1-1-2024

Conditional variational transformer for bearing remaining useful life prediction

Yupeng Wei
San Jose State University, yupeng.wei@sjsu.edu

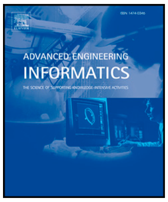
Dazhong Wu
University of Central Florida

Recommended Citation
Yupeng Wei and Dazhong Wu. "Conditional variational transformer for bearing remaining useful life prediction" *Advanced Engineering Informatics* (2024). https://doi.org/10.1016/j.aei.2023.102247

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei



Full length article

Conditional variational transformer for bearing remaining useful life prediction

Yupeng Wei [a,*], Dazhong Wu [b]

[a] *Department of Industrial and Systems Engineering, San Jose State University, San Jose, 95192, CA, USA*

[b] *Department of Mechanical and Aerospace Engineering, University of Central Florida, Orlando, 32816, FL, USA*

ARTICLE INFO

Keywords:
Bearing
Remaining useful life
Variational inference
Transformer
Deep learning

ABSTRACT

Transformer, built on the self-attention mechanism, has been demonstrated to be effective in numerous applications. However, in the context of prognostics and health management, the self-attention mechanism in the Transformer is not effective in selecting the most important features that are highly correlated with the remaining useful life (RUL) of a component. To address this issue, we developed a novel conditional variational transformer architecture consisting of four networks: two generative networks and two predictive networks. The first generative network uses the transformer encoder–decoder as well as both condition monitoring data and RUL as input to extract the most important features in one feature space from condition monitoring data. The second generative network uses the transformer encoder and condition monitoring data to extract features in another feature space. The two predictive networks use the extracted features in two different feature spaces to make predictions. A KL-divergence is used to minimize the distance between the two feature spaces learned by the first and second generative networks so that the feature space extracted from the second generative network can approximate the feature space extracted from the first generative network. We demonstrated that the proposed method is effective in predicting the RUL of bearings using two datasets.

1. Introduction

Rotating machinery refers to a broad range of machines that rotate around an axis, such as turbines, generators, pumps, compressors, and engines [1]. These machines have a wide range of applications in many fields, including power generation, oil and gas, transportation, and manufacturing [2]. Turbines generate mechanical energy from fluid energy by rotating blades, while generators produce electrical energy from mechanical energy through a rotor rotating within a stator [3]. Pumps and compressors, on the other hand, transfer fluids or gases using rotating impellers or pistons. Bearings play a crucial role in ensuring the smooth and efficient operation of rotating machinery by supporting and facilitating shaft rotation through a low-friction interface between a shaft and its housing, reducing wear and tear on machine components [4]. However, bearings degrade over time due to high loads, inadequate lubrication, contamination, or misalignment [5]. Bearing degradation can lead to excessive vibration and noise, low efficiency, and high energy consumption, and if left unchecked, can result in catastrophic failures that cause costly repairs, machine downtime, and potential safety hazards [6]. Therefore, it is critical to monitor and predict the health condition and remaining useful life (RUL) of bearings to reduce machine downtime.

Over the past decade, new sensing technology allows one to monitor the health condition of bearings more effectively. With more cost-effective sensors, data-driven methods have become widely adopted for predicting the RUL of bearings, as they can perform prognostics using real-time condition monitoring data without prior knowledge. These data-driven methods can be grouped into two classes: traditional machine learning and deep learning methods. Traditional machine learning methods include support vector regression [7], Gaussian process [8], ensemble learning [9], Markov model [10], and others. For instance, Wang et al. [11] introduced a multi-support vector regression approach to acquire optimized sub-model parameters for RUL predictions of bearings. After obtaining the optimized model parameters, an automatic weight updating mechanism was proposed to assess the appropriateness of each sub-model for more robust prediction performance. A publicly accessible dataset was used to assess the prediction performance of the presented method, and the results reflected that the RMSE of prediction was 14.98 cycles. Meng et al. [12] integrated the gray Markov model with the fractal spectrum theory to track the degradation trajectory of rolling bearings. The presented method was compared to the generalized mathematical morphology particle characteristic, showing that the presented method can reduce the RMSE

[*] Corresponding author.
E-mail address: yupeng.wei@sjsu.edu (Y. Wei).

https://doi.org/10.1016/j.aei.2023.102247
Received 4 August 2023; Received in revised form 27 October 2023; Accepted 3 November 2023
Available online 20 November 2023

by 4%. Wang et al. [13] combined a feature-level fusion approach with an ensemble learning approach to fully identify the deep representation of condition monitoring data to estimate the RUL of bearings. In the presented ensemble learning approach, a diversity of base learners were aggregated to boost the RUL prediction performance. The results showed that combining feature-level fusion with ensemble learning outperforms solely using ensemble learning approaches.

Traditional machine learning methods are not effective in learning complex and nonlinear relationships from condition monitoring data. To address this issue, in recent years, deep learning methods have become increasingly popular to predict the RUL due to their ability to learn complex patterns and handle large volumes of data. These deep learning methods can be broadly categorized into two subgroups: recurrent and non-recurrent networks. The non-recurrent deep learning techniques include convolutional neural network (CNN) [14], artificial neural network (ANN) [15], autoencoders [16], graph neural network (GNN) [17,18], and generative models such as variational autoencoder (VAE) [19] and generative adversarial network (GAN) [20]. As an example, Zhu et al. [21] proposed a multiscale CNN to extract deep representations from condition monitoring data, and the extracted deep representations were combined with time–frequency attributes for RUL prediction of bearings. The multiscale CNN was able to preserve global and local representations in comparison with traditional CNN approaches, and the results reflected that the presented multiscale CNN enhances prediction performance. Xu et al. [16] presented a convolutional autoencoder to extract features from condition monitoring data collected from the depreciated rolling bearings, and a health index scaling function was adopted to downscale the extracted features. The results showed that the presented method is efficient in predicting the RUL and evaluating the degradation stages of rolling bearings. Yang et al. [22] utilized a GNN to estimate the RUL of bearings, where regression shapelet was first adopted to build graphs of the condition monitoring data, and GNN was adopted to handle the topological structures of the built graphs. Suh et al. [23] employed a GAN to generate multiscale features for estimating the RUL of bearings. To capture sequence patterns in one-dimensional vibration signals, they introduced a U-Net architecture. The results revealed that the proposed technique extracted effective features to improve prediction accuracy.

