributions of this paper are summarized as follows:

1. A structure-preserving signal representation learning strategy for fault diagnosis is proposed, in which reconstruction and category supervision are combined to train the encoder for the subsequent interface.
2. A probability-guided alignment module is designed, combining fault-class probability guidance with a residual feature path, and projects their gated fusion into the embedding space of the large language model.

3. A progressive three-stage optimization framework is constructed which unifies encoder pre-training, alignment module learning, and language model adaptation into one training workflow. Qwen2.5-1.5B and LoRA are adapted after signal-side representation learning and interface optimization.

The rest of the paper is organized as follows. Section 2 reviews existing studies related to structure-preserving representation, probability-guided alignment, and language-driven diagnostic generation. Section 3 provides the proposed PGA-LLM framework, including the encoder structure, probability-guided alignment mechanism, and progressive training strategy. Section 4 presents the experimental setup and the quantitative results obtained on four datasets. Section 5 concludes the paper and provides potential future research directions.

2. Related Work

2.1. Vibration Signal Representation Learning for Diagnosis

Vibration signal representation learning is a fundamental research issue in intelligent fault diagnosis; a clear evolutionary trajectory has formed through existing studies in the field. In early research, convolutional neural networks were widely adopted to directly extract discriminative features from raw time-domain signals. For example, Chen et al. [7] explored feature reuse in rotating machinery diagnosis by using transferable convolutional networks; Zhang et al. [4] further investigated the training modes used for deep convolutional network diagnosis under noisy interference; and Liao et al. [8] enhanced nonlinear modeling capacity via a secondary convolutional structure. With the recent development of large-scale benchmark datasets and unified evaluation frameworks, the focus of fault diagnosis research has gradually shifted from single classification models to generalized representation learning, transfer learning, and pre-training paradigms [9,10]. Generalized time series learning methods can also provide continuous technical support for diagnostic tasks. Typical methods, including TimesNet [11], multi-task self-supervised representation learning [12], TS2Vec [13], and CRT [14], have improved performance in temporal dynamic modeling, multi-task pre-training, contrastive representation learning, and time-frequency self-supervised reconstruction. On the whole, existing studies have fully proven that high-quality signal representations can be effectively learned automatically from original vibration data.

Most existing methods still take closed-set classification as their core optimization objective, and their evaluation indicators mainly concentrate on classification accuracy. Consequently, these models tend to prioritize distinguishing between category boundaries while attaching insufficient importance for retaining waveform structures, impact characteristics, and background pattern information. Such representations can meet the demands of simple label classification tasks; however, when the model needs to perform further semantic mapping and diagnostic generation, the discriminative features obtained through simple compression cannot guarantee stable feature transmission across modules. In short, while current research has solved the problem of improving classification performance, there is still a lack of in-depth discussion on which signal representations are optimal for subsequent linguistic diagnosis.

Focusing on this limitation, variational modeling, autoencoder-based learning, and self-supervised methods offer valuable alternative directions. Zhang et al. [15] investigated uncertainty-aware and generalizable representation learning for rotating machinery fault diagnosis through a Bayesian variational transformer, showing the value of organized latent representations beyond single-run classification accuracy. In the field of fault diagnosis, multi-task representation learning [16] and self-supervised frameworks integrated with time–frequency consistency constraints [17] both demonstrate that relying

merely on classification boundaries is insufficient for maintaining model robustness under limited samples, strong noise, and cross-working-condition scenarios. It is also essential to preserve structural information and maintain well-organized feature representations. As such, this work provides a more in-depth focus on structure-preserving signal representation learning, and the proposed encoder is trained with reconstruction and classification constraints before subsequent interface optimization.

2.2. Semantic Alignment Methods for Mapping Vibration Signals to Language Models

In recent years, the incorporation of large language models into fault diagnosis has become increasingly popular in the field of industrial intelligent diagnosis. On a broader research scale, there have been systematic investigations into building effective interfaces between non-text sequential data and language models. One Fits All [18] demonstrates the possibility of adapting time series data to language models via pre-trained model reuse. Chronos [19], MOMENT [20], and AutoTimes [21] further illustrate that time series data can establish direct linkage with the language model paradigm through tokenization, foundation model pre-training, and autoregressive generation mechanisms. Though these works are mainly oriented toward prediction and general time series analysis, they reflect a critical trend: large language models are evolving from dedicated text processing tools into generalized semantic interfaces for sequential data.

Within industrial fault diagnosis research, existing studies predominantly center on transforming continuous industrial signals into language-model-readable input formats. Current mainstream methods can be classified into three categories. The first category is text-oriented: spectral characteristics, statistical metrics, and diagnostic rules are converted into natural language descriptions, which are subsequently processed by LLMs for fault classification and interpretation, as seen in the studies of Qaid et al. [22] and Zhang et al. [23]. The second category involves an image and multimodal learning pipeline: raw vibration signals are converted into time-frequency images, and diagnostic inference is achieved by combining visual encoders with large language models, with typical frameworks including DiagLLM [24] and FD-MVLLM [25]. The third category involves discrete modeling: continuous vibration segments are encoded into token sequences, which are directly embedded into the input sequence of a language model, as exemplified by the discretization strategy proposed in BearLLM. These efforts have verified the effectiveness of LLM-enabled fault diagnosis and greatly diversified the output forms of diagnostic results.

Nevertheless, when placed within the broader research context of multimodal alignment, existing fault diagnosis methods still exhibit noticeable research gaps. Representative cross-modal models such as CLIP [26], Flamingo [27], BLIP-2 [28], and LLaVA [29] have consistently proven that the quality of cross-modal mapping relies not only on encoder performance but also on the semantic arrangement of intermediate bridging layers. In comparison, although existing fault diagnosis methods have made substantial advances in textualization, imaging, and tokenization, most of them merely focus on enabling signal input into LLMs, with little attention paid to designing proper intermediate semantic representations before feature embedding. Textualization methods inevitably lose information during manual semantic abstraction; imaging-based approaches require extra time-frequency transformation and visual encoding modules; and discrete tokenization methods face inherent trade-offs between quantization error, sequence length, and training cost. To address this gap, we propose a compact interface that combines fault-class probability guidance with a residual feature path before projection to the LLM embedding space.

2.3. Progressive Optimization Strategy for Diagnostic Generation

As large language models are increasingly introduced into diagnostic tasks, the importance of training strategies has become increasingly prominent. In the field of multimodal learning, studies have proven that feature encoding, bridging mapping, and language generation cannot be arbitrarily combined in a casual manner. Existing multimodal research further indicates that the coordination between shared semantic space construction, bridging module pre-training, and fine-tuning instructions directly determines the final alignment accuracy and generation performance. Meanwhile, parameter-efficient fine-tuning methods (represented by LoRA [30]) have demonstrated that, during the adaptation of large language models, the selection of trainable parameters and training order also exert significant impacts on optimization stability and downstream task performance. Accordingly, for a diagnostic system integrating signal representation, semantic mapping, and text generation, the training pipeline itself is an indispensable part of methodological design.

This challenge is particularly pronounced in fault diagnosis applications. The vibration signal encoder, semantic alignment layer, and language generation module possess inconsistent optimization targets, and industrial diagnostic datasets are typically far smaller in scale than general vision–language corpora. Under such circumstances, fully coupling all optimization objectives from the start of training tends to lead to unstable intermediate representations, mutual gradient interference, and overfitting on the language modeling side. Recent reviews of PHM-oriented large models [31,32] have outlined the general development trend of combining large language models with diagnostic tasks. Even so, few existing studies have provided clear and implementable solutions for arranging representation learning, semantic alignment, and language generation in stages following inherent task logic.

Against the above background, we avoid treating the training process as a mere sequential splicing of independent modules and instead embed a progressive optimization strategy into the overall model design. This work adopts a three-stage optimization workflow: structure-preserving signal representation learning $\rightarrow$ probability-guided alignment $\rightarrow$ language-based diagnostic generation. This staged paradigm allows the model to build mapping relations step by step, ranging from low-level signal characterization to high-level semantic expression.

3. Methodology

3.1. Problem Definition

Consider a dataset of dual-channel vibration samples denoted as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$. Herein, each input sample $x_i \in \mathbb{R}^{2 \times L}$ is a vibration signal with a fixed length $L = 24,000$, where the two channels separately correspond to the monitored signal and the reference signal; $y_i \in \{0, 1, \dots, C - 1\}$ stands for the fault label, and C denotes the total number of fault categories contained in the dataset. Traditional fault diagnosis aims to learn a mapping function $g : x \rightarrow y$ that directly maps vibration signals to fault categories. However, when deployed for large language model-based diagnosis, the model is expected not only to identify fault categories, but also to produce expressive intermediate representations for operating state interpretation, judgment reasoning, and diagnostic text generation. Accordingly, this work defines the research problem as constructing a coherent diagnostic pipeline that integrates vibration signal representation, diagnostic semantic alignment, and linguistic output, while maintaining satisfactory classification performance.

Current deep learning fault diagnosis methods predominantly prioritize category discrimination but may not explicitly constrain structural information such as waveform morphology and local impact characteristics. To tackle this limitation, this work develops a multi-scale convolution-based feature-embedding approach combined with

structure-preserving pre-training. By introducing reconstruction constraints in addition to classification supervision, the Stage 1 objective balances discriminative category learning with signal reconstruction before subsequent interface optimization.

