10.37965/jdmd.2026.1588

Vibration-Language Model for Fault Diagnosis with Numerically Reliable Evidence

Chenyang Liu[a], Xiwei Li[a], Bin Yang[a]*, Xiang Li[a], Naipeng Li[a], Jiaqi Ye[b], Yaguo Lei[a]

[a]Key Laboratory of Education Ministry for Modern Design and Rotor-Bearing System, Xi'an Jiaotong University, Xi'an 710049, China

[b]School of Engineering, University of Birmingham, Birmingham, B15 2TT, United Kingdom

Received 30 June 2026; Revised 28 July 2026; Accepted 01 September 2026; Published online 03 September 2026

*Corresponding author: binyang@xjtu.edu.cn

Fault diagnosis methods for mechanical equipment should not only identify fault categories but also provide verifiable diagnostic evidence. Existing deep learning models usually output only labels or confidence scores, making it difficult to establish an interpretable diagnostic reasoning process. Large language models have strong capabilities in evidence organization and explanation generation. However, a modality gap exists between vibration signals and discrete language tokens. In addition, key numerical evidence in generated diagnostic reports may be inaccurate or hallucinated. To address these issues, this paper proposes a Vibration-Language Model (ViLM). The proposed method encodes angle-domain waveforms and order spectra into learnable vibration tokens and maps them into the embedding space of a large language model. With signal-description alignment and diagnostic instruction tuning, this architecture enables interpretable fault diagnosis based on vibration-informed language generation. Furthermore, a numerical evidence-constrained decoding method is designed to embed the computation and backfilling of key numerical evidence into the generation process. Experiments show that ViLM improves fault classification and evidence-supported explanation while maintaining high numerical reliability in generated diagnostic reports.

Keywords: Fault diagnosis, Large language model, Interpretable diagnosis, Numerical reliability

1 Introduction

Fault diagnosis is essential for ensuring the safe and reliable operation of mechanical equipment. Machine learning methods [1] and Transformer-based models [2] have been widely used for vibration-based fault diagnosis. Under varying operating conditions and data distribution shifts, transfer learning and contrastive learning have further improved model adaptability [3-5].

However, engineering diagnosis requires more than label prediction. It also requires the rationale behind the conclusion to be clearly explained, and the key evidence

should be verifiable. Existing deep models usually compress signals into latent representations and directly map them to fault categories. Their explanations often remain at the level of attention weights or feature importance. Recent studies have begun to improve the readability and stability of diagnostic evidence through explainable modules and causal constraints [6,7]. Nevertheless, how to form an interpretable reasoning process supported by explicit evidence remains an open issue.

The development of large language models (LLMs) provides a new opportunity for interpretable diagnosis. Through large-scale pretraining, instruction tuning, and reasoning enhancement, LLMs have shown strong capabilities in task understanding, instruction following, and explanation generation [8-10]. Multimodal alignment studies further show that non-textual information can be connected to language models through encoders and mapping modules [11-13]. These advances suggest that if structural evidence in vibration signals can be effectively integrated into an LLM, the model may generate diagnostic reports that better match engineering practice.

However, directly applying LLMs to mechanical fault diagnosis still faces two key challenges. The first is the modality gap. Vibration signals are continuous one-dimensional signals, and fault-related information is embedded in their temporal and spectral structures. In contrast, the inputs of LLMs are discrete language tokens, and LLMs do not have a natural interface for vibration representation. Simply converting manually selected features into text may lead to information loss. The second challenge is numerical reliability. Diagnostic reports often need to provide numerical evidence such as dominant orders, amplitudes, and energy ratios. Due to the autoregressive generation mechanism of LLMs, numerical deviations and hallucinated values may occur. For fault diagnosis, numerical errors are not minor wording issues because they directly undermine the evidence chain and the credibility of the conclusion.

To address these challenges, this paper proposes an interpretable fault diagnosis method based on a Vibration–Language Model (ViLM) and numerical evidence-constrained decoding. The proposed method connects angle-domain and order-domain information to a language model through vibration tokens, so that diagnostic reasoning can be performed in a unified representation space. Meanwhile, a numerical evidence-constrained decoding method is proposed. It enables the model to call deterministic signal analysis tools when quantitative evidence is required during decoding. The computed results are then backfilled into the context for subsequent explanation generation. The main contributions of this paper are as follows.

(1) A Vibration-Language Model (ViLM) is proposed for fault diagnosis. One-dimensional vibration signals are represented as learnable vibration tokens and connected to the embedding space of an LLM through two-stage training. This enables signal understanding, evidence organization, and explanation generation for diagnostic question answering.

(2) A numerical evidence-constrained decoding method is proposed. It introduces structured tool calling and result backfilling into LLM decoding to reduce numerical hallucination and metric mismatch. This improves the numerical reliability of diagnostic reports.

The remainder of this paper is organized as follows. Section 2 reviews related work.

Section 3 presents the proposed ViLM framework and numerical evidence-constrained decoding method. Section 4 reports the experiments and analysis. Section 5 concludes this paper.

2 Related Work

2.1 Large Language Models and Multimodal Alignment

Cross-modal alignment studies offer useful insights into how non-textual information can be connected to LLMs. Vision–language models usually extract image representations with a visual encoder and transform them into a language-model-compatible space through a lightweight mapping or query module. For example, BLIP-2 uses a frozen image encoder and a frozen language model, with a Q-Former as the visual-to-language bridge [12]. Visual Instruction Tuning further enables language models to answer questions, describe content, and reason based on images [13]. These studies show that non-text modalities can be connected to LLMs through an encoder–mapping structure for subsequent understanding and generation.

In industrial fault diagnosis, FD-LLM constructs an LLM-based diagnostic process for complex equipment fault diagnosis [14]. TF-ProFM combines time-frequency representations with a prototype-based foundation model to improve the generalization ability of bearing fault diagnosis [15]. CNC-VLM explores a vision–language large model for imbalanced CNC fault diagnosis [16]. Related studies also include adaptive industrial LLMs for mechanical diagnosis under variable operating conditions [17], as well as industrial fault explanation and causal-chain reasoning methods combining digital twins, vision–language models, and knowledge graphs [18,19].