While many studies have explored the use of non-recurrent neural networks for bearing RUL prediction, these methods are often not effective in dealing with time series data because they are not able to recognize the sequential characteristics of condition monitoring data. By contrast, recurrent deep learning algorithms such as recurrent neural network (RNN) [24], long short-term memory (LSTM) [25,26], gated recurrent network (GRU) [27], and their bidirectional versions such as bidirectional LSTM and bidirectional GRU [28] have been demonstrated to be more effective in estimating the RUL of bearings. As an example, Ma et al. [29] introduced a deep convolutional LSTM for bearing RUL prediction. The convolutional operation in the proposed LSTM cell was capable of extracting time–frequency features and preserving long-term dependencies simultaneously. Numerical results have shown that the proposed convolutional LSTM outperforms the deep CNN and LSTM cell. Zhang et al. [30] developed a parallel hybrid deep learning method that integrates 1D CNN and bidirectional GRU for real-time RUL predictions, enabling the parallel extraction of spatial and temporal features from condition monitoring data. Han et al. [31] combined the stacked autoencoder and RNN to estimate the remaining lifetime of bearings. The stacked autoencoder was adopted to fuse features into health indices, and RNN was then implemented to construct the predicted model. Moreover, to deal with the issue brought by insufficient data, the spline curve interpolation was adopted to increase the model accuracy and robustness.

Although recurrent deep learning techniques have been widely used for bearing RUL predictions, they are not effective in capturing the long-term dependency or memory in condition monitoring data [32]. To address this issue, attention mechanisms, especially self-attention mechanisms [33], are increasingly adopted. The self-attention mechanism includes three components: the query, the key, and the value, which are generated using the condition monitoring data. The self-attention mechanism computes a score for each key–value pair based on how well it matches the query, and the scores are then converted into probabilities using a softmax function. These probabilities are used to weight the values, so that the predictive model can utilize the most significant features of the condition monitoring data to make predictions. Among the deep learning methods that use the self-attention mechanism, transformer is one of the most powerful and effective algorithms. The transformer architecture combines attention with other features such as positional encoding and feed-forward neural networks to reveal the nonlinear correlation in condition monitoring data, thereby significantly improving prediction performance. For instance, Su et al. [34] employed the transformer encoder to predict the RUL of bearings. Their proposed method included two stages, where the first stage extracted low-level features using a feature extraction mechanism, and the second stage utilized the transformer encoder to estimate the RUL. Two publicly available datasets were utilized to verify the effectiveness of the presented method, and results have shown that the transformer encoder leads to an increased prediction performance. Ding et al. [35] introduced a convolutional transformer that integrates the convolutional operation and the self-attention mechanism to estimate the RUL of bearings. The convolutional operation was used to reveal the local dependencies in condition monitoring data, and the self-attention mechanism was adopted to reveal the global dependencies of condition monitoring data. Zhang et al. [36] proposed a novel Transformer model to predict the RUL, where a multi-head dual sparse self-attention mechanism was introduced to improve the computational efficiency. Experimental results have shown that the proposed method outperforms conventional Transformer and other data-driven methods.

While several studies have shown that the transformer model is effective in predicting the RUL of bearings [37], the current transformer model that uses the self-attention mechanism also has limitations. For example, the self-attention mechanism relies solely on the condition monitoring data to generate queries, keys, and values. These queries and keys are used to create an attention matrix, which is multiplied by the generated value to extract features for RUL prediction. Because both queries and keys do not include any information about the RUL, the attention matrix does not necessarily extract the most important features that are highly correlated with the RUL. To address the limitation of the self-attention mechanism in the conventional transformer models, we developed a novel conditional variational transformer architecture that consists of four networks: two generative networks and two predictive networks. Both generative networks learn deep-level representations in two different feature spaces of the condition monitoring data, while the predictive networks use these representations to predict RUL. The first generative network uses a transformer encoder–decoder architecture to learn deep-level representations in one feature space of the condition monitoring data. The inputs of the transformer encoder are the condition monitoring data, while the inputs of the transformer decoder are the true RUL of bearings. We also introduced a cross-attention mechanism in the first generative network such that queries are generated using the true RUL data, while keys and values are generated using condition monitoring data. The cross-attention mechanism allows one to construct a more effective attention matrix to extract the most important features that are highly correlated with the true RUL. The deep-level representations in one feature space of the condition monitoring data learned by the first generative network are then fed into the first predictive network to predict RUL. One issue with the cross-attention mechanism is that the true RUL data are not available in the testing phase, although the true RUL data are available in the training phase. To address this issue, we introduced the second generative network that uses the transformer encoder architecture with only condition monitoring data as input to learn deep-level representations in another feature space. The learned representations in another feature space are then fed into the second predictive network to predict the RUL. Then, we used a Kullback–Leibler (KL) divergence to minimize the distance between two feature spaces so that the feature space extracted from the second generative network can approximate the feature space extracted from the first generative network. Therefore, even without the true RUL, the feature space extracted from the first generative network can be taken into account when the feature space extracted from the second generative network is used for testing.