Currently, semantically clear mapping mechanisms used to connect vibration features with the embedding spaces of large language models are lacking. Existing solutions, including textualization, time-frequency imaging, and discrete tokenization, cannot simultaneously satisfy the requirements of structural information preservation, semantic explicitness, and low training cost. To fill this gap, this paper presents a probability-guided alignment module which uses fault-category probability distribution for label-indexed guidance and a complementary feature path before projecting their fusion into the embedding space of a large language model. This design defines an explicit label-indexed interface between vibration features and LLM embeddings.

3.2. PGA-LLM Method

Motivated by the above research challenges, we propose the PGA-LLM framework for complex equipment fault diagnosis, with its overall architecture depicted in Figure 1. The following subsections elaborate on the core modules of the proposed method in detail, covering multi-scale convolutional feature embedding, structure-preserving pre-training, probability-guided alignment, and LLM-based diagnostic generation.

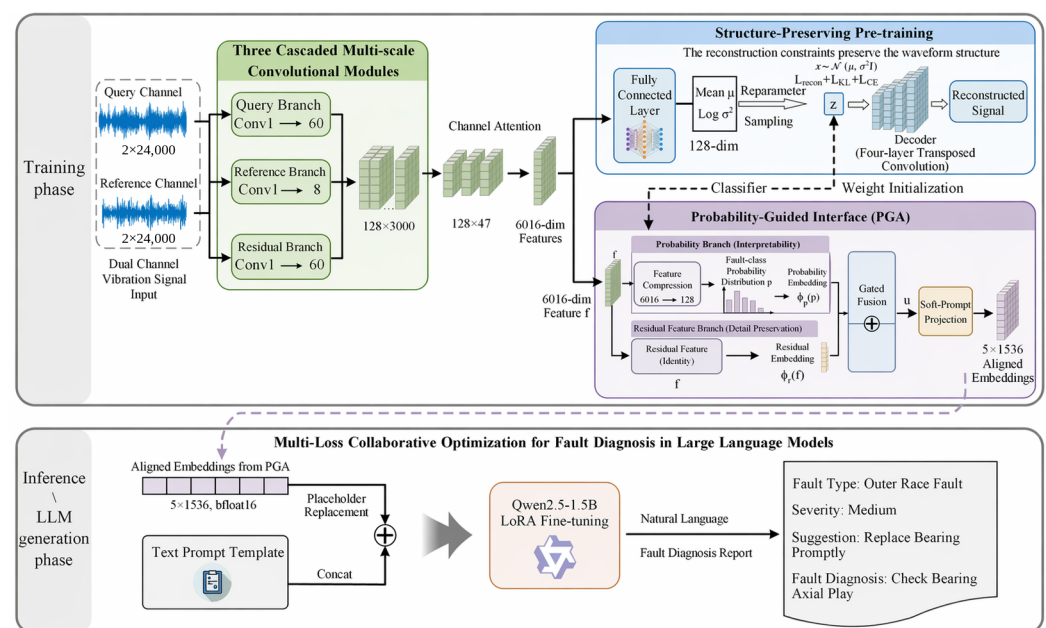


Figure 1. Overall framework of the PGA-LLM method. Solid arrows denote data flow, dashed arrows denote initialization or stage transition, and colored panels distinguish the encoder/VAE, PGA, and LLM-generation modules.

3.2.1. Multi-Scale Convolution-Based Feature Embedding

Within the multi-scale convolution-based feature-embedding framework, the input signal is first fed into three parallel branches: the query branch, the reference branch, and the residual branch. The query branch is responsible for extracting response features of the signal to be diagnosed; the reference branch provides a stable baseline operating state; the residual branch characterizes the state deviation by capturing the difference between the diagnosed signal and the reference signal. The three branches describe vibration signals respectively from three perspectives: diagnostic response, reference baseline, and state offset. Their outputs are concatenated along the channel dimension to form a unified feature input for subsequent multi-scale convolution processing.

The first layer of each branch adopts a wide convolution kernel for initial feature extraction:

$$h_{\text{branch}} = \text{LeakyReLU}(\text{BN}(\text{Conv1d}(x_{\text{branch}}))) \quad (1)$$

The input of the residual branch is defined as

$$x_{\text{res}} = x_{\text{query}} - x_{\text{ref}} \quad (2)$$

This branch does not aim to capture high-amplitude responses directly; instead, it focuses on characterizing the variation pattern that deviates from the reference state. Meanwhile, the adoption of a wide convolutional kernel ($k = 8$) together with stride downsampling ($s = 8$) enables the first layer to cover a long local period, making it more sensitive to impact-related faults and periodically modulated signals. The convolutional configuration of both the query branch and the residual branch is set as $\text{Conv1d}(1 \rightarrow 60, k = 8, s = 8)$, while the reference branch adopts $\text{Conv1d}(1 \rightarrow 8, k = 8, s = 8)$. The outputs of the three branches are concatenated along the channel dimension to yield a feature map with dimensions of 128×3000 .

To further process fault patterns at different temporal scales and mitigate the scale bias introduced by a single convolution kernel, we designed the multi-scale convolution module illustrated in Figure 2. The module consists of four parallel convolution branches with kernel sizes of 1, 3, 5, and 7, which extract local patterns under different receptive fields at the same network layer.

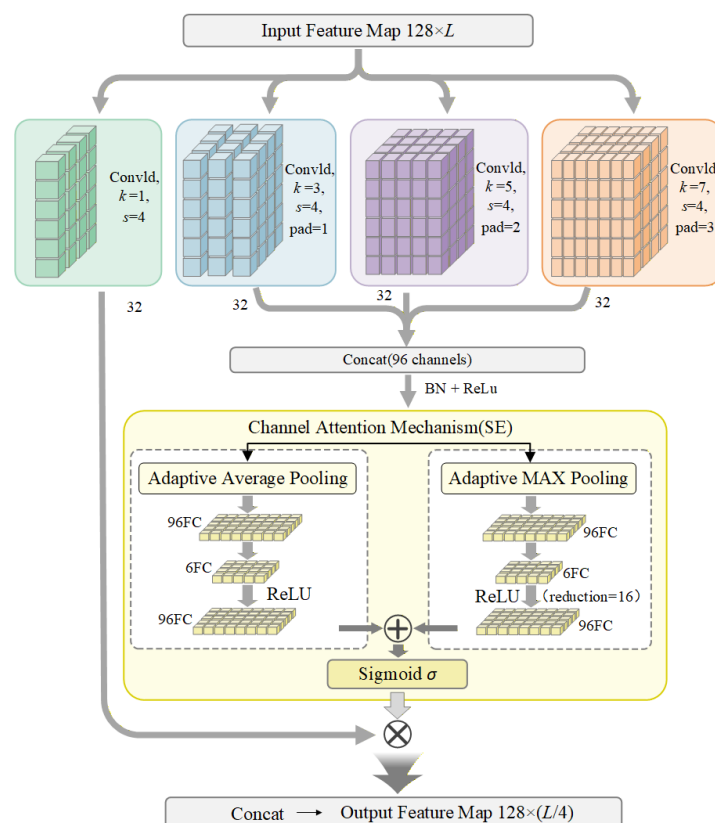


Figure 2. Multi-scale convolutional feature-embedding module.

After obtaining the initial feature map, the multi-scale convolution module feeds it into four parallel convolutional branches. Smaller convolution kernels are adept at capturing

short-term variations, while larger kernels can better cover long-range local contextual information:

$$h_k = \text{Conv1d}(h_{\text{in}}, k, s=4), \quad k \in \{1, 3, 5, 7\} \quad (3)$$

Here, k denotes the temporal convolution kernel size. The $k = 1$ branch performs point-wise channel mixing without enlarging the temporal receptive field, thereby preserving instantaneous response information and providing a local channel-recombination path. In contrast, the $k = 3, 5, 7$ branches capture progressively wider temporal patterns. All four parallel convolutional branches adopt the same downsampling stride, which synchronously compresses the temporal dimension within each multi-scale module. Feature dimensions are kept controllable via branch concatenation and channel reorganization. This design enables parallel extraction of local patterns under different receptive fields at the same network layer, and realizes cross-scale feature fusion gradually in subsequent modules.

The outputs of the three branches where $k = 3, 5, 7$ are concatenated and then weighted by a channel attention mechanism:

$$h_{\text{att}} = \sigma(\text{SE}(\text{AvgPool}(h_{3,5,7})) + \text{SE}(\text{MaxPool}(h_{3,5,7}))) \odot h_{3,5,7} \quad (4)$$

where $\text{SE}(\cdot)$ denotes the squeeze-and-excitation network, σ represents the Sigmoid function, and $\odot$ denotes element-wise multiplication. Finally, the feature of the $k = 1$ branch is concatenated with the attention-weighted features while maintaining a constant channel number. After passing through three cascaded multi-scale modules, the feature map is compressed to 128×47 and then flattened into a vector $f \in \mathbb{R}^{6016}$. This compact representation is used by the VAE and PGA modules.