However, existing large-model-based fault diagnosis studies mainly focus on knowledge fusion and textual reasoning. Their input signals are often compressed into handcrafted features, image summaries, or textual descriptions, making it difficult for the model to directly perceive continuous waveform details and local impact patterns.

2.2 Reliable Evidence Generation with Tool-Augmented Decoding

LLMs may produce numerical hallucinations during autoregressive generation [20]. To improve numerical reliability, tool-augmented generation has become a promising technical direction. Toolformer enables language models to learn when to call external tools [21]. ReAct combines language-model reasoning with external actions, allowing models to complete complex tasks through interactive tool use [22].

In mechanical fault diagnosis, existing studies mainly follow two routes. One route converts vibration signals or diagnostic knowledge into textual representations for language models. For example, PHM-GPT constructs PHM instruction data through a signal-to-text process and unifies anomaly detection, fault diagnosis, and maintenance decision-making as language interaction tasks [23]. Tao et al. proposed an LLM-based framework for bearing fault diagnosis, where vibration features are organized into hierarchical textual evidence to support fault identification and explanation generation [24]. The other route uses LLM-based agents to organize the diagnostic process and call computational tools or process sensor data according to task requirements [25].

Nevertheless, these methods still have limitations. Textual representations compress raw signal information, including continuous waveforms, local impacts, and order-domain structures. Agent-based diagnosis methods

mainly rely on multi-step planning and discrete tool outputs. They lack a unified vibration representation and usually involve higher interaction and computational overhead.

3 Proposed Method

3.1 Overall Framework

ViLM formulates vibration-based fault diagnosis as a vibration-conditioned text generation task. Given a vibration signal $\boldsymbol{X}_{\mathrm{v}}$, ViLM generates different types of diagnostic text according to the input instruction. In the signal-description alignment stage, $\boldsymbol{X}_{\mathrm{t}}^{\mathrm{s}}$ denotes the signal-description prompt, and the target output is the signal description $\boldsymbol{Y}^{\mathrm{s}}$. In the diagnostic instruction tuning stage, $\boldsymbol{X}_{\mathrm{t}}^{\mathrm{d}}$ denotes the diagnostic instruction, and the target output is the diagnostic report $\boldsymbol{Y}^{\mathrm{d}}$, including the fault category, numerical evidence, and explanatory text. The overall framework is shown in Fig. 1.

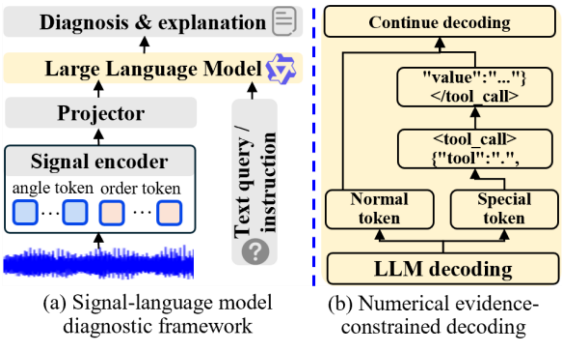


Fig. 1. Overall framework of ViLM with numerical evidence-constrained decoding

The proposed method consists of vibration preprocessing, a signal encoder, a projector, an LLM decoder, and a numerical evidence-constrained decoding module. Vibration preprocessing transforms the raw time-domain signal into angle-domain waveforms and order-domain representations, which are more suitable for analysis under different rotating speeds. The signal encoder $g(\cdot)$ extracts the vibration representation $\boldsymbol{Z}_{\mathrm{v}}$ from the angle-domain waveform and order spectrum. The projector $\boldsymbol{W}$ maps $\boldsymbol{Z}_{\mathrm{v}}$ into the LLM embedding space to obtain the projected vibration tokens $\boldsymbol{H}_{\mathrm{v}}$. The LLM then generates response text based on the vibration context and the text instruction. When decoding reaches a field requiring key numerical evidence, the numerical evidence-constrained decoding module calls deterministic signal analysis tools and inserts the computed results into the generated context.

3.2 Signal Encoder

The signal encoder takes the angle-domain waveform and the order spectrum as inputs, as shown in Fig. 2. Each raw vibration segment is first standardized and resampled in the angle domain according to the rotating speed. The order spectrum is then calculated from the angle-domain waveform. The angle-domain waveform is used to describe local impacts and inter-revolution periodic patterns, while the order spectrum is used to describe fault-related orders, harmonics, and modulation sidebands. Therefore, the signal is represented by angle tokens and order tokens to preserve both waveform structures and order-domain evidence.

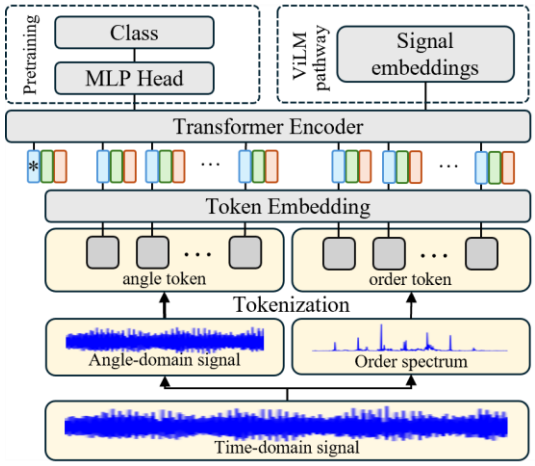


Fig. 2. Architecture of the signal encoder

Note: Blue, green, and red denote value, position, and type embeddings, respectively; * denotes the CLS token.

Angle tokens are obtained by segmenting the angle-domain waveform, and each token corresponds to a local waveform segment. These tokens capture short-time impacts, amplitude variations, and repetitive patterns across revolutions. Order tokens are obtained by segmenting the order spectrum with a non-uniform partition. The low-order range is represented with finer resolution to retain fault-characteristic orders, harmonics, and adjacent sidebands. The high-order range is represented more coarsely to describe broadband energy distribution.