We also introduced a two-stage training process to train the proposed conditional variational transformer. In the first training stage, we trained both generative networks and both predictive networks by minimizing two prediction losses and the KL-divergence loss simultaneously. In the second training stage, we employed a fine-tuning mechanism to tune the parameters in the second predictive network by minimizing a single prediction loss only. The first training stage mainly aims at minimizing the distance between two feature spaces generated by two generative networks, and the second training stage aims at minimizing the prediction loss so that the prediction performance can be optimized. The contributions of this work are summarized as follows:

- Two generative networks were introduced to learn deep-level features in two different feature spaces from condition monitoring data, where the first generative network involves a cross-attention mechanism to select the most important features that are highly correlated with the RUL of a bearing. Two predictive networks were introduced to use the learned features to predict the RUL.
- The KL-divergence was introduced to minimize the distance between two feature spaces, allowing the feature space extracted from the second generative network to approximate the feature space extracted from the first generative network.
- A two-stage training process was introduced to train a predictive model. In the first stage, both generative networks and both predictive networks were trained. In the second stage, a fine-tuning mechanism was adopted to optimize the parameters in the second predictive network.

The remaining sections of this work are organized as follows: Section 2 presents the theoretical background and architecture of the proposed conditional variational transformer. Sections 3 and 4 utilize two publicly available datasets to demonstrate the efficiency of the presented conditional variational transformer. Section 5 provides a summary of this work and discussions on future work.

2. Conditional variational transformer

In this section, the conditional variational transformer is introduced. First, the theoretical background of the conditional variational transformer is presented. Second, the architecture of the proposed conditional variational transformer is detailed. Last, the two-stage training process for the proposed conditional variational transformer is presented.

2.1. Conditional variational inference

To reduce computational costs, we begin by extracting features from the condition monitoring data of bearings in both time and frequency domains. We then sample these features using a sliding window of size $\mathscr{S}$. The tth sampled features for bearing unit i at time t is represented by $\mathbf{X}_{i,t} \in \mathbb{R}^{\mathscr{S}\times\mathscr{F}}$, where $\mathscr{F}$ denotes the number of extracted features. In the context of prognostics health management, most predictive models aim to map the distribution of extracted features $\mathbf{X}_{i,t}$ to the true RUL $y_{i,t}$. Mathematically, this can be expressed as Eq. (1),

$$\max_{\Phi} \mathbb{E}\left[\log p_{\Phi}\left(y_{i,t}|\mathbf{X}_{i,t}\right)\right] \tag{1}$$

where $y_{i,t}$ represents the ground truth of RUL for bearing i at time t, while Φ denotes the parameters used to map $\mathbf{X}_{i,t}$ to $y_{i,t}$. The notation $\mathbb{E}[\cdot]$ denotes the expectation. The objective of Eq. (1) is to maximize the expectation of the log-likelihood of the true RUL given the extracted features, so that the distribution of $\mathbf{X}_{i,t}$ can be mapped to $y_{i,t}$. By applying Bayes' theorem, we can re-express Eq. (1) as Eq. (2).

$$\max_{\Phi} \mathbb{E}\left[\log \frac{p_{\Phi}\left(\mathbf{X}_{i,t}, y_{i,t}\right)}{p_{\Phi}\left(\mathbf{X}_{i,t}\right)}\right] \tag{2}$$

Most deep learning-based predictive models for RUL prediction of bearings use $\mathbf{X}_{i,t}$ to learn deep-level representations $\mathbf{F}_{i,t}$, and subsequently employ $\mathbf{F}_{i,t}$ to predict the RUL $y_{i,t}$ of the bearings [38]. Consequently, we can express the joint probability distribution of $\mathbf{X}_{i,t}$, $\mathbf{F}_{i,t}$, and $y_{i,t}$ as Eq. (3).

$$p_{\Phi}\left(\mathbf{X}_{i,t}, \mathbf{F}_{i,t}, y_{i,t}\right) = p_{\Phi}\left(y_{i,t}|\mathbf{F}_{i,t}\right) \cdot p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right) \cdot p_{\Phi}\left(\mathbf{X}_{i,t}\right) \tag{3}$$

We can use Bayes' theorem to express the joint distribution of $\mathbf{X}_{i,t}$ and $y_{i,t}$ as Eq. (4).

$$p_{\Phi}\left(\mathbf{X}_{i,t}, y_{i,t}\right) = \frac{p_{\Phi}\left(\mathbf{X}_{i,t}, \mathbf{F}_{i,t}, y_{i,t}\right)}{p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)} = \frac{p_{\Phi}\left(y_{i,t}|\mathbf{F}_{i,t}\right) \cdot p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right) \cdot p_{\Phi}\left(\mathbf{X}_{i,t}\right)}{p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)} \tag{4}$$

By incorporating Eq. (4) into Eq. (2), Eq. (5) can be obtained.

$$\max_{\Phi} \mathbb{E}\left[\log \frac{p_{\Phi}\left(y_{i,t}|\mathbf{F}_{i,t}\right) \cdot p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right)}{p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)}\right] \tag{5}$$

As the conditional posterior inference $p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)$ is intractable, it is commonly accepted that a variational inference method can be used to approximate this posterior inference. This leads us to Eq. (6),

$$\max_{\Phi,\Pi} \mathbb{E}\left[\log \frac{p_{\Phi}\left(y_{i,t}|\mathbf{F}_{i,t}\right) \cdot p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right)}{p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)} \cdot \frac{q_{\Pi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)}{q_{\Pi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)}\right] \tag{6}$$

where $q_{\Pi}(\cdot)$ denotes the variational inference, while Π refers to the set of parameters in the variational inference. We can then rewrite Eq. (6) as Eq. (7), where $\mathbb{KL}(\cdot)$ represents the KL-divergence between two distributions.