3.2.2. Structure-Preserving Pre-Training

After completing convolutional feature extraction, the encoder needs to construct a latent space that adapts to subsequent semantic alignment. This work adopts the VAE framework to implement structure-preserving pre-training, so that latent variables are jointly regulated by reconstruction constraints, distribution constraints, and category supervision. The reconstruction path explicitly forces the model to recover raw vibration signals from latent variables. By following the mapping relationship $x \rightarrow z \rightarrow \hat{x}$ and the approximation constraint $\hat{x} \approx x$, waveform morphology, local impact features, and background structural information are integrated into the representation learning process. More precisely, the Stage 1 path is $x \rightarrow f \rightarrow (\mu, \log \sigma^2) \rightarrow z \rightarrow \hat{x}$.

Concretely, based on the feature f , the encoder estimates the mean $\mu \in \mathbb{R}^{128}$ and logarithmic variance $\log \sigma^2 \in \mathbb{R}^{128}$ of the latent distribution, while outputting the classification logits $z_{\text{cls}} \in \mathbb{R}^C$. The log variance used in sampling and KL computation is clipped element-wise as $\tilde{\ell} = \text{clip}(\log \sigma^2, -10, 10)$. The latent variable is obtained using the following reparameterization trick:

$$z = \mu + \epsilon \odot \exp(0.5 \cdot \tilde{\ell}), \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (5)$$

The decoder then reconstructs the raw vibration signal $\hat{x} \in \mathbb{R}^{2 \times 24000}$ from the latent variable z . This reconstruction branch is not used for final inference output; instead, it acts as an auxiliary constraint in the pre-training phase.

The overall loss function for structure-preserving pre-training is composed of reconstruction loss, KL divergence, and classification loss.

$$\mathcal{L}_{\text{VAE}} = \lambda_r \cdot \mathcal{L}_{\text{recon}} + \lambda_k \cdot \mathcal{L}_{\text{KL}} + \lambda_c \cdot \mathcal{L}_{\text{CE}} \quad (6)$$

where

$$\mathcal{L}_{\text{recon}} = \text{MSE}(\hat{x}, x) + 0.5 \cdot \text{L1}(\hat{x}, x) \quad (7)$$

$$\mathcal{L}_{\text{KL}} = \text{clip} \left(-\frac{1}{2Bd_z} \sum_{i=1}^B \sum_{j=1}^{d_z} (1 + \tilde{\ell}_{i,j} - \mu_{i,j}^2 - \exp(\tilde{\ell}_{i,j})), 0, 100 \right), \quad d_z = 128 \quad (8)$$

$$\mathcal{L}_{\text{CE}} = -\frac{1}{B} \sum_{i=1}^B \sum_{c=0}^{C-1} \mathbf{1}(y_i = c) \log \frac{\exp(z_{\text{cls},i,c})}{\sum_{m=0}^{C-1} \exp(z_{\text{cls},i,m})} \quad (9)$$

In the formula, B denotes the batch size, and $z_{\text{cls},i,c}$ refers to the classification logit of the i -th sample on the c fault category. The default weight hyperparameters are set to $\lambda_r = 1.0$, $\lambda_k = 0.1$, and $\lambda_c = 0.5$. The reconstruction term combines MSE, which penalizes overall waveform discrepancy, with L1 loss, which emphasizes local deviations. Together with the KL and classification terms, this objective is intended to balance reconstruction and category discrimination during Stage 1. $\mathcal{L}_{\text{KL}}$ regularizes the latent distribution to approach the standard normal distribution, maintaining the smoothness of the latent space. The cross-entropy classification loss $\mathcal{L}_{\text{CE}}$ maintains basic category discrimination capability, keeping representation learning from leaning excessively toward generative reconstruction.

At the initial stage of training, the KL term and logarithmic variance are prone to drastic numerical fluctuations. To avoid these terms dominating the optimization process in a short amount of time, we adopt truncation processing for KL divergence and logarithmic variance. The log variance and scalar KL term are clipped to $[-10, 10]$ and $[0, 100]$, respectively, to stabilize optimization.

3.2.3. Probability-Guided Alignment

The probability-guided alignment (PGA) interface maps the convolutional encoder output to the LLM embedding space through complementary probability-guidance and residual-feature paths. The flattened encoder feature f feeds both paths: the fault-class probability vector provides label-indexed guidance, while the residual branch retains additional signal information. Their gated fusion is projected to soft prompts. The VAE variables μ and z are used only for Stage 1 reconstruction. The interface-level contribution is the joint use of these complementary paths before soft-prompt projection, while VAE pre-training, LoRA adaptation, and continuous embedding injection serve as established components of the workflow. Figure 3 illustrates the PGA interface.

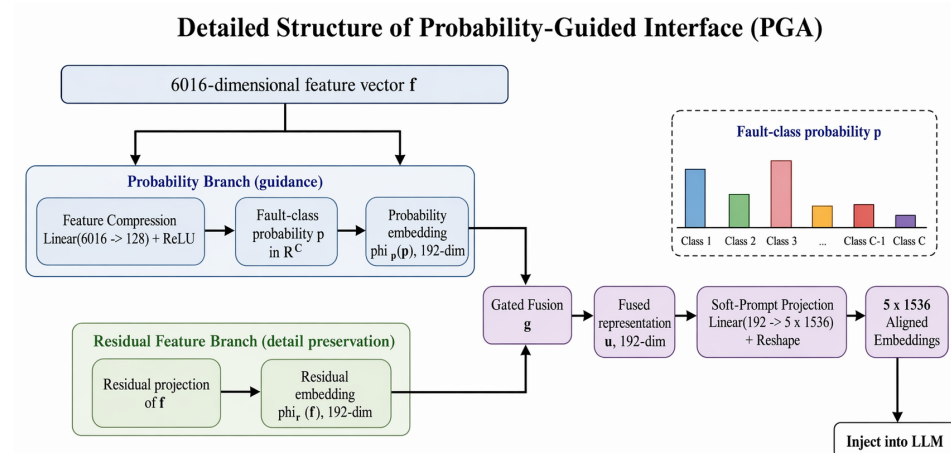


Figure 3. Structure of the probability-guided interface (PGA).

The probability branch first compresses the 6016-dimensional encoded feature into a 128-dimensional representation:

$$h_1 = \text{ReLU}(W_1 f + b_1), \quad W_1 \in \mathbb{R}^{128 \times 6016} \quad (10)$$

It is then mapped to a temperature-scaled C -dimensional fault-class probability distribution:

$$p = \text{softmax}((W_2 h_1 + b_2)/\tau), \quad W_2 \in \mathbb{R}^{C \times 128} \quad (11)$$

Here, $p \in \Delta^{C-1}$ denotes a probability vector on the simplex. The temperature τ is fitted on the validation partition, and the probability branch is initialized from the classifier obtained during structure-preserving pre-training.

The probability path supplies label-indexed guidance to the PGA interface. In subsequent evaluations, calibration and the sensitivity of reports to probability errors will be a suitable area for further research.

To retain signal information not represented by p alone, PGA forms a 192-dimensional probability embedding $\phi_p(p)$ and a 192-dimensional residual feature embedding $\phi_r(f)$. A learned gate fuses the two representations:

$$g = \sigma(h_g([\phi_p(p); \phi_r(f)])), \quad u = g \odot \phi_p(p) + (1 - g) \odot \phi_r(f). \quad (12)$$

The fused representation is projected to the continuous soft prompts used by the LLM:

$$E_{\text{soft}} = \text{reshape}(W_3 u + b_3, [D_{\text{desc}}, D_{\text{LLM}}]), \quad W_3 \in \mathbb{R}^{(D_{\text{desc}} \cdot D_{\text{LLM}}) \times 192}. \quad (13)$$

The final output $E_{\text{soft}} \in \mathbb{R}^{5 \times 1536}$ contains five continuous description embeddings, where $D_{\text{desc}} = 5$ is the prompt-side embedding length and $D_{\text{LLM}} = 1536$ is the hidden size of Qwen2.5-1.5B.

3.2.4. Signal-to-LLM Embedding Injection

The proposed framework does not discretize the vibration waveform into ordinary text tokens. Instead, it injects the continuous signal-side representation into the LLM input embedding sequence. A special marker `<|signal|>` is placed in the user prompt as a locating placeholder. After tokenization, this marker may correspond to a short token span rather than a single token. Let i denote the start position of this span and k denote its token length. The original token embeddings over this span are replaced by the PGA soft prompts E_{soft} :

$$E_{\text{input}} = [e_1, \dots, e_{i-1}, E_{\text{soft}}, e_{i+k}, \dots, e_T]. \quad (14)$$

The attention mask is then constructed according to the length of the resulting embedding sequence. Through this operation, the LLM receives continuous soft prompts formed by probability guidance and residual signal features, while the textual prompt continues to provide task constraints, candidate label information, and output format instructions. Figure 4 illustrates this embedding replacement mechanism.

3.2.5. LLM Fault Diagnosis Based on Multi-Loss Collaborative Optimization

PGA-LLM maps the soft prompts to structured diagnostic text through progressive multi-loss optimization. Structure-preserving pre-training constrains signal representation learning; Stage 2 directly trains the probability-guided interface; and Stage 3 jointly adapts the active interface and LoRA parameters for diagnostic text generation. The stages are scheduled sequentially, while the language-modeling loss in Stage 3 retains a differentiable path to the active PGA interface.