The angle tokens and order tokens are first mapped into a unified hidden space. Let a_i denote the i-th angle token and o_j denote the j-th order token, where $i = 1, \ldots, N_a$ and $j = 1, \ldots, N_o$. Since the three order ranges produce tokens with different input dimensions, separate linear projection matrices are used for the three ranges. Their value embeddings are obtained as:

$$\boldsymbol{e}_{\mathrm{i}}^{\mathrm{a}} = \boldsymbol{W}_{\mathrm{a}}\boldsymbol{a}_{\mathrm{i}} + \boldsymbol{b}_{\mathrm{a}},$$

$$\boldsymbol{e}_{\mathrm{j}}^{\mathrm{o}} = \begin{cases} \boldsymbol{W}_o^{(1)}\mathbf{o}_j + \boldsymbol{b}_o^{(1)}, 1 \le j \le 60 \\ \boldsymbol{W}_o^{(2)}\mathbf{o}_j + \boldsymbol{b}_o^{(2)}, 61 \le j \le 85 \\ \boldsymbol{W}_o^{(3)}\mathbf{o}_j + \boldsymbol{b}_o^{(3)}, 86 \le j \le 110 \end{cases}$$

Here, $\boldsymbol{W}_o^{(1)}$, $\boldsymbol{W}_o^{(2)}$, and $\boldsymbol{W}_o^{(3)}$ correspond to the 0–30, 30–150, and 150–500 order ranges, respectively.

The two types of embeddings are concatenated into a unified vibration token sequence. Let $\boldsymbol{E} = [\boldsymbol{e}_1^a, \ldots, \boldsymbol{e}_{N_a}^a, \boldsymbol{e}_1^o, \ldots, \boldsymbol{e}_{N_o}^o]$ denote the concatenated embedding sequence, where $N = N_a + N_o$. The k-th element of E is denoted as e_k. For the k-th token, type and position embeddings are added to distinguish token sources and preserve angle-domain or order-domain location information:

$$\boldsymbol{h}_{\mathrm{k}}^0 = \boldsymbol{e}_{\mathrm{k}} + \boldsymbol{e}_{\mathrm{k}}^{\mathrm{type}} + \boldsymbol{e}_{\mathrm{k}}^{\mathrm{pos}}$$

On this basis, a learnable CLS token is inserted before the vibration token sequence to aggregate global signal information during Transformer encoding. The final input sequence is expressed as:

$$\boldsymbol{H}^0 = [\boldsymbol{h}_{\mathrm{CLS}}^0; \boldsymbol{h}_1^0; \boldsymbol{h}_2^0; \cdots; \boldsymbol{h}_{\mathrm{N}}^0]$$

which is fed into the Transformer encoder to obtain the vibration embedding representation:

$$\boldsymbol{Z}_{\mathrm{v}} = \mathrm{Transformer}(\boldsymbol{H}^0)$$

The CLS output represents the global signal state, and the remaining token outputs retain local angle-domain and order-domain structures. The resulting $\boldsymbol{Z}_{\mathrm{v}}$ serves as the vibration representation for subsequent vibration-language alignment and diagnostic text generation.

Before the two-stage ViLM training, the signal encoder is pretrained with a lightweight classification head to learn mechanical-state-aware vibration representations. The classification head is then removed, and the pretrained encoder is kept frozen during the subsequent vibration-language training.

3.3 Two-Stage Training

As shown in Fig. 3, ViLM is trained in two stages: signal-description alignment and diagnostic instruction tuning. The two stages share the same vibration representation pipeline but differ in training objectives and trainable modules.

For compactness, the signal encoder described in Section 3.2 is denoted as $g(\cdot)$, which includes vibration token construction, embedding, CLS-token insertion, and Transformer encoding. In both stages, the vibration signal is first processed by $g(\cdot)$ to extract high-dimensional signal representations:

$$\boldsymbol{Z}_{\mathrm{v}} = g(\boldsymbol{X}_{\mathrm{v}})$$

where $\boldsymbol{X}_{\mathrm{v}}$ is the input vibration signal and $\boldsymbol{Z}_{\mathrm{v}}$ is the encoder output. The projector W maps $\boldsymbol{Z}_{\mathrm{v}}$ into the LLM embedding space:

$$\boldsymbol{H}_{\mathrm{v}} = \boldsymbol{W}\boldsymbol{Z}_{v}$$

The projected vibration tokens $\boldsymbol{H}_{\mathrm{v}}$ have the same embedding dimension as the text token embeddings and can therefore be concatenated with the instruction embeddings as LLM input context.

In the signal-description alignment stage, the model takes the vibration signal $\boldsymbol{X}_{\mathrm{v}}$ and the signal description prompt $\boldsymbol{X}_{\mathrm{t}}^{\mathrm{s}}$ as input, and generates the signal description $\boldsymbol{Y}^{\mathrm{s}}$. After embedding the prompt as $\boldsymbol{H}_{\mathrm{t}}^{\mathrm{s}}$, the projected vibration tokens and text tokens are concatenated as the conditional context of the LLM:

$$\boldsymbol{Y}^{\mathrm{s}} = f([\boldsymbol{H}_{\mathrm{v}}; \boldsymbol{H}_{t}^{s}])$$

where $[\cdot;\cdot]$ denotes token sequence concatenation, and $f(\cdot)$ represents LLM-based conditional generation. The corresponding autoregressive generation process is formulated as:

$$p(\boldsymbol{Y}^{\mathrm{s}}|\boldsymbol{X}_{\mathrm{v}}, \boldsymbol{X}_{\mathrm{t}}^{\mathrm{s}}) = \prod_{i=1}^{N_s} p(y_i^s|\boldsymbol{Y}_{<\mathrm{i}}^{\mathrm{s}}, \boldsymbol{H}_{\mathrm{v}}, \boldsymbol{H}_{t}^{s})$$

In this stage, the signal encoder $g(\cdot)$ and the LLM $f(\cdot)$ are frozen, and only the projector $\boldsymbol{W}$ is trained. Through this stage, the projector learns to map the vibration features into a representation space compatible with the LLM embedding space. This enables subsequent language generation to use time-domain and order-domain information from the vibration signal.