$$\begin{aligned}\max_{\Phi,\Pi} \mathbb{E}_{\mathbf{F}_{i,t}\sim q_{\Pi}} &\left[\log \frac{p_{\Phi}\left(y_{i,t}|\mathbf{F}_{i,t}\right) \cdot p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right)}{q_{\Pi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)}\right] \\ &+ \mathbb{KL}\left(q_{\Pi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right) || p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)\right)\end{aligned} \tag{7}$$

Since the KL-divergence cannot be negative and the KL-divergence term in Eq. (7) is difficult to handle, it is common practice to maximize the evidence lower bound (ELBO) of Eq. (7) instead of maximizing the original optimization problem [39]. The ELBO of Eq. (7) can be represented as Eq. (8), where the KL-divergence term has been removed.

$$\text{ELBO} := \max_{\Phi,\Pi} \mathbb{E}_{\mathbf{F}_{i,t}\sim q_{\Pi}}\left[\log \frac{p_{\Phi}\left(y_{i,t}|\mathbf{F}_{i,t}\right) \cdot p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right)}{q_{\Pi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)}\right] \tag{8}$$

We can express this ELBO as the expectation of the log-likelihood of the true RUL given the learned deep-level representations and the KL-divergence of two distributions. This is represented as Eq. (9).

$$\begin{aligned}\text{ELBO} := \max_{\Phi,\Pi} \mathbb{E}_{\mathbf{F}_{i,t}\sim q_{\Pi}} &\left[\log p_{\Phi}\left(y_{i,t}|\mathbf{F}_{i,t}\right)\right] \\ &- \mathbb{KL}\left(q_{\Pi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right) || p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right)\right)\end{aligned} \tag{9}$$

The introduced ELBO consists of two terms: the expectation term and the KL-divergence term. The expectation term aims to maximize the expectation of the log-likelihood of the true RUL $y_{i,t}$ given the deep-level representations $\mathbf{F}_{i,t}$ learned from $\mathbf{X}_{i,t}$. The KL-divergence term aims to minimize the gap between the posterior inference $p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right)$ and the variational inference $q_{\Pi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)$. To approximate the three probability distributions in Eq. (9), any sophisticated neural networks can be adopted based on the Universal Approximation Theorem [40]. In this work, we use transformer encoders and decoders for this purpose.

2.2. Conditional variational transformer architecture

The conditional probability distributions in the ELBO are approximated with transformer encoders and decoders due to their superior predictive capability. In the proposed architecture, one transformer encoder is used to approximate $p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right)$, one transformer encoder–decoder is used to approximate $q_{\Pi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)$, and two feedforward artificial neural networks are used to approximate $p_{\Phi}\left(y_{i,t}|\mathbf{F}_{i,t}\right)$. The following subsections will introduce the details of these approximations.

2.2.1. Approximate $p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right)$ with transformer encoder

The conditional probability distribution $p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right)$ is approximated using a transformer encoder network. The input to this transformer encoder network is the sampled feature matrix $\mathbf{X}_{i,t}$ for all bearing units i and time t, and the output is the learned deep-level representations $\mathbf{F}_{i,t}$. We refer to this transformer encoder network as the generative encoder network and denote it as $\mathbb{E}_p$ since its purpose is to generate the deep-level representations $\mathbf{F}_{i,t}$. To leverage the time dependency of $\mathbf{X}_{i,t}$, the transformer encoder network starts with positional encoding, as the transformer encoder does not include any recurrent attributes. Here, we employ conventional sine and cosine functions with distinct frequencies as the positional encoding function, as this approach enables the model to extend its capability to handle sequences of greater length compared to those encountered in the training data [41]. The positional encoding function is given by Eq. (10),

$$\begin{aligned} PE_{(pos,2j)} &= \sin\left(pos/10000^{2j/d_{model}}\right) \\ PE_{(pos,2j+1)} &= \cos\left(pos/10000^{2j/d_{model}}\right) \end{aligned} \tag{10}$$

where pos denotes the position, j is the dimension, d_{model} refers to feature size, which can be set as $d_{model} = \mathscr{F}$. Next, the positional encoding is added to the sampled feature matrices $\mathbf{X}_{i,t}$, which can be expressed as Eq. (11).

$$\mathbf{X}_{i,t}^{(PE)} = \mathbf{X}_{i,t} + PE_{i,t} \tag{11}$$

In Eq. (11), $PE_{i,t}$ refers to the positional encoding for the tth sampled feature matrix of bearing unit i. The sampled feature matrices with positional encoding are then inputted into a multi-head self-attention mechanism to select the most important features of the sampled $\mathbf{X}_{i,t}$. A single-head self-attention mechanism is represented mathematically as Eq. (12), where $\mathbf{W}_Q^{(h,\mathbb{E}_p)} \in \mathbb{R}^{\mathscr{F}\times\mathscr{D}}$ refers to a parameter matrix used to generate queries in the generative encoder network $\mathbb{E}_p$ for the hth head, $\mathbf{W}_K^{(h,\mathbb{E}_p)} \in \mathbb{R}^{\mathscr{F}\times\mathscr{D}}$ refers to a parameter matrix used to generate keys in $\mathbb{E}_p$ for the hth head, $\mathbf{W}_V^{(h,\mathbb{E}_p)} \in \mathbb{R}^{\mathscr{F}\times\mathscr{D}}$ refers to a parameter matrix used to generate values in $\mathbb{E}_p$ for the hth head, and $\mathbf{A}_{i,t}^{(h,\mathbb{E}p)} \in \mathbb{R}^{\mathscr{S}\times\mathscr{S}}$ refers to the learned attention matrix in $\mathbb{E}_p$ for the hth head.