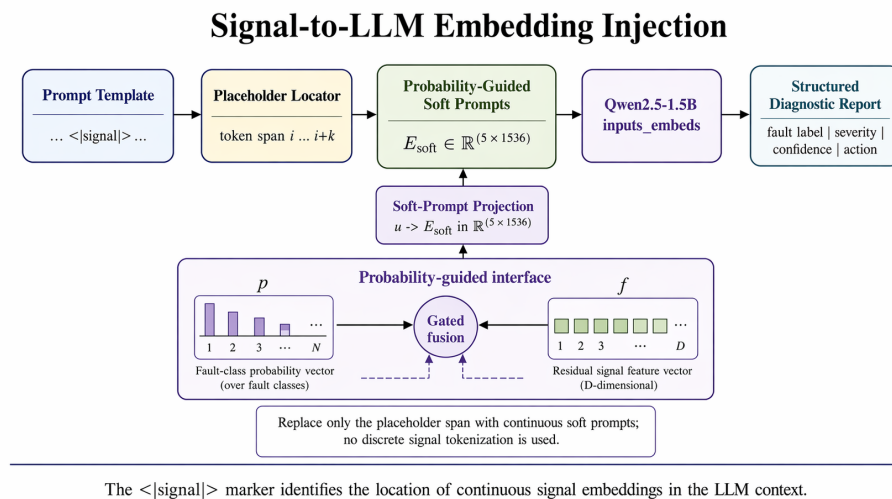


Figure 4. Signal-to-LLM embedding injection. The marker `<|signal|>` is used only to locate the token span to be replaced; the vibration signal is imported as probability-guided continuous soft prompts rather than as discrete signal tokens.

In the diagnostic generation phase, the continuous soft prompts output by PGA act as signal-side diagnostic conditions through the embedding injection mechanism described in Section 3.2.4. Meanwhile, textual prompts delineate task constraints and provide semantic information of candidate faults, and Qwen2.5-1.5B completes diagnostic report generation within the integrated context. LoRA is used for parameter-efficient language-model adaptation rather than retraining the full language-model backbone.

Traditional methods usually follow a two-step pipeline: first classify the sample, then generate descriptive text solely from the predicted label. By contrast, the proposed module does not compress vibration signals into a single discrete label for LLM paraphrasing. The LLM receives continuous soft prompts produced from probability guidance and residual signal features rather than a single categorical output.

Prompt templates are mainly used to define task boundaries and embed domain prior knowledge. Given that different datasets differ in label systems, fault naming rules and report granularity, this framework keeps the signal embedding injection interface fixed and only adjusts candidate label sets, professional terms, and output specifications in textual prompts. In this way, cross-dataset migration requires no modification to the signal processing architecture, while dataset-specific diagnostic requirements can be explicitly configured at the prompt level. Taking the CWRU dataset as an example, we obtain the following:

System Prompt: You are an industrial diagnostic reporting assistant. Return exactly one JSON object with the string fields `fault_label`, `severity`, `confidence_or_uncertainty`, and `maintenance_action`. Do not add explanation, signal mechanisms, frequency analysis, or text outside JSON. The confidence field is a fixed signal-side posterior-source identifier.

User Prompt: `</signal|>` The prompt also supplies a fixed CWRU candidate glossary that enumerates every permitted label-to-fault, severity, and maintenance-action mapping, but does not disclose the sample prediction or ground-truth label. Produce the constrained diagnostic report for this vibration input.

For the auditable report-generation implementation, each target is a canonical JSON object constructed deterministically from the record's ground-truth class using a fixed label-to-field rule table. CWRU therefore has ten schema-level target reports, with one report per class. This setup defines constrained structured rendering and

label–field consistency targets; the full prompt contract and rule table are provided in the Supplementary Materials.

To clearly demonstrate the three-stage iterative training workflow of PGA-LLM, structure-preserving pre-training, probability-guided alignment, and LLM fine-tuning were integrated, as shown in the schematic diagram presented in Figure 5. The overall optimization process is implemented sequentially around signal representation learning, fault category probability intermediate mapping, and diagnostic text generation. The training objectives and loss functions of each stage are specified in the following content.

PGA-LLM: Three-Stage Training Framework for Workflow, Objective Functions, and Parameter Update Boundaries

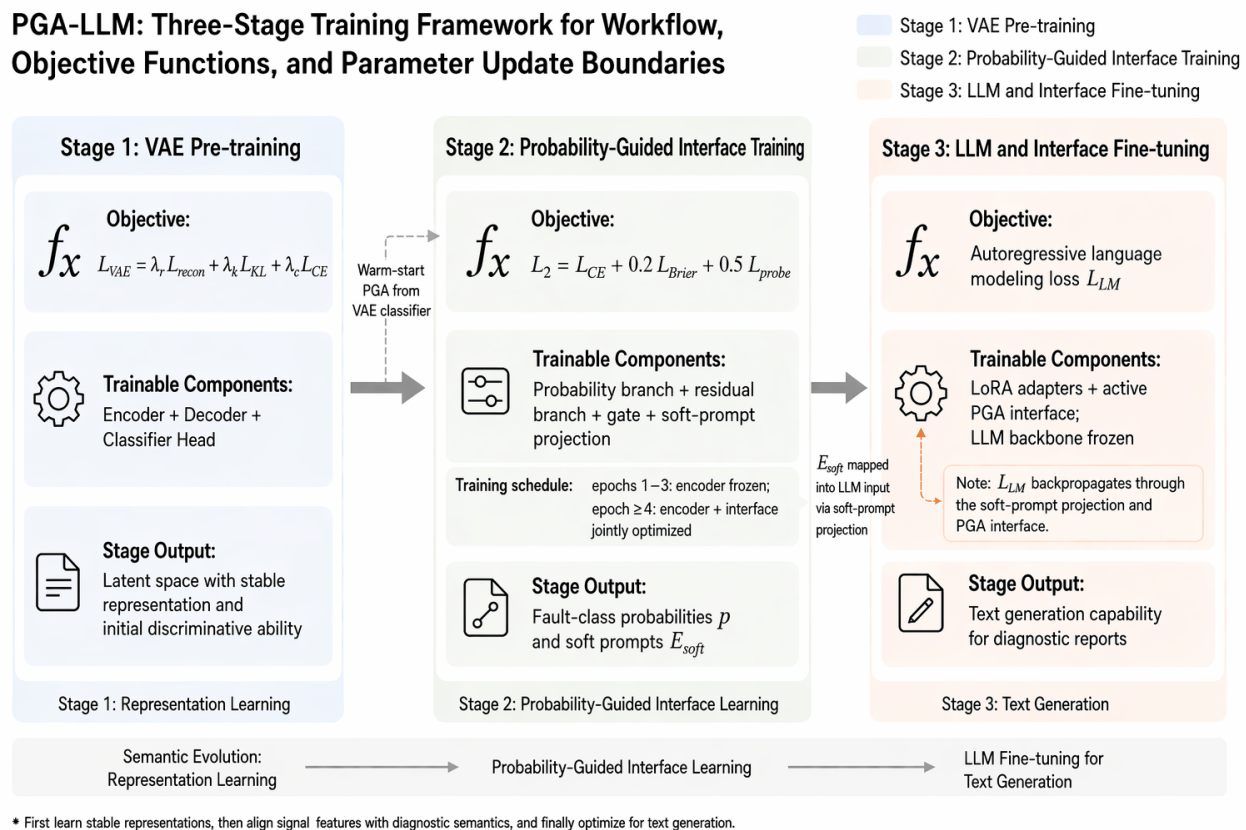


Figure 5. Three-stage training strategy for PGA-LLM. The asterisk denotes the sequential training order from representation learning to interface learning and then text generation.

The primary goal of the first stage is to acquire signal representations. The encoder and decoder are jointly optimized via the composite loss $\mathcal{L}_{VAE}$ (Equation (6)). This stage applies reconstruction, distribution, and classification constraints before the probability-guided interface is optimized.

The second stage trains the probability-guided interface. The probability branch, residual feature branch, gating module, and soft-prompt projection are trainable throughout this stage. The encoder is frozen during the first three epochs; from the fourth epoch onward, the encoder and PGA interface are jointly optimized. The Stage 2 objective is

$$\mathcal{L}_2 = \mathcal{L}_{CE} + 0.2\mathcal{L}_{Brier} + 0.5\mathcal{L}_{probe}. \quad (15)$$

Here, $\mathcal{L}_{CE}$ supervises class prediction, $\mathcal{L}_{Brier}$ regularizes the probability output, and $\mathcal{L}_{probe}$ is the classification loss of an auxiliary probe applied to E_{soft} . The probe loss provides a direct supervised gradient path to the final soft-prompt projection in Stage 2. The third stage performs diagnostic adaptation of Qwen2.5-1.5B. The language-model backbone remains

frozen, while LoRA parameters and the active PGA interface, including the soft-prompt projection, are jointly optimized by the autoregressive language-modeling loss:

$$\mathcal{L}_{\text{LM}} = - \sum_{t=1}^T \log p_{\theta}(w_t \mid w_{<t}, \mathbf{E}_{\text{soft}}) \quad (16)$$

In the formula, w_t is the t -th token in the target diagnostic text and $\mathbf{E}_{\text{soft}}$ denotes the PGA soft prompts. The final projection therefore receives gradients in both Stage 2 and Stage 3.