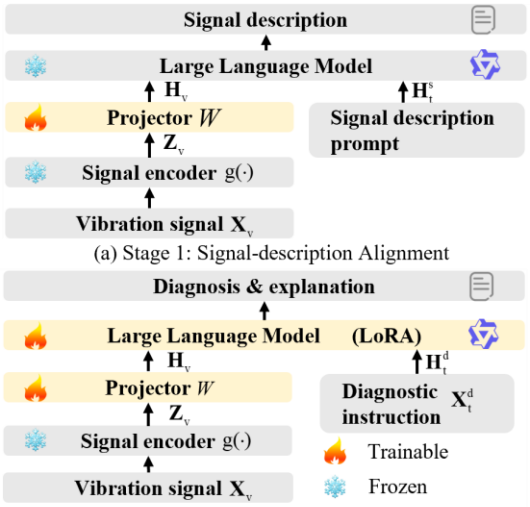


Fig. 3. Two-stage training pipeline of ViLM

In the diagnostic instruction tuning stage, the model takes the vibration signal $\boldsymbol{X}_{\mathrm{v}}$ and diagnostic instruction $\boldsymbol{X}_{\mathrm{t}}^{\mathrm{d}}$ as input, and generates the diagnosis and explanation $\boldsymbol{Y}^{\mathrm{d}}$. The diagnostic instruction is embedded as $\boldsymbol{H}_{\mathrm{t}}^{\mathrm{d}}$, and the LLM generates the final response from both vibration and instruction tokens:

$$\boldsymbol{Y}^{\mathrm{d}} = f\left([\boldsymbol{H}_{\mathrm{v}}; \boldsymbol{H}_{t}^{d}]\right)$$

The corresponding autoregressive generation process is expressed as:

$$p\left(\boldsymbol{Y}^{\mathrm{d}} \mid \boldsymbol{X}_{\mathrm{v}}, \boldsymbol{X}_{\mathrm{t}}^{\mathrm{d}}\right) = \prod_{i=1}^{N_d} p\left(y_i^d \mid \boldsymbol{Y}_{<\mathrm{i}}^{\mathrm{d}}, \boldsymbol{H}_{\mathrm{v}}, \boldsymbol{H}_{\mathrm{t}}^{\mathrm{d}}\right)$$

In this stage, the signal encoder $g(\cdot)$ remains frozen, while the projector W and the LoRA adapters of the LLM $f(\cdot)$ are jointly updated. This enables the model to generate fault categories, diagnostic evidence, and explanatory text according to diagnostic instructions.

3.4 Numerical Evidence-Constrained Decoding

This paper proposes a numerical evidence-constrained decoding method to improve the reliability of quantitative evidence in diagnostic reports. As shown in Fig. 4, key numerical fields are represented as structured

tool calls during LLM decoding. After deterministic tool execution, the computed results are filled back into the generated sequence.

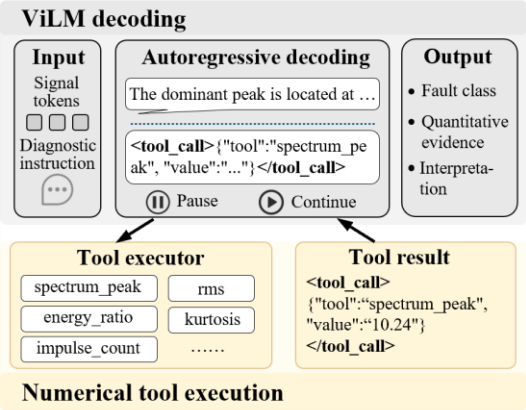


Fig. 4. Structured tool decoding process

For training samples that require quantitative evidence, the target output is constructed with tool-call annotations. During training, natural-language tokens, tool-call tokens, and tool-result tokens are included in a unified autoregressive target sequence and jointly optimized using the standard cross-entropy loss. Thus, the model learns when to trigger a tool call, which metric to select, the representation format of the returned result, and how to continue the explanation after the numerical result is inserted. Although the tool-result tokens participate in the training loss, their numerical values are not generated by the LLM during inference. Instead, they are computed by deterministic signal-analysis tools and backfilled into the decoding context. During inference, the LLM generates diagnostic text conditioned on the vibration tokens and the diagnostic instruction. When a quantitative item is required, the model does not directly generate the numerical value. Instead, it outputs a structured tool-call field that specifies the required metric, such as peak order, order-band energy ratio, kurtosis, or impact-related statistics. The decoding controller then parses the tool-call field and dispatches it to the corresponding deterministic signal analysis tool.

The tool computes the requested metric from the input vibration signal and returns the numerical result in a predefined format. The returned value is inserted into the current generated text, and the LLM continues decoding based on the updated context. In this way, the subsequent explanation is conditioned on reproducible numerical evidence rather than freely generated values.

Numerical evidence-constrained decoding integrates tool triggering, deterministic calculation, and result backfilling into the generation process. This design keeps the diagnostic explanation readable while improving the accuracy and verifiability of key quantitative evidence.

4 Experiments

4.1 Experimental Setup

To evaluate the proposed method, experiments are conducted on seven bearing datasets. These datasets include six public datasets and one in-house test-rig dataset. They differ in rotating speed, sampling frequency, data source, and fault type, as shown in Table 1. Within-dataset experiments in Sections 4.2 and 4.4 used a 70:15:15 train/validation/test split, with results reported on the test set.

Cross-dataset generalization was further evaluated using seven-fold leave-one-dataset-out (LODO) testing. In each fold, one dataset was used exclusively as the unseen target test set, while signal-encoder pretraining, Stage-1 alignment, Stage-2 instruction tuning, validation, and model selection were all performed exclusively on the other six source datasets. No sample

from the target dataset was used at any stage of model development. Samples were grouped by dataset, fault class, and original signal record during source-domain splitting to prevent segment leakage. The target fault-label set was provided to all methods. ViLM results are reported as the mean ± standard deviation over seeds 42, 43, and 44; the two frozen baselines used greedy decoding and produced deterministic outputs.