$$\begin{aligned} \mathbf{A}_{i,t}^{(h,\mathbb{E}_p)} &= \text{Softmax}\left(\mathbf{X}_{i,t}^{(PE)}\mathbf{W}_Q^{(h,\mathbb{E}_p)}\left(\mathbf{X}_{i,t}^{(PE)}\mathbf{W}_K^{(h,\mathbb{E}_p)}\right)^T/\sqrt{\mathscr{D}}\right) \\ \mathbf{O}_{i,t}^{(h,\mathbb{E}_p)} &= \left(\mathbf{X}_{i,t}^{(PE)}\mathbf{W}_V^{(h,\mathbb{E}_p)}\right)\cdot\mathbf{A}_{i,t}^{(h,\mathbb{E}_p)} \end{aligned} \tag{12}$$

Then, the multi-head self-attention mechanism is mathematically represented as Eq. (13),

$$\mathbf{O}_{i,t}^{(\mathbb{E}_p)} = \left(||_{h=1}^{H}\mathbf{O}_{i,t}^{(h,\mathbb{E}_p)}\right)\cdot\mathbf{W}_O^{(\mathbb{E}_p)} \tag{13}$$

where H denotes the quantity of heads in the multi-head self-attention mechanism, $||$ denotes the concatenation operator, and $\mathbf{W}_O^{(\mathbb{E}_p)}$ is a parameter matrix to project the resulting tensors after performing concatenation. Next, the resulting tensor $\mathbf{O}_{i,t}^{(\mathbb{E}_p)}$ are fed into a normalization layer to update the obtained $\mathbf{O}_{i,t}^{(\mathbb{E}_p)}$, such a layer includes both residual connection and normalization, which can be represented as Eq. (14).

$$\mathbf{O}_{i,t}^{(\mathbb{E}_p)} = \text{LayerNorm}\left(\mathbf{O}_{i,t}^{(\mathbb{E}_p)} + \mathbf{X}_{i,t}^{(PE)}\right) \tag{14}$$

Next, the updated $\mathbf{O}_{i,t}^{(\mathbb{E}_p)}$ is passed through two parallel feedforward layers, followed by two parallel normalization layers, to generate the mean $\mu_{i,t}^{(1)}$ and variance $\Sigma_{i,t}^{(1)}$ of the probability distribution $p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right)$. This process can be mathematically represented as Eq. (15), where $\mathbf{W}_1^{(\mathbb{E}_p)}$ and $\mathbf{W}_2^{(\mathbb{E}_p)}$ are the kernel weight matrices in the two feedforward layers used to generate $\mu_{i,t}^{(1)}$ and $\Sigma_{i,t}^{(1)}$, respectively. Similarly, $\mathbf{b}_1^{(\mathbb{E}p)}$ and $\mathbf{b}_2^{(\mathbb{E}p)}$ are the bias weight vectors in the two feedforward layers used to generate $\mu_{i,t}^{(1)}$ and $\Sigma_{i,t}^{(1)}$, respectively. The rectified linear unit activation function is denoted by the term Relu.

$$\begin{aligned} \mu_{i,t}^{(1)} &= \text{LayerNorm}\left(\text{Relu}\left(\mathbf{W}_1^{(\mathbb{E}_p)}\mathbf{O}_{i,t}^{(\mathbb{E}_p)} + \mathbf{b}_1^{(\mathbb{E}_p)}\right) + \mathbf{O}_{i,t}^{(\mathbb{E}_p)}\right) \\ \Sigma_{i,t}^{(1)} &= \text{LayerNorm}\left(\text{Relu}\left(\mathbf{W}_2^{(\mathbb{E}_p)}\mathbf{O}_{i,t}^{(\mathbb{E}_p)} + \mathbf{b}_2^{(\mathbb{E}_p)}\right) + \mathbf{O}_{i,t}^{(\mathbb{E}_p)}\right) \end{aligned} \tag{15}$$

Here, it is commonly presumed that $p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right)$ follows a normal distribution, which can be denoted as Eq. (16).

$$p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right) \sim \mathcal{N}\left(\mathbf{F}_{i,t}; \mu_{i,t}^{(1)}, \Sigma_{i,t}^{(1)}\right) \tag{16}$$

Next, we use the generated $\mu_{i,t}^{(1)}$ and $\Sigma_{i,t}^{(1)}$ to sample the deep-level representations $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$ for the generative encoder network $\mathbb{E}_p$, which can be expressed as Eq. (17), where ξ denotes a random variable that conforms to a multivariate normal distribution characterized by a zero mean and unit variance.

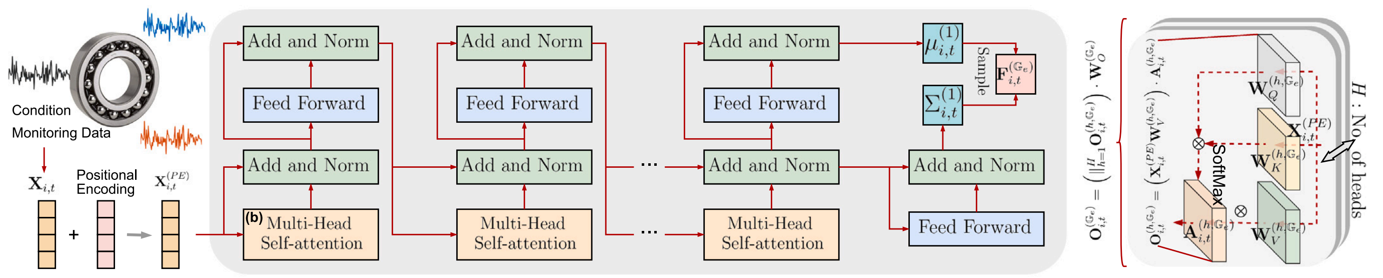
$$\mathbf{F}_{i,t}^{(\mathbb{E}_p)} = \mu_{i,t}^{(1)} + \Sigma_{i,t}^{(1)} \odot \xi, \quad \xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{17}$$

Fig. 1 illustrates the architecture of the transformer encoder that utilizes multiple encoder layers to generate deep-level representations $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$. In the first encoder layer, the feature matrices $\mathbf{X}_{i,t}$ for all i and t are inputted into the multi-head attention mechanism to extract the most important features. The extracted features are then projected into a different space using a feedforward layer, and residual connections and normalization are applied to increase model robustness. The proposed transformer encoder stacks such layers multiple times. The last two parallel feedforward layers are followed by two parallel normalization layers to generate the mean $\mu_{i,t}^{(1)}$ and the variance $\Sigma_{i,t}^{(1)}$. The generated mean and variance are utilized to sample $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$. By following the above process, we can generate the deep-level representations $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$ to approximate $p_{\Phi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}\right)$.