4. Experiments

This chapter elaborates on the experimental results obtained with the PGA-LLM model. It begins with a description of the experimental datasets, data partitioning strategies, and hyperparameter configurations, followed by the presentation of core results across different datasets. Model performance is further analyzed from multiple perspectives, including category-specific diagnostic outcomes, intermediate representation visualization, ablation experiments, and loss convergence trends.

4.1. Dataset Introduction and Partitioning

To evaluate PGA-LLM under the reported source-series and condition-group allocations, experiments are carried out on four fault diagnosis datasets covering bearing faults, gearbox faults, and compound-fault scenarios. The basic information of all datasets is summarized in Table 1.

Table 1. Source series and constructed dual-channel instances used in this study.

Dataset	No. Classes	Source Series/Files	Query-Ref. Instances	Format	Fault Type
CWRU [33]	10	40	456	JSON	Bearing faults: normal + inner race/rolling element/outer race faults $\times$ 3 severity levels
Gear	2	48	1524	JSON	Gearbox faults: normal/broken tooth
Mixed	15	64	2488	JSON	5 bearing fault classes + 4 gear fault classes + 6 compound fault classes
MBHM [34]	10	135,516 files (122,792 eligible)	368,376	HDF5	Multi-condition bearing faults: normal + inner race/rolling element/outer race faults $\times$ 3 severity levels

The four datasets span from small-scale, few-category scenarios to large-scale, multi-condition environments. The two-class Gear dataset contains varying operating conditions. The Mixed dataset includes bearing, gear, and compound faults, while MBHM provides a larger collection with diverse operating conditions. All signals are uniformly formatted as dual-channel inputs, consisting of the signal under diagnosis and a reference signal, with a sampling length of 24,000 points.

Table 1 distinguishes the source data from the constructed dual-channel learning instances. For CWRU and Mixed, 24,000-point query windows were extracted with a stride of 12,000 points (50% overlap), while Gear used a 2400-point stride (90% overlap). Each query window was paired with a fixed reference: CWRU fault windows use the same-index normal window at the corresponding RPM/load; Gear broken-tooth windows use the same-index healthy window at the corresponding load and sensor channel; and Mixed fault windows use a same-speed normal window, with the bearing-normal record used as the available proxy for mixed bearing-gear faults. Normal queries use the following normal window as their reference, and the reference index is clamped when a reference record is shorter. Thus, this study evaluates reference-assisted diagnosis, where a validation or test query may use a healthy reference from the same operating condition but no target

fault label is provided through the reference. In MBHM, the original metadata contain 135,516 files. Only the 122,792 files whose `condition_id` has at least one normal file are eligible; each is paired with three fixed-seed draws with replacement from normal files sharing that `condition_id`, producing $122,792 \times 3 = 368,376$ query-reference instances.

The choice of data partitioning strategy is critical to the reliability of fault diagnosis evaluations. The primary protocol is denoted as StrictGroup. For CWRU, Gear, and Mixed, all query windows sharing a source-series key are assigned to one subset: the original source file under a condition, the load/channel/health-state series, and the source subset/speed/fault-label series, respectively. For MBHM, all constructed instances with the same `condition_id` are assigned together. The target train/validation/test allocation is 70%/15%/15%, but it is not an exact realized sample ratio. The fixed seeds 42, 43, and 44 shuffle the indivisible groups; JSON-dataset groups are assigned within labels toward the target while retaining the validation and test groups, whereas MBHM-condition groups are assigned toward the target by their paired-instance counts. StrictGroup separates source-series groups while allowing a healthy reference from the matched operating condition; it is therefore a source-series grouping protocol rather than a condition-held-out evaluation. A visual comparison of the split families is shown in Figure 6.



Figure 6. Schematic of the allocation protocols. Bars show the target 70%/15%/15% allocation only; realized StrictGroup shares are reported in Table 2.

In addition to StrictGroup, CWRU, Gear, and Mixed are evaluated with condition-wise continuous blocks, random source-series groups, and window-level random pairs. MBHM has no continuous-block result because its metadata provide file identifiers rather than continuous acquisition positions. For the JSON datasets, CondContig sorts windows within each source series and assigns contiguous blocks to different subsets, whereas RecordRandom assigns whole source-series groups randomly; WindowRandom allocates individual constructed pairs and can place overlapping windows from one source series in different subsets. Together, these protocols characterize sensitivity to the allocation unit.

Table 2 reports the realized allocation averaged over the three fixed seeds. Per-partition class counts are emitted by the released split-summary routine for every seed. The deviations from 70%/15%/15% are caused by the indivisibility and unequal sizes of source-series groups, not by rounding: for example, the four source groups available for each CWRU fault label impose an approximate 2/1/1 train/validation/test allocation.

Table 2. StrictGroup allocation statistics averaged over seeds 42, 43, and 44.

Dataset	Group Unit	Instances (Train/Val/Test)	Groups (Train/Val/Test)	Share (%) (Train/Val/Test)	Min. Test	Max Imbalance
CWRU	source series	231.3/112.3/112.3	20.0/10.0/10.0	50.7/24.6/24.6	9	4.222
Gear	source series	1090.7/256.7/176.7	34.3/8.0/5.7	71.6/16.8/11.6	65	1.477
Mixed	source series	1320.0/584.0/584.0	34.0/15.0/15.0	53.1/23.5/23.5	38	1.026
MBHM	condition ID	265.4 k/57.9 k/45.0 k	180.0/40.3/37.7	72.1/15.7/12.2	3	10,447

The MBHM allocation has a maximum test imbalance ratio of 10,447 and a minimum class support of three instances. Its aggregate accuracy is therefore reported together with the corresponding class-support distribution.

4.2. Hyperparameter Settings

The comparison methods include the information-matched non-LLM BearLLM_FCN control, QCNN, TCNN, and two classic convolutional fault diagnosis baselines, covering a two-channel non-LLM FCN control, representative temporal convolutional networks, and common end-to-end diagnostic models. The baseline models are trained with three different random seeds (42/43/44), and results are reported as mean $\pm$ standard deviation. For PGA-LLM and its key ablation configurations, the final performance is reported as the average of multiple independent runs.

All partitioning protocols are executed under the same training budget (epochs = 80; the MBHM baseline uses 20 epochs). The canonical single-channel baselines (WDCNN/QCNN/TCNN/BearingFM) uniformly use only the diagnostic signal channel, with signals resampled according to each model's input requirements. BearLLM_FCN is evaluated with the same diagnostic/reference inputs and derived residual as PGA-LLM, whereas the other listed baselines retain their canonical single-channel configurations.

The main hyperparameter settings are presented in Table 3.

Table 3. Main hyperparameters.

Hyperparameter	VAE Pre-Training	Adapter Training
Optimizer	AdamW	AdamW
Learning rate	1×10^{-3}	1×10^{-4}
Weight decay	1×10^{-4}	1×10^{-4}
Batch size	32 (128 for MBHM)	32 (128 for MBHM)
Training epochs	80	80
LR scheduler	ReduceLROnPlateau	ReduceLROnPlateau
Gradient clipping	1.0	1.0
Frozen encoder epochs	–	3
Latent dimension	128	128
Description token count D_{desc}	–	5
LLM hidden dimension D_{LLM}	–	1536

Table 4 summarizes the focused CWRU StrictGroup subset used for the Stage 3 report-generation audit.

Table 4. Audited CWRU Stage 3 report-generation configuration.

Item	Configuration
Protocol and records	CWRU StrictGroup audit (seed 42); 180/90 training/validation records; 18/9 records per class.
Backbone and LoRA	Frozen Qwen2.5-1.5B (bf16); LoRA $r = 8$, $\alpha = 16$, dropout 0.05 on attention and MLP projections.
Optimization	AdamW; LoRA/PGA learning rates $2 \times 10^{-5} / 5 \times 10^{-5}$; weight decay 10^{-4} , clipping 1.0; batch size 1, four-step accumulation, 12 epochs.
Decoding and selection	Greedy decoding, at most 128 new tokens; EOS or <code>< im_end ></code> termination; checkpoint with minimum validation language-modeling loss.
Parameter accounting	1.565 B instantiated parameters; 11.946 M trainable parameters (active PGA bridge and LoRA).
Environment and duration	Two RTX A5000 GPUs (24,564 MiB each); CUDA 12.4, PyTorch 2.6.0, Transformers 4.57.3; PEFT 0.18.0, Accelerate 1.12.0; Stage 3 loop: 539.96 s after initialization.

4.3. Experimental Results

Classification and report generation are evaluated as separate outputs. Values labeled “PGA-LLM (signal-side)” or “PGA-LLM (Adapter)” are calculated from $\arg \max_c p_c$, where p is the Stage 2 PGA probability output. Other baselines and ablation rows use the classification outputs of their stated signal-side configurations. The report-generation route produces schema-constrained text from the signal-conditioned soft prompts, and LoRA is optimized for this route rather than for the signal-side classification values. The benchmark datasets provide class labels rather than report-level reference annotations.