All samples are processed using the same pipeline. The pipeline includes standardization, angle-domain resampling, and order spectrum calculation. The angle-domain waveform and order spectrum are then converted into vibration tokens, following the encoder design in Section 3.2. All experiments use Qwen2.5-7B-Instruct as the LLM backbone and are conducted on two NVIDIA RTX 3090 GPUs. The main implementation settings of ViLM are summarized in Table 2.

Each input contains 20 revolutions resampled to 1,000 points per revolution. The resulting waveform is divided into 100 angle tokens of 200 points (0.2 revolution) each. The order spectrum is represented over 0–500 orders with a resolution of 0.05 order. Unavailable high-order bins are zero-padded, and the spectrum is divided into 110 tokens: 60/25/25 tokens over 0–30/30–150/150–500 orders. All angle tokens share one value-projection matrix, whereas the order tokens in the three ranges use three separate value-projection matrices. The signal encoder comprises four Transformer layers with a hidden dimension of 512, eight attention heads, and an FFN dimension of 2,048, producing 211 tokens of dimension 512 including the CLS token.

Three methods are compared: Feature text + LLM, Tool-calling Agent, and the proposed ViLM. Feature text + LLM uses manually organized statistical and spectral indicators as text prompts. Tool-calling Agent obtains diagnostic indicators through external signal-analysis tools. ViLM directly connects vibration tokens to the LLM. It applies numerical evidence-constrained decoding when quantitative evidence is required. The Tool-calling Agent first collects tool outputs and then generates a report, whereas ViLM uses learned vibration tokens and inserts each tool result during decoding, enabling a more direct link between numerical evidence and the diagnostic conclusion. For each dataset, all compared methods use the same data split, available fault labels, and LLM backbone unless otherwise specified. The methods are evaluated in terms of three aspects: fault classification, evidence-supported explanation, and numerical reliability. Accuracy and Macro-F1 are used to evaluate fault classification performance. Numerical Accuracy (NA) is the percentage of evaluated fields satisfying an absolute error of no more than 0.2 or a relative error of no more than 3%: $\mathrm{NA} = \frac{1}{M}\sum_{j=1}^{M} \mathrm{I}\left(\left|\widehat{y_j} - y_j\right| \le 0.2 \ \vee \ \frac{|\widehat{y_j} - y_j|}{|y_j|} \le 0.03\right) \times 100\%$, where M is the number of evaluated fields, $\widehat{y_j}$ is the generated value, and y_j is the tool-computed reference. For percentages, 0.2 denotes percentage points; integers require exact matching, and missing or unparsable fields are incorrect. Evidence adequacy evaluates the sufficiency of the diagnostic evidence provided in the generated report. Evidence Adequacy was assessed by a blinded ChatGPT-5.5 evaluator (Responses API; snapshot: gpt-5.5-2026-04-23; reasoning effort: medium; temperature: 0; top-p: 1.0; maximum output: 1,024 tokens; low verbosity; no tools). After removing method

identifiers, the evaluator scored signal phenomena, quantitative evidence, and mechanism linkage on a three-level scale: 0 indicates absent or incorrect evidence, 1 indicates partially correct but incomplete evidence, and 2 indicates complete and relevant evidence. For sample i, $EA_i = (s_i + q_i + m_i)/6$. Each report was evaluated three times, with median dimension scores retained. Dataset-level scores were averaged and reported as percentages. The same prompt, scoring criteria, and evaluation settings were used for all methods. SAR measures the proportion of fault attributions supported by the reported evidence. For method k on dataset d, $\mathrm{SAR}_{k,d} = \frac{1}{N_{k,d}} \sum_{i=1}^{N_{k,d}} I_i^{sup} \times 100\%$, where $N_{k,d}$ is the number of test samples and $I_i^{sup} = 1$ if the fault attribution is supported by sufficient evidence; otherwise, $I_i^{sup} = 0$. Diagnostic correctness is evaluated separately by Accuracy.

Table 1. Statistics of the Bearing Datasets

ID	Dataset	Data source	Rotational frequency (Hz)	Sampling frequency (Hz)	Fault types
D1	CWRU [26]	Public	28.83/29.17/29.53/29.95	12000	N/IF/OF/BF
D2	Paderborn [27]	Public	15/25	64000	N/IF/OF
D3	SEU [28]	Public	20/30	5120	N/IF/OF/BF
D4	RTS	In-house	25/30/35/40	12800	N/IF/OF/BF
D5	Ottawa [29]	Public	29.17	42000	N/IF/OF/BF/CF
D6	MFPT [30]	Public	25	48828/97656	N/IF/OF
D7	BJTU [31]	Public	20/40/60	64000	N/IF/OF/BF/CF

Note: N, IF, OF, BF, and CF denote the normal condition, inner race fault, outer race fault, ball fault, and cage fault, respectively. D1–D7 denote the dataset IDs used in Fig. 5.

Table 2. Implementation settings of ViLM

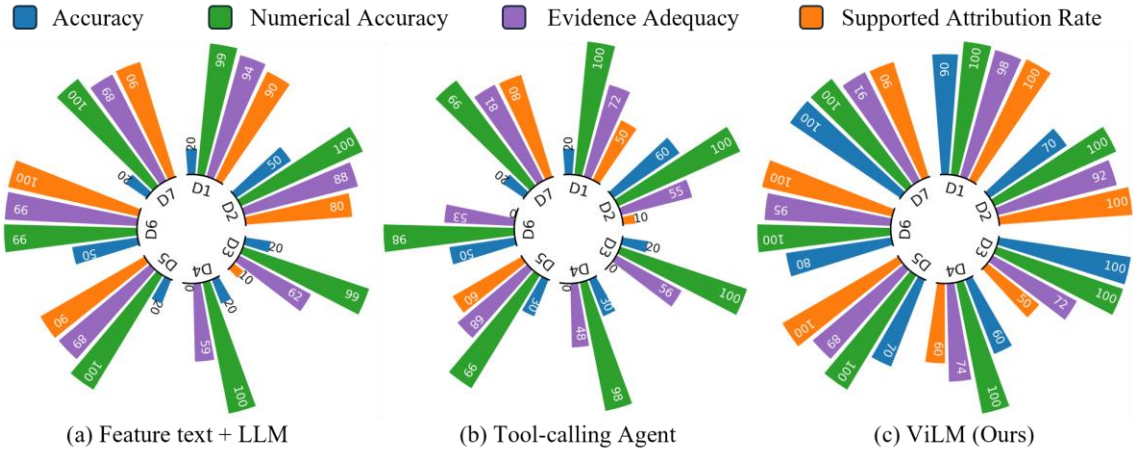
Item	Setting	Item	Setting
Input revolutions	20	Batch size	1
Points per revolution	1000	Gradient accumulation steps	4
Angle/order/CLS tokens	100/110/1	Projector/LoRA learning rates	2e-5/5e-6
Signal encoder pretraining epochs	30	LoRA layers/rank/alpha/dropout	4/8/16/0.05
Stage-1/Stage-2 epochs	5/10	Optimizer	AdamW
Decoding temperature	0	Max new tokens	1024