2.2.2. Approximate $q_{\Pi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)$ with transformer encoder–decoder

The conditional probability distribution $q_{\Pi}\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)$ is approximated using a transformer encoder–decoder network. The input to the transformer encoder are the sampled feature matrices $\mathbf{X}_{i,t}$ for all bearing units i and all time steps t. Based on the condition given in the conditional probability, the input to the transformer decoder is the ground truth of RUL $y_{i,t}$ for all units i at time t. To better use the true RUL to select the most relevant features for making predictions, we use a vector $\mathbf{y}_{i,t}$ as the input to the transformer decoder. This vector can be mathematically represented as $\mathbf{y}_{i,t} = \left(y_{i,t-\mathscr{S}+1}, \ldots, y_{i,t}\right)$. The output of this transformer encoder–decoder network is also the learned deep-level representations $\mathbf{F}_{i,t}$. We designate this transformer encoder–decoder network as the generative encoder–decoder network ($\mathbb{ED}_q$), as it also intends to generate the deep-level representations. We denote the transformer encoder as $\mathbb{E}_q$ and the transformer decoder as $\mathbb{D}_q$.

The transformer encoder $\mathbb{E}_q$ is similar to the transformer encoder $\mathbb{E}_p$, with the exception that $\mathbb{E}_q$ does not generate mean and variance vectors. Specifically, each encoder layer of $\mathbb{E}_q$ consists of four sublayers that are connected to each other: a multi-head self-attention layer, a residual connection and normalization layer, a feedforward layer, and another residual connection and normalization layer. The outputs of the transformer encoder $\mathbb{E}_q$ are denoted as $\mathbf{O}_{i,t}^{(\mathbb{E}_q)}$, which are obtained by applying Eq. (10) to Eq. (14) repeatedly. The outputs of the transformer encoder $\mathbb{E}_q$ are subsequently inputted into the transformer decoder network $\mathbb{D}_q$. The transformer decoder network $\mathbb{D}_q$ also starts with positional encoding, which makes use of the order of the sequence by using sine and cosine functions with distinct frequencies as the


Condition Monitoring Data
$\mathbf{X}_{i,t}$
Positional Encoding
$\mathbf{X}_{i,t}^{(PE)}$
+
Add and Norm
Feed Forward
Add and Norm
(b) Multi-Head Self-attention
···
$\mu_{i,t}^{(1)}$
$\Sigma_{i,t}^{(1)}$
Sample
$\mathbf{F}_{i,t}^{(\mathbb{G}_e)}$
$\mathbf{O}_{i,t}^{(\mathbb{G}_e)} = \left(\|_{h=1}^{H}\mathbf{O}_{i,t}^{(h,\mathbb{G}_e)}\right)\cdot\mathbf{W}_O^{(\mathbb{G}_e)}$
$\mathbf{O}_{i,t}^{(h,\mathbb{G}_e)} = \left(\mathbf{X}_{i,t}^{(PE)}\mathbf{W}_V^{(h,\mathbb{G}_e)}\right)\cdot\mathbf{A}_{i,t}^{(h,\mathbb{G}_e)}$
$\mathbf{W}_Q^{(h,\mathbb{G}_e)}$
$\mathbf{X}_{i,t}^{(PE)}$
$\mathbf{W}_K^{(h,\mathbb{G}_e)}$
$\mathbf{W}_V^{(h,\mathbb{G}_e)}$
$\mathbf{A}_{i,t}^{(h,\mathbb{G}_e)}$
SoftMax
H : No. of heads

(a) Transformer Encoder with multiple encoder layers

(b) Multi-Head Self-attention

Fig. 1. (a) The architecture of the transformer encoder with multiple encoder layers that is utilized to generate deep-level representations $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$; (b) multi-head self-attention mechanism.

positional encoding function. The positional encoding is added to the true RUL vectors $\mathbf{y}_{i,t}$, which can be represented as Eq. (18).

$$\mathbf{y}_{i,t}^{(PE)} = \mathbf{y}_{i,t} + PE_{i,t} \tag{18}$$

Next, the vectors $\mathbf{y}_{i,t}^{(PE)}$ for all bearings i at time t are fed into the multi-head self-attention mechanism using Eqs. (12) and (13), and the resulting outputs are denoted as $\mathbf{y}'_{i,t}$. Next, the resulting tensor $\mathbf{y}'_{i,t}$ is fed into a normalization layer, which includes both residual connection and normalization and can be denoted as Eq. (19).

$$\mathbf{y}'_{i,t} = \text{LayerNorm}\left(\mathbf{y}'_{i,t} + \mathbf{y}_{i,t}^{(PE)}\right) \tag{19}$$