Table 5 reports the aggregate signal-side test accuracy under StrictGroup. The PGA-LLM (signal-side) row is calculated from the Stage 2 PGA probability output, while the other rows use the outputs of their stated signal-side configurations. Boldface marks the highest reported value in each dataset. The baseline rows summarize seeds 42, 43, and 44, whereas the PGA rows summarize ten runs.

Table 5. Signal-side test accuracy under the StrictGroup protocol (%).

Method	CWRU (10 Classes)	Gear (2 Classes)	Mixed (15 Classes)	MBHM (10 Classes)
BearLLM_FCN	96.92 $\pm$ 2.21	91.87 $\pm$ 5.79	55.94 $\pm$ 3.31	93.22 $\pm$ 2.64
WDCNN	61.20 $\pm$ 9.80	83.80 $\pm$ 7.05	40.24 $\pm$ 6.32	77.98 $\pm$ 2.76
QCNN	66.81 $\pm$ 10.31	91.69 $\pm$ 5.54	32.71 $\pm$ 5.83	78.31 $\pm$ 5.54
TCNN	54.42 $\pm$ 5.72	73.40 $\pm$ 12.51	35.56 $\pm$ 4.99	79.59 $\pm$ 3.94
BearingFM	89.74 $\pm$ 5.86	97.93 $\pm$ 2.93	64.90 $\pm$ 9.81	84.61 $\pm$ 3.22
PGA-LLM (w/o PGA)	94.7	97.4	90.2	93.9
PGA-LLM (signal-side)	97.1	99.0	93.4	96.3

Note: Bold values indicate the highest reported result in each dataset.

Table 6 reports PGA-LLM (Adapter) accuracy under alternative allocation protocols. The MBHM metadata provide file identifiers rather than continuous acquisition positions, so the CondContig entry is unavailable. WindowRandom yields higher values than StrictGroup across the four datasets, consistent with the placement of overlapping windows from one source series in different subsets.

Table 6. Signal-side test accuracy of PGA-LLM (Adapter) under different partitioning protocols (%).

Dataset	StrictGroup	CondContig	RecordRandom	WindowRandom
CWRU	97.10	98.40	97.80	99.20
Gear	99.00	99.20	99.10	100.00
Mixed	93.40	95.60	93.40	98.90
MBHM	96.30	–	98.60	99.30

Note: Bold values indicate the highest result within each dataset row.

To provide a comprehensive comparison of all methods across different partitioning protocols, Tables 7–10 present the full performance results for each of the four datasets.

The protocol-specific tables list point estimates from the named allocation runs, whereas Table 5 summarizes the runs specified above.

Table 7. CWRU: signal-side test accuracy under different partitioning protocols (%).

Method	StrictGroup	CondContig	RecordRandom	WindowRandom
PGA-LLM (VAE)	94.70	96.50	95.20	98.40
PGA-LLM (Adapter)	97.10	98.40	97.80	99.20
BearLLM_FCN	95.80	100.00	95.80	100.00
BearingFM	91.60	98.97	91.60	100.00
WDCNN	66.39	90.72	66.39	95.95
QCNN	76.47	93.81	76.47	93.24
TCNN	59.66	90.72	59.66	93.24

Note: Bold values indicate the highest reported result within each partitioning-protocol column.

Table 8. Gear: signal-side test accuracy under different partitioning protocols (%).

Method	StrictGroup	CondContig	RecordRandom	WindowRandom
PGA-LLM (VAE)	97.40	98.30	97.80	99.50
PGA-LLM (Adapter)	99.00	99.20	99.10	100.00
BearLLM_FCN	88.65	96.64	88.65	100.00
BearingFM	100.00	98.74	100.00	100.00
WDCNN	82.70	96.64	82.70	98.68
QCNN	88.65	94.54	88.65	100.00
TCNN	69.73	92.86	69.73	100.00

Note: Bold values indicate the highest reported result within each partitioning-protocol column.

Table 9. Mixed: signal-side test accuracy under different partitioning protocols (%).

Method	StrictGroup	CondContig	RecordRandom	WindowRandom
PGA-LLM (VAE)	90.20	93.10	90.20	96.80
PGA-LLM (Adapter)	93.40	95.60	93.40	98.90
BearLLM_FCN	53.42	90.18	53.42	94.49
BearingFM	57.53	94.20	57.53	98.43
WDCNN	46.23	75.22	46.23	85.56
QCNN	36.30	76.12	36.30	88.98
TCNN	29.11	75.89	29.11	87.14

Note: Bold values indicate the highest reported result within each partitioning-protocol column.

Table 10. MBHM: signal-side test accuracy under different partitioning protocols (%).

Method	StrictGroup	CondContig	RecordRandom	WindowRandom
PGA-LLM (VAE)	93.90	–	97.20	98.70
PGA-LLM (Adapter)	96.30	–	98.60	99.30
BearLLM_FCN	92.81	–	99.45	99.88
BearingFM	85.90	–	95.16	99.62
WDCNN	74.16	–	90.03	95.55
QCNN	71.55	–	88.81	93.44
TCNN	82.97	–	90.07	90.34

Note: Bold values indicate the highest reported result within each partitioning-protocol column.

On CWRU, PGA-LLM (Adapter) reaches 97.10% under StrictGroup, exceeding the VAE, WDCNN, QCNN, and TCNN results reported under that allocation. BearLLM_FCN and BearingFM attain higher values under some alternative allocations.

Table 11 and Figure 7 show the CWRU StrictGroup seed-44 per-class results. The normal class has eighteen test instances and each fault class has nine. The severe rolling-element-fault recall is 0.22, with most errors assigned to minor or moderate rolling-element categories.

Figures 8 and 9 visualize the 6016-dimensional encoder features and PGA fault-class probability vectors for the Mixed and MBHM test sets under StrictGroup (seed 42).

The t-SNE panels provide two-dimensional qualitative visualizations of encoder features and probability vectors. No representation metric is calculated from the projected coordinates.

Figure 10 shows the optimization trajectory of each training stage for the Mixed dataset under StrictGroup. The VAE, PGA, and language-modeling objectives have different numerical scales.

Table 11. Per-class performance and test support on CWRU (StrictGroup, seed = 44).

Fault Type	Precision	Recall	F1-Score	Support
Normal condition	1.00	1.00	1.00	18
Inner race fault (minor)	1.00	1.00	1.00	9
Inner race fault (moderate)	1.00	1.00	1.00	9
Inner race fault (severe)	1.00	1.00	1.00	9
Rolling element fault (minor)	0.82	1.00	0.90	9
Rolling element fault (moderate)	0.64	1.00	0.78	9
Rolling element fault (severe)	1.00	0.22	0.36	9
Outer race fault (minor)	1.00	1.00	1.00	9
Outer race fault (moderate)	1.00	1.00	1.00	9
Outer race fault (severe)	1.00	1.00	1.00	9
Macro average	0.95	0.92	0.90	99
Weighted average	0.95	0.93	0.91	99

Note: Bold values highlight perfect (1.00) class-specific scores.

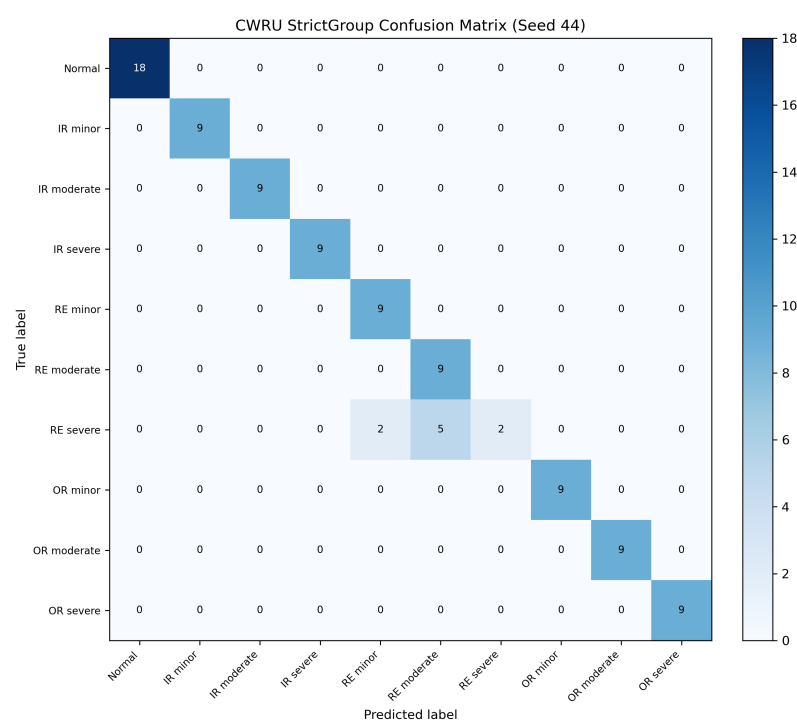


Figure 7. Confusion matrix on CWRU under the StrictGroup protocol with seed = 44. The main errors are concentrated within rolling element severity levels rather than across different fault locations.