Note: Stage 1 denotes signal-description alignment, and Stage 2 denotes diagnostic instruction tuning. LoRA layers indicate the last LLM layers equipped with LoRA adapters.

Table 3. Overall performance comparison of different diagnostic methods

Method	Accuracy	Macro-F1	Evidence adequacy	SAR	Numerical accuracy
Feature text + LLM	28.57%	13.41%	82.82%	65.71%	99.64%
Tool-calling Agent	32.86%	20.61%	61.77%	28.57%	99.02%
ViLM (ours)	**81.43%**	**81.87%**	**87.23%**	**85.71%**	**100%**

Note: Evidence adequacy is scored by ChatGPT-5.5 using a fixed rubric after removing method names from the generated reports. SAR denotes the supported attribution rate.



(a) Feature text + LLM (b) Tool-calling Agent (c) ViLM (Ours)

Fig. 5. Dataset-wise performance comparison of different diagnostic methods

4.2 Diagnostic Performance and Numerical Reliability

Table 3 compares the three methods under the same train/validation/test split, fault label space, and LLM backbone. The feature text and tool outputs provided to the two baselines contain the same types of quantitative indicators used for numerical evidence in ViLM. Therefore, the results reflect the combined effects of vibration representation and evidence organization in the compared frameworks. ViLM outperforms the two baseline methods in both diagnostic accuracy and explanation reliability. The two baselines achieve nearly 100% numerical accuracy, but their diagnostic accuracy remains low. This indicates that the main limitation is not inaccurate calculation, but the ineffective use of calculated evidence. In contrast, ViLM connects vibration tokens with diagnostic generation and triggers tool calls during decoding, leading to better diagnosis and explanation quality.

Fig. 5 shows the dataset-wise performance of the three methods on seven datasets. The two baseline methods show large fluctuations in diagnostic accuracy and explanation reliability. This suggests that their diagnosis and attribution are sensitive to data distribution changes. ViLM maintains relatively high and consistent performance on most datasets. This indicates a more stable coupling among vibration representation, tool calling, and diagnostic generation.

In Fig. 6, each small marker represents one method on one dataset, while each large black-edged marker denotes the unweighted mean over the seven datasets; the two coordinates are calculated using the Evidence Adequacy and SAR definitions given in Section 4.1. Feature text + LLM can generate evidence-rich descriptions, but its attribution support remains lower than that of ViLM. This shows that evidence description does not necessarily lead to reliable explanation. Tool-calling Agent performs poorly on both dimensions, indicating that tool calling alone cannot ensure explanation reliability. The mean point of ViLM is closest to the upper-right region, showing better evidence adequacy and attribution reliability.

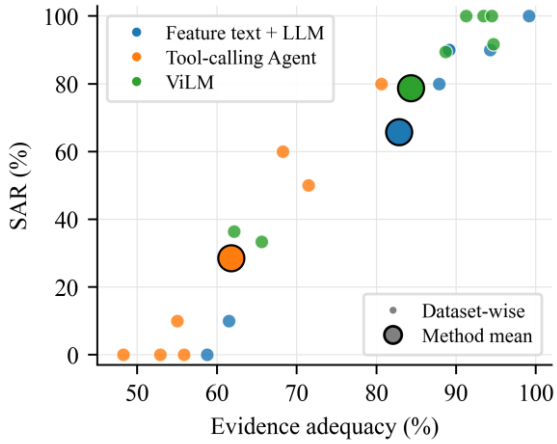


Fig. 6. Dataset-wise evidence reliability comparison

Fig. 7 shows the diagnostic results of different methods on the same outer-race fault sample. All three methods can provide accurate numerical evidence, but the sources of these values are different. Feature text + LLM relies on precomputed features that are directly included in the prompt. The Tool-calling Agent obtains features through external tool calls. ViLM generates the special token <tool_call> during decoding, which enables tool invocation and returns quantitative evidence.

Although the numerical evidence is mostly correct, the diagnostic conclusions are different. The two baseline methods both identify impact responses and concentrated order-domain energy. However, they fail to link the observed spectral peak to the characteristic order of the outer-race fault, and therefore misclassify the sample as an inner-

race fault. In contrast, ViLM captures the match between the 3.60-order spectral peak and the candidate outer-race fault order of 3.58. It correctly links the numerical evidence to the fault mechanism and diagnoses the sample as an outer-race fault.

"id": "14-sample_76".ground truth: "label": "OF", "impulse factor":"9.9", "kurtosis":"8.3", "impact count":"62", "main peak order":"97.20", "energy band": "110-120", "candidate outer-race fault order":"3.58", "matched order":"3.60".

Feature text + LLM:
Time-domain evidence: The time-domain features show a low RMS value (0.164), but a high impulse factor (9.886). Order-domain evidence: The order spectrum shows that the main order is 97.20, indicating that the main vibration component of this sample is related to the rotor rotation frequency. The order band with the highest energy is 110-120 orders, accounting for 48.5% of the total energy. Based on the time-domain and order-domain features, this sample can be diagnosed as an inner-race fault.

Tool-calling Agent:
According to the tool observation results, the signal has a high kurtosis value (8.2991) and contains many impact segments (97 segments). The spectrum analysis shows several frequency components. Among them, the frequency near 97.2 Hz has a relatively high amplitude (0.073493), which may correspond to the characteristic frequency of an inner-race fault. The energy analysis shows that the 110-120 Hz band has the highest energy ratio (49.0%), which also supports the possibility of an inner-race fault. The feature matching results show no clear features of other fault types.