Next, the outputs $\mathbf{y}'_{i,t}$ and $\mathbf{O}_{i,t}^{(\mathbb{E}q)}$ are fed into the multi-head cross-attention mechanism, where $\mathbf{y}'_{i,t}$ is employed to generate query, and $\mathbf{O}_{i,t}^{(\mathbb{E}_q)}$ is used to generate both keys and values. With using the $\mathbf{y}'_{i,t}$ learned from the true RUL to generate query, the prediction model is capable of selecting the most important features that is highly correlated with the true RUL to make predictions. A single head cross-attention mechanism can be mathematically represented as Eq. (20), where $\mathbf{W}_Q^{(h,\mathbb{D}_q)}$ refers to a parameter matrix to generate query for the hth head in $\mathbb{D}_q$, $\mathbf{W}_K^{(h,\mathbb{D}_q)}$ refers to a parameter matrix to generate key for the hth head in $\mathbb{D}_q$, $\mathbf{W}_V^{(h,\mathbb{D}_q)}$ refers to a parameter matrix to generate value for the hth head in $\mathbb{D}_q$, and $\mathbf{A}_{i,t}^{(h,\mathbb{D}_q)}$ refers to the learned cross-attention matrix in $\mathbb{D}_q$ for head h.

$$\begin{aligned}\mathbf{A}_{i,t}^{(h,\mathbb{D}_q)} &= \text{Softmax}\left(\mathbf{y}'_{i,t}\mathbf{W}_Q^{(h,\mathbb{D}_q)}\left(\mathbf{O}_{i,t}^{(\mathbb{E}_q)}\mathbf{W}_K^{(h,\mathbb{D}_q)}\right)^T/\sqrt{\mathscr{D}}\right)\\ \mathbf{O}_{i,t}^{(h,\mathbb{D}_q)} &= \left(\mathbf{O}_{i,t}^{(\mathbb{E}_q)}\mathbf{W}_V^{(h,\mathbb{D}_q)}\right)\cdot\mathbf{A}_{i,t}^{(h,\mathbb{D}_q)}\end{aligned} \tag{20}$$

The multi-head cross-attention mechanism is mathematically represented as Eq. (21), where $\mathbf{W}_O^{(\mathbb{D}_q)}$ is a parameter matrix used to project the concatenated output tensors.

$$\mathbf{O}_{i,t}^{(\mathbb{D}_q)} = \left(\|_{h=1}^{H}\mathbf{O}_{i,t}^{(h,\mathbb{D}_q)}\right)\cdot\mathbf{W}_O^{(\mathbb{D}_q)} \tag{21}$$

After the multi-head cross-attention mechanism is employed, the resulting tensor $\mathbf{O}_{i,t}^{(\mathbb{D}_q)}$ is normalized to update the obtained values. These updated values are then passed through two parallel feedforward layers, followed by two parallel normalization layers, to generate the mean $\mu_{i,t}^{(2)}$ and variance $\Sigma_{i,t}^{(2)}$ of the probability distribution $q_\Pi\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)$. This generation process can be represented mathematically as Eq. (22), where $\mathbf{W}_1^{(\mathbb{D}_q)}$ and $\mathbf{W}_2^{(\mathbb{D}q)}$ are the kernel weight matrices used in the two feedforward layers to generate $\mu_{i,t}^{(2)}$ and $\Sigma_{i,t}^{(2)}$, respectively, while $\mathbf{b}_1^{(\mathbb{D}_q)}$ and $\mathbf{b}_2^{(\mathbb{D}q)}$ are the bias weight vectors used in the two feedforward layers to generate $\mu_{i,t}^{(2)}$ and $\Sigma_{i,t}^{(2)}$, respectively.

$$\begin{aligned}\mu_{i,t}^{(2)} &= \text{LayerNorm}\left(\text{Relu}\left(\mathbf{W}_1^{(\mathbb{D}_q)}\mathbf{O}_{i,t}^{(\mathbb{D}_q)} + \mathbf{b}_1^{(\mathbb{D}_q)}\right) + \mathbf{O}_{i,t}^{(\mathbb{D}_q)}\right)\\ \Sigma_{i,t}^{(2)} &= \text{LayerNorm}\left(\text{Relu}\left(\mathbf{W}_2^{(\mathbb{D}_q)}\mathbf{O}_{i,t}^{(\mathbb{D}_q)} + \mathbf{b}_2^{(\mathbb{D}_q)}\right) + \mathbf{O}_{i,t}^{(\mathbb{D}_q)}\right)\end{aligned} \tag{22}$$

Here, we also assume that the probability distribution $q_\Pi\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)$ follows a normal distribution, which can be written mathematically as Eq. (23).

$$q_\Pi\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right) \sim \mathcal{N}\left(\mathbf{F}_{i,t}; \mu_{i,t}^{(2)}, \Sigma_{i,t}^{(2)}\right) \tag{23}$$

After generating $\mu_{i,t}^{(2)}$ and $\Sigma_{i,t}^{(2)}$, we use these values to sample the deep-level representations $\mathbf{F}_{i,t}^{(\mathbb{ED}_q)}$ for the generative encoder–decoder network $\mathbb{ED}_q$. This process can be represented mathematically as Eq. (24).

$$\mathbf{F}_{i,t}^{(\mathbb{ED}_q)} = \mu_{i,t}^{(2)} + \Sigma_{i,t}^{(2)} \odot \xi, \quad \xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{24}$$