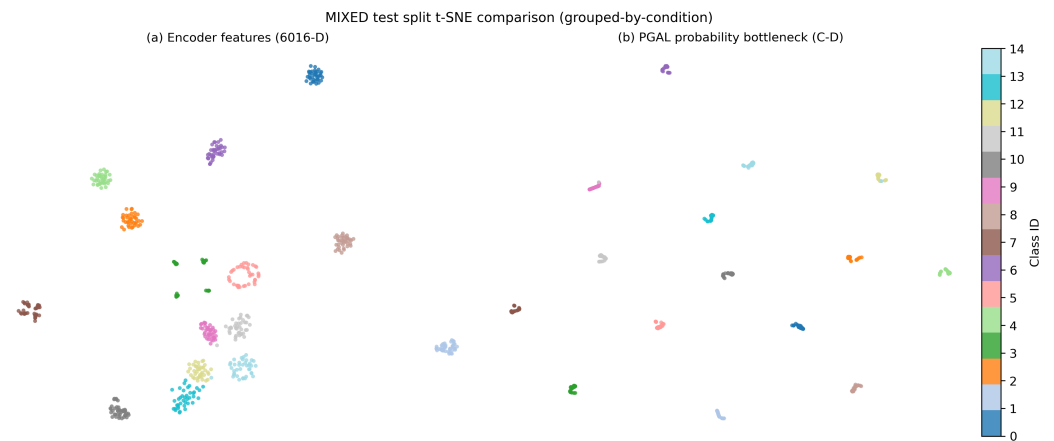


Figure 8. Mixed t-SNE visualization of encoder features and fault-class probabilities.

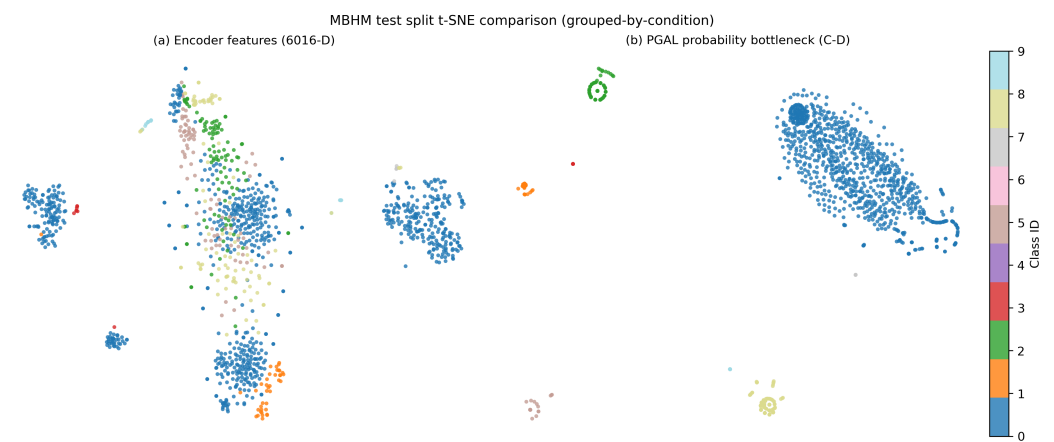


Figure 9. MBHM t-SNE visualization of encoder features and fault-class probabilities.

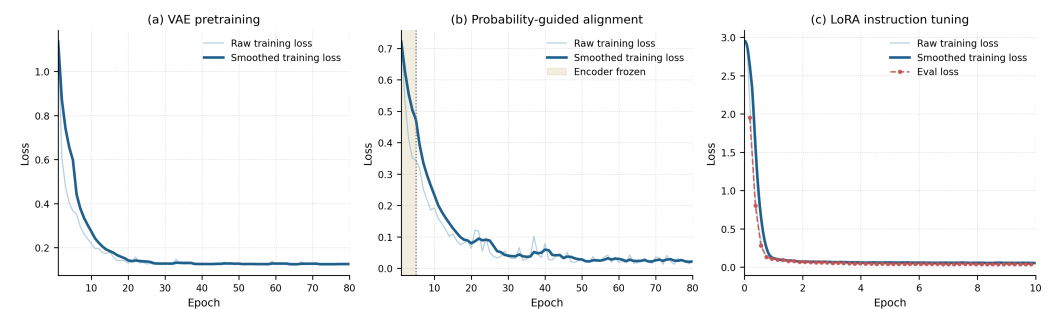


Figure 10. Loss variation during progressive collaborative optimization.

4.4. Ablation Results

Table 12 presents signal-side ablations of the PGA interface, training budget, and encoder-freezing strategy.

Table 12. Signal-side classification ablations (% , 10-run aggregate means per setting).

Dataset	PGA-LLM	w/o PGA	Only PGA	Reduced Training Epochs	No Encoder Freezing
MBHM	96.3	93.9	94.6	94.8	95.7
CWRU	97.1	94.7	95.2	95.4	96.2
Gear	99.0	97.4	97.8	98.0	98.6
Mixed	93.4	90.2	91.0	91.3	92.1

Note: Bold values indicate the highest result within each dataset row.

“w/o PGA” removes the probability-guided interface; “Only PGA” retains the PGA module but omits the full multi-loss constraints; “Reduced Training Epochs” shortens training; and “No Encoder Freezing” removes early encoder freezing. The largest decrease occurs for “w/o PGA” on Mixed, from 93.4% to 90.2%. The entries are ten-run aggregate means, and boldface identifies the largest value in each dataset.

4.5. Practical Deployment Considerations

The encoder and PGA module provide the signal-side output, whereas autoregressive report generation introduces latency dependent on model size, decoding length, and hardware. Table 13 reports a fixed-checkpoint CWRU audit of these routes.

Table 13. CWRU StrictGroup validation-path efficiency audit (batch size 1).

Route	Parameters	Mean/P95 Latency (ms)	Peak Memory (MiB)	FLOPs
Signal-side classifier	6.02 M	2.10/2.17	36.87	6.17 M
Classifier + template	6.02 M	2.13/2.22	36.87	6.17 M
Full PGA-LLM report	1.565 B	4260/5539	3091.12	1.30 T

The audit uses one validation record per CWRU class, two warm-up iterations, and 30/5 timed iterations per record for the signal-side/report routes. Inputs are preloaded to one RTX A5000 GPU, and transfer time is excluded. Full-report decoding is greedy with at most 128 new tokens. Peak memory is measured during timing; FLOPs are operator-accounted lower bounds for one representative record. M, B, and T denote 10^6 , 10^9 , and 10^{12} , respectively.

Separate cached passes measured 1.99 ms for the signal encoder, 0.34 ms for the PGA prompt bridge, and 4240.20 ms for greedy LLM decoding from a prebuilt prompt; these are component-level measurements.

5. Conclusions

In this study, we propose PGA-LLM for mapping vibration signals to LLM input through structure-preserving representation learning and probability-guided alignment. The encoder produces discriminative signal features, and the PGA interface fuses probability guidance with residual features before projecting them to continuous soft prompts for LoRA-adapted report generation. The Stage 2 signal-side output achieves aggregate accuracies of 97.1%, 99.0%, 93.4%, and 96.3% on CWRU, Gear, Mixed, and MBHM, respectively. Table 2 reports the StrictGroup class-support distribution for MBHM; the ablations summarize the contributions of the PGA interface and staged training schedule to these signal-side results.

Future work will extend this evaluation to cross-device and cross-domain settings, quantify the factual and practical quality of generated reports, and investigate lighter-weight report-generation strategies for deployment.

Further extensions include open-set and out-of-distribution fault recognition, continual adaptation, multimodal sensing, and traceable evidence for generated report statements.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/technologies14080494/s1>, PGA-LLM Technologies Supplementary Materials: data-construction and StrictGroup allocation protocol; seed-level split statistics and per-class split counts; allocation-schematic generation script; LLM-stage reproducibility record (prompt/target-report contract, Stage 3 configuration, parameter accounting, environment snapshot, and reproduction commands); and CWRU efficiency-audit inputs, manifest, timing/memory results, and FLOP record.

Author Contributions: Conceptualization, T.W. and Y.D.; methodology, T.W.; software, T.W.; validation, T.W., S.F., Q.L. (Qiongyao Liu) and H.C.; formal analysis, T.W.; investigation, T.W. and H.C.; resources, Y.D. and Q.L. (Qing Liu); data curation, T.W.; writing—original draft preparation, T.W.; writing—review and editing, Y.D., S.F., Q.L. (Qiongyao Liu), H.C. and Q.L. (Qing Liu); visualization, T.W.; supervision, Y.D. and Q.L. (Qing Liu). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The public datasets analyzed in this study are available from the following repositories: the CWRU bearing dataset is available from the Case Western Reserve University Bearing Data Center at <https://engineering.case.edu/bearingdatacenter/download-data-file> (accessed on 2 August 2026); the Gearbox Fault Diagnosis Data used for the Gear dataset is available from OEDI/OpenEI at <https://data.openei.org/submissions/623> (accessed on 2 August 2026); the Mixed fault dataset of bearings and gearboxes is available from Mendeley Data at <https://data.mendeley.com/datasets/yrms2v89p7/1> (DOI: <https://doi.org/10.17632/yrms2v89p7.1>; accessed on 2 August 2026); and the MBHM dataset is available from Hugging Face at <https://huggingface.co/datasets/SIA-IDE/MBHM> (accessed on 2 August 2026). The source code, experiment scripts, and related implementation files for this study are publicly available at <https://github.com/oasislvzhouu/pgallm.git> (accessed on 2 August 2026). The Supplementary Materials include the processed split statistics, per-class split counts, data-construction protocol, and allocation-schematic generation script. Large trained model checkpoints and processed data splits are available from the corresponding author upon reasonable request, subject to dataset and model-license constraints.