ViLM:
In the time domain, the impulse factor is <tool_call> {"metric": "impulse_ factor","value":"9.9"}</tool_call>, the kurtosis is <tool_call> {"metric": "kurtosis","value":"8.3"}</tool_call>, and a total of <tool_call> {"metric": "impact_count","value":"62"} </tool_call> impacts are detected, indicating a relatively clear impact response in the signal. In the order domain, the main peak appears at <tool_call> {"metric":"peak_ order","value":"97.20"} </tool_call> order, and the energy is concentrated in the <tool_call> {"metric":"energy_band","value":"110-120"} </tool_call> order band. In addition, an actual spectral peak can be found near the candidate outer-race fault order, at <tool_call> {"metric":"matched_order","candidate_order":3.58,"value":"3.60"} </tool_call> order. Therefore, the signal is diagnosed as an outer-race fault.

Fig. 7. Case study of diagnostic reasoning across different methods

Note: Blue and red denote correct and incorrect content, respectively.

4.3 Cross-Dataset Generalization

Table 4 presents the LODO results, where each dataset is used once as an unseen target domain. ViLM achieves the highest accuracy on all seven target datasets and obtains an average accuracy of 63.45% ± 0.60%. This exceeds Feature text + LLM and the Tool-calling Agent by 37.41 and 35.04 percentage points, respectively. The improvement is observed across datasets with different data sources, speeds, sampling frequencies, and experimental platforms. These results indicate that the learned vibration tokens and signal–language alignment provide more transferable diagnostic information than fixed feature-text inputs or tool observations alone.

Table 4. Classification accuracy in leave-one-dataset-out experiments

Target dataset	Feature text + LLM	Tool-calling Agent	ViLM (ours)
D1	31.07%	15.53%	69.51% ± 0.40%
D2	54.95%	53.15%	60.87% ± 0.44%
D3	21.05%	31.58%	67.40% ± 5.82%
D4	13.64%	28.18%	50.65% ± 0.59%
D5	12.77%	11.06%	57.84% ± 1.27%
D6	36.11%	41.67%	80.67% ± 0.94%
D7	12.71%	17.68%	57.25% ± 2.16%
Avg.	26.04%	28.41%	63.45% ± 0.60%

Note: ViLM results are reported as the mean ± standard deviation over seeds 42, 43, and 44. The two frozen baselines use greedy decoding and therefore produce deterministic outputs.

4.4 Ablation Study

Two targeted ablation variants are evaluated under the same settings as the complete model to assess the overall effects of the complete Stage 2 configuration and the numerical evidence-constrained scheme. w/o Stage 2 retains signal-description alignment but removes the complete Stage 2 configuration, including diagnostic instruction tuning, LoRA adaptation, and further projector updating. The w/o Numerical Decoding variant retains the same two-stage training framework and is retrained under the same data splits and optimization settings as the complete model, but removes structured tool-call annotations during training and

deterministic result backfilling during inference. In this variant, reference numerical values are represented as ordinary text tokens in the training targets and are generated directly by the LLM during inference.

Table 5. Ablation results of ViLM

Model setting	Acc.	EA	SAR	NA
ViLM (ours)	81.43 %	87.23 %	85.71 %	100%
w/o Stage 2	N/A	48.13 %	N/A	89.85 %
w/o Numerical Decoding	30.00 %	38.83 %	26.00 %	26.85 %

Note: EA and NA denote Evidence Adequacy and Numerical Accuracy, respectively. Accuracy and SAR are not applicable to w/o Stage 2, because Stage 1 generates signal descriptions without fault diagnoses or fault attributions.

As shown in Table 5, Accuracy and SAR are not applicable to w/o Stage 2, because Stage 1 performs signal-description alignment without generating diagnostic conclusions. Its Numerical Accuracy of 89.85% indicates that Stage 1 has already acquired partial tool-calling capability. The higher Numerical Accuracy of the complete model suggests that this capability is further improved by the complete Stage 2 configuration. Its Evidence Adequacy is only 48.13%, showing that signal-description alignment alone cannot organize complete diagnostic evidence. After the complete Stage 2 configuration is introduced, Evidence Adequacy increases to 87.23%, demonstrating its overall effectiveness for diagnostic evidence organization.

Removing the numerical evidence-constrained training-and-decoding scheme reduces Accuracy, Evidence Adequacy, SAR, and Numerical Accuracy by 51.43, 48.40, 59.71, and 73.15 percentage points, respectively. This substantial degradation demonstrates the overall contribution of the numerical evidence-constrained scheme to numerical reliability, fault attribution, and explanation generation. Therefore, Stage 2 and numerical evidence-constrained decoding are both essential to the complete ViLM.

5 Conclusion

This paper developed ViLM for bearing fault diagnosis with evidence-supported diagnostic report generation. Experiments on multiple bearing datasets yield the following conclusions.

(1) The two-stage training strategy enables ViLM to connect vibration representations with diagnostic language generation. Compared with feature text prompting and Tool-calling Agent, ViLM achieves better fault classification performance and generates more reliable diagnostic explanations.

(2) During inference, numerical evidence-constrained decoding ensures that key quantitative evidence is obtained from deterministic signal-analysis tools rather than freely generated by the LLM. This mechanism maintains numerical consistency and helps the model link quantitative evidence to fault mechanisms more reliably.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (52305129, 52435003) and the Fundamental Research Funds for the Central Universities.

References

1. Yaguo Lei, Bin Yang, Xinwei Jiang, et al., "Applications of machine learning to machine fault diagnosis: A review and roadmap," Mechanical Systems and Signal Processing 138, 106587 (2020).

2. Wenjun Sun, Ruqiang Yan, Ruibing Jin, et al., "LiteFormer: A Lightweight and Efficient Transformer for Rotating Machine Fault Diagnosis," IEEE Transactions on Reliability 73, 2, 1258-1269 (2024).