Acknowledgments: The authors thank the providers of the public datasets used in this study. AI-assisted tools were used for language editing, formatting assistance, and support with manuscript preparation. The authors reviewed and edited all AI-assisted outputs and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LLM	Large Language Model
PGA	Probability-Guided Alignment
VAE	Variational Autoencoder
LoRA	Low-Rank Adaptation
CWRU	Case Western Reserve University
MBHM	Machinery Bearing Health Management

References

1. Matania, O.; Dattner, I.; Bortman, J.; Kenett, R.S.; Parmet, Y. A systematic literature review of deep learning for vibration-based fault diagnosis of critical rotating machinery: Limitations and challenges. *J. Sound Vib.* **2024**, *590*, 118562. [\[CrossRef\]](#)
2. Lei, Y.; Yang, B.; Jiang, X.; Jia, F.; Li, N.; Nandi, A.K. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mech. Syst. Signal Process.* **2020**, *138*, 106587. [\[CrossRef\]](#)
3. Zhao, R.; Yan, R.; Chen, Z.; Mao, K.; Wang, P.; Gao, R.X. Deep learning and its applications to machine health monitoring. *Mech. Syst. Signal Process.* **2019**, *115*, 213–237. [\[CrossRef\]](#)
4. Zhang, W.; Li, C.; Peng, G.; Chen, Y.; Zhang, Z. A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mech. Syst. Signal Process.* **2018**, *100*, 439–453. [\[CrossRef\]](#)
5. Rezazadeh, N.; Caputo, F.; Aversano, A.; Lamanna, G.; De Luca, A.; Perfetto, D. Prototype-attention domain adaptation for explainable bearing fault diagnosis. *Procedia Struct. Integr.* **2026**, *80*, 411–417. [\[CrossRef\]](#)

6. Rezazadeh, N.; Perfetto, D.; de Oliveira, M.; De Luca, A.; Lamanna, G. A fine-tuning deep learning framework to palliate data distribution shift effects in rotary machine fault detection. *Struct. Health Monit.* **2024**, *25*, 661–683. [\[CrossRef\]](#)
7. Chen, Z.; Gryllias, K.; Li, W. Intelligent fault diagnosis for rotary machinery using transferable convolutional neural network. *IEEE Trans. Ind. Inform.* **2020**, *16*, 339–349. [\[CrossRef\]](#)
8. Liao, J.X.; Dong, H.C.; Sun, Z.Q.; Sun, J.; Zhang, S.; Fan, F.L. Attention-Embedded Quadratic Network (Qtention) for Effective and Interpretable Bearing Fault Diagnosis. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 3511113. [\[CrossRef\]](#)
9. Wang, J.; Chen, Y.; Hao, S.; Peng, X.; Hu, L. A deep transfer learning method for intelligent fault diagnosis of machines with unlabeled data. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3925–3935.
10. Zhao, B.; Zhang, X.; Li, H.; Yang, Z. Applications of unsupervised deep transfer learning to intelligent fault diagnosis: A survey and comparative study. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 3525828. [\[CrossRef\]](#)
11. Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; Long, M. TimesNet: Temporal 2D-variation modeling for general time series analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 10751–10765.
12. Choi, H.; Kang, P. Multi-task self-supervised time-series representation learning. *Inf. Sci.* **2024**, *671*, 120654. [\[CrossRef\]](#)
13. Yue, Z.; Wang, Y.; Duan, J.; Yang, T.; Huang, C.; Tong, Y.; Xu, B. TS2Vec: Towards universal representation of time series. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 8980–8987. [\[CrossRef\]](#)
14. Zhang, W.; Yang, L.; Geng, S.; Hong, S. Self-supervised time series representation learning via cross reconstruction transformer. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 16129–16138. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Xiao, Y.; Shao, H.; Wang, J.; Yan, S.; Liu, B. Bayesian variational transformer: A generalizable model for rotating machinery fault diagnosis. *Mech. Syst. Signal Process.* **2024**, *207*, 110936. [\[CrossRef\]](#)
16. Zhang, J.; Zhang, K.; An, Y.; Luo, H.; Yin, S. An integrated multitasking intelligent bearing fault diagnosis scheme based on representation learning under imbalanced sample condition. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 6231–6242. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Fu, D.; Liu, J.; Zhong, H.; Zhang, X.; Zhang, F. A novel self-supervised representation learning framework based on time-frequency alignment and interaction for mechanical fault diagnosis. *Knowl.-Based Syst.* **2024**, *295*, 111846. [\[CrossRef\]](#)
18. Zhou, T.; Niu, P.; Wang, X.; Sun, L.; Jin, R. One Fits All: Power General Time Series Analysis by Pretrained LM. In *Proceedings of the Advances in Neural Information Processing Systems; Neural Information Processing Systems Foundation, Inc. (NeurIPS): La Jolla, CA, USA, 2023; Volume 36.* [\[CrossRef\]](#)
19. Ansari, A.F.; Stella, L.; Turkmen, C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S.S.; Pineda Arango, S.; Kapoor, S.; et al. Chronos: Learning the Language of Time Series. *arXiv* **2024**. [\[CrossRef\]](#)
20. Goswami, M.; Szafer, K.; Choudhry, A.; Cai, Y.; Li, S.; Dubrawski, A. MOMENT: A Family of Open Time-series Foundation Models. In *Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024; Volume 235, pp. 16115–16152.* [\[CrossRef\]](#)
21. Liu, Y.; Qin, G.; Huang, X.; Wang, J.; Long, M. AutoTimes: Autoregressive Time Series Forecasters via Large Language Models. In *Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 9–15 December 2024; Volume 37.* [\[CrossRef\]](#)
22. Qaid, H.A.A.M.; Zhang, B.; Su, S.; Li, D.; Ng, S.K.; Li, W. Large language models for explainable fault diagnosis of machines. *Eng. Appl. Artif. Intell.* **2026**, *163*, 113131. [\[CrossRef\]](#)
23. Zhang, J.; Zhang, C.; Lu, J.; Zhao, Y. Domain-specific large language models for fault diagnosis of heating, ventilation, and air conditioning systems by labeled-data-supervised fine-tuning. *Appl. Energy* **2025**, *377*, 124378. [\[CrossRef\]](#)
24. Wang, J.; Li, T.; Yang, Y.; Chen, S.; Zhai, W. DiagLLM: Multimodal reasoning with large language model for explainable bearing fault diagnosis. *Sci. China Inf. Sci.* **2025**, *68*, 160103. [\[CrossRef\]](#)
25. Li, D.; Pang, Z.; Chen, Y.; Yang, K.; Shao, J.; Luo, Y.; Zeng, Y.; He, C.; Gao, Y. FD-MVLLM: Fault diagnosis based on multimodal vibration data and large language model for bearing system. *Mech. Syst. Signal Process.* **2025**, *239*, 113226. [\[CrossRef\]](#)
26. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 8748–8763.* [\[CrossRef\]](#)
27. Alayrac, J.B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. Flamingo: A Visual Language Model for Few-Shot Learning. In *Proceedings of the Advances in Neural Information Processing Systems, Orleans, LA, USA, 28 November–9 December 2022; Volume 35.* [\[CrossRef\]](#)
28. Li, J.; Li, D.; Savarese, S.; Hoi, S. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; Volume 202, pp. 19730–19742.* [\[CrossRef\]](#)
29. Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual instruction tuning. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 34892–34916. [\[CrossRef\]](#)
30. Ding, N.; Qin, Y.; Yang, G.; Wei, F.; Yang, Z.; Su, Y.; Hu, S.; Chen, Y.; Chan, C.M.; Chen, W.; et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nat. Mach. Intell.* **2023**, *5*, 220–235. [\[CrossRef\]](#)

31. Li, Y.F.; Wang, H.; Sun, M. ChatGPT-like large-scale foundation models for prognostics and health management: A survey and roadmaps. *Reliab. Eng. Syst. Saf.* **2024**, *243*, 109850. [[CrossRef](#)]
32. Tao, L.; Li, S.; Liu, H.; Huang, Q.; Ma, L.; Ning, G.; Chen, Y.; Wu, Y.; Li, B.; Zhang, W.; et al. An outline of Prognostics and health management Large Model: Concepts, Paradigms, and challenges. *Mech. Syst. Signal Process.* **2025**, *232*, 112683. [[CrossRef](#)]
33. Hendriks, J.; Dumond, P.; Knox, D. Towards better benchmarking using the CWRU bearing fault diagnosis dataset. *Mech. Syst. Signal Process.* **2022**, *169*, 108732. [[CrossRef](#)]
34. Peng, H.; Liu, J.; Du, J.; Gao, J.; Wang, W. BearLLM: A Prior Knowledge-Enhanced Bearing Health Management Framework with Unified Vibration Signal Representation. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March 2025; Volume 39, pp. 19866–19874. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.