3. Peng Zhu, Sai Ma, Qinkai Han, et al., "Deep spiking transfer learning for rotating machinery fault diagnosis," Mechanical Systems and Signal Processing 241, 113499 (2025).

4. Peng Zhu, Sai Ma, Qinkai Han, and Fulei Chu, "Deep Contrastive Transfer Learning for Rotating Machinery Fault Diagnosis," IEEE Transactions on Instrumentation and Measurement 74, 1-10 (2025).

5. Bin Yang, Yaguo Lei, Xiang Li, and Clive Roberts, "Deep Targeted Transfer Learning Along Designable Adaptation Trajectory for Fault Diagnosis Across Different Machines," IEEE Transactions on Industrial Electronics 70, 9, 9463-9473 (2023).

6. Daoguang Yang, Hamid Reza Karimi, and Len Gelman, "An explainable intelligence fault diagnosis framework for rotating machinery," Neurocomputing 541, 126257 (2023).

7. Chang Guo, Zhibin Zhao, Jiaxin Ren, et al., "Causal explaining guided domain generalization for rotating machinery intelligent fault diagnosis," Expert Systems with Applications 243, 122806 (2024).

8. Tom B. Brown, Benjamin Mann, Nick Ryder, et al., "Language models are few-shot learners," Advances in Neural Information Processing Systems 33, 1877-1901 (2020).

9. Long Ouyang, Jeffrey Wu, Xu Jiang, et al., "Training language models to follow instructions with human feedback," Advances in Neural Information Processing Systems 35, 27730-27744 (2022).

10. Jason Wei, Xuezhi Wang, Dale Schuurmans, et al., "Chain-of-thought prompting elicits reasoning in large language models," Advances in Neural Information Processing Systems 35, 24824-24837 (2022).

11. Alec Radford, Jong Wook Kim, Chris Hallacy, et al., "Learning transferable visual models from natural language supervision," Proceedings of the International Conference on Machine Learning 139, 8748-8763 (2021).

12. Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi, "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models," Proceedings of the International Conference on Machine Learning 202, 19730-19742 (2023).

13. Haotian Liu, Chunyuan Li, Qingyang Wu, et al., "Visual instruction tuning," Advances in Neural Information Processing Systems 36, 34892-34916 (2023).

14. Lin Lin, Sihao Zhang, Song Fu, et al., "FD-LLM: Large language model for fault diagnosis of complex equipment," Advanced Engineering Informatics 65, 103208 (2025).

15. Songyan Pi, Bojian Chen, Xinmin Zhang, and Zhihuan Song, "TF-ProFM: Prototype-Based Time-Frequency Foundation Model for Generalizable Industrial Bearing Fault Diagnosis," IEEE Transactions on Industrial Informatics 22, 7, 5898-5908 (2026).

16. Zisheng Wang, Junjie Chen, Chisen Wang, et al., "CNC-VLM: An RLHF-optimized industrial large vision-language model with multimodal learning for imbalanced CNC fault detection," Mechanical Systems and Signal Processing 245, 113838 (2026).

17. Chaojun Xu, Zisheng Wang, Yaqiang Jin, et al., "An adaptive industrial large language model for mechanical fault diagnosis under variable operating conditions," Advanced Engineering Informatics 74, 104821 (2026).

18. Quanning Xu, Guangrui Wen, Zihao Lei, et al., "Deep digital twin-powered large vision-language model for multi-scenario industrial fault diagnosis," Advanced Engineering Informatics 69, 103997 (2026).

19. Yun Gao, Yuhua Qin, Tao Wang, et al., "Knowledge graph enhanced large language model framework for causal chain reasoning in industrial fault diagnosis," Expert Systems with Applications 318, 132027 (2026).

20. Ziwei Ji, Nayeon Lee, Rita Frieske, et al., "Survey of Hallucination in Natural Language Generation," ACM Computing Surveys 55, 12, 1-38 (2023).

21. Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, et al., "Toolformer: Language models can teach themselves to use tools," Advances in Neural Information Processing Systems 36, 68539-68551 (2023).

22. Shunyu Yao, Jeffrey Zhao, Dian Yu, et al., "ReAct: Synergizing Reasoning and Acting in Language Models," International Conference on Learning Representations (2023).

23. Jiaxin Ren, Xue Liu, Tianlei Wang, et al., "PHM-GPT: A Large Language Model for Prognostics and Health Management," Engineering 63, 8, 156-172 (2026).

24. Laifa Tao, Haifei Liu, Guoao Ning, Wenyan Cao, Bohao Huang, and Chen Lu, "LLM-based Framework for Bearing Fault Diagnosis," Mechanical Systems and Signal Processing 224, 112127 (2025).

25. X. Y. Lee, L. Vidyaratne, A. Farahat, and C. Gupta, "Exploring LLM-based Agentic Frameworks for Fault Diagnosis," Annual Conference of the PHM Society 17, 1 (2025).

26. Wade A. Smith and Robert B. Randall, "Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study," Mechanical Systems and Signal Processing 64-65, 100-131 (2015).

27. Christian Lessmeier, James Kuria Kimotho, Detmar Zimmer, and Walter Sextro, "Condition Monitoring of Bearing Damage in Electromechanical Drive Systems by Using Motor Current Signals of Electric Motors: A Benchmark Data Set for Data-Driven Classification," PHM Society European Conference 3, 1 (2016).

28. Shiyu Shao, Stephen McAleer, Ruqiang Yan, et al., "Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning," IEEE Transactions on Industrial Informatics 15, 4, 2446-2455 (2019).

29. Mert Sehri and Patrick Dumond, "University of Ottawa Rolling-element Dataset: Vibration and Acoustic Faults under Constant Load and Speed Conditions (UORED-VAFCLS)," Mendeley Data V5 (2023).

10.37965/jdmd.2026.1588

30. Eric Bechhoefer, "Condition Based Maintenance Fault Database for Testing Diagnostics and Prognostic Algorithms," MFPT Data (2013).

31. Ao Ding, Yong Qin, Biao Wang, et al., "Evolvable graph neural network for system-level incremental fault diagnosis of train transmission systems," Mechanical Systems and Signal Processing 210, 111175 (2024).