Fig. 2 illustrates the architecture of the transformer encoder–decoder network, which includes multiple encoder and decoder layers used to generate deep-level representations $\mathbf{F}_{i,t}^{(\mathbb{ED}_q)}$. In each encoder layer, four operations are performed sequentially: multi-head self-attention, residual connection and normalization, feed-forward operation, and another residual connection and normalization. These layers are stacked to obtain the resulting tensor $\mathbf{O}_{i,t}^{(\mathbb{E}_q)}$, which is then used in the multi-head cross-attention mechanism in each decoder layer to generate corresponding keys and values. Each decoder layer includes six operations, performed sequentially: multi-head self-attention, residual connection and normalization, multi-head cross-attention, residual connection and normalization, feed-forward operation, and another residual connection and normalization. Specifically, within each multi-head cross-attention mechanism, $\mathbf{O}_{i,t}^{(\mathbb{E}_q)}$ is used to generate keys and values, while the conditional information learned from $\mathbf{y}_{i,t}$ is used to generate the query. The transformer decoder stacks multiple transformer decoder layers, where the last two parallel feedforward layers and followed by two parallel normalization layers are used to generate the mean $\mu_{i,t}^{(2)}$ and variance $\Sigma_{i,t}^{(2)}$. These values are utilized to sample $\mathbf{F}_{i,t}^{(\mathbb{ED}_q)}$, resulting in the approximation of $q_\Pi\left(\mathbf{F}_{i,t}|\mathbf{X}_{i,t}, y_{i,t}\right)$. By following this process, we can generate the deep-level representations $\mathbf{F}_{i,t}^{(\mathbb{ED}_q)}$.

2.2.3. Approximate $p_\Phi\left(y_{i,t}|\mathbf{F}_{i,t}\right)$ with feed forward neural network

The purpose of the conditional probability distribution $p_\Phi\left(y_{i,t}|\mathbf{F}_{i,t}\right)$ is to estimate the RUL of bearing unit i at time t by utilizing the learned deep-level representations $\mathbf{F}_{i,t}^{(\mathbb{ED}_q)}$. In this work, we employ a feedforward neural network to approximate this distribution. This neural network is linked to the transformer encoder–decoder network $\mathbb{ED}_q$. Additionally, following the suggestions in [42], we introduce another feedforward neural network that connects to the transformer encoder network $\mathbb{E}_p$ to improve the prediction performance and robustness. This network uses the learned deep-level representations $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$ to make RUL predictions. We refer to these two feedforward neural networks as predictive networks because their goal is to use learned deep-level representations to predict the RUL of bearings. The predictive network connected to the transformer encoder network $\mathbb{E}_p$ is denoted as $\mathbb{P}_p$, and the predictive network connected to the transformer encoder–decoder network $\mathbb{ED}_q$ is denoted as $\mathbb{P}_q$. The feedforward neural network with a single layer that is used for RUL predictions can be represented

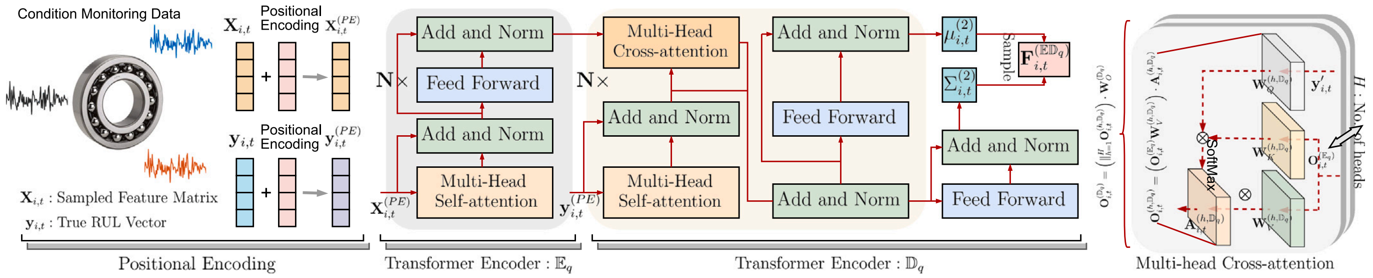

Condition Monitoring Data
$\mathbf{X}_{i,t}$
Positional Encoding $\mathbf{X}_{i,t}^{(PE)}$
$\mathbf{y}_{i,t}$
Positional Encoding $\mathbf{y}_{i,t}^{(PE)}$
$\mathbf{X}_{i,t}$: Sampled Feature Matrix
$\mathbf{y}_{i,t}$: True RUL Vector
Positional Encoding
N×
Add and Norm
Feed Forward
Add and Norm
Multi-Head Self-attention
$\mathbf{X}_{i,t}^{(PE)}$
Transformer Encoder : $\mathbb{E}_q$
N×
Multi-Head Cross-attention
Add and Norm
Multi-Head Self-attention
$\mathbf{y}_{i,t}^{(PE)}$
Add and Norm
Feed Forward
Add and Norm
$\mu_{i,t}^{(2)}$
$\Sigma_{i,t}^{(2)}$
Sample
$\mathbf{F}_{i,t}^{(\mathbb{ED}_q)}$
Add and Norm
Feed Forward
Transformer Encoder : $\mathbb{D}_q$
$\mathbf{O}_{i,t}^{(\mathbb{D}_q)} = \left(\Vert_{h=1}^{H} \mathbf{O}_{i,t}^{(h,\mathbb{D}_q)}\right) \cdot \mathbf{W}_O^{(\mathbb{D}_q)}$
$\mathbf{O}_{i,t}^{(h,\mathbb{D}_q)} = \left(\mathbf{O}_{i,t}^{(\mathbb{E}_q)} \mathbf{W}_V^{(h,\mathbb{D}_q)}\right) \cdot \mathbf{A}_{i,t}^{(h,\mathbb{D}_q)}$
$\mathbf{W}_Q^{(h,\mathbb{D}_q)}$
$\mathbf{y}'_{i,t}$
SoftMax
$\mathbf{W}_K^{(h,\mathbb{D}_q)}$
$\mathbf{O}_{i,t}^{(\mathbb{E}_q)}$
$\mathbf{A}_{i,t}^{(h,\mathbb{D}_q)}$
$\mathbf{W}_V^{(h,\mathbb{D}_q)}$
H : No. of heads
Multi-head Cross-attention

Fig. 2. The architecture of the transformer encoder–decoder network with multiple encoder layers and decoder layers that is utilized to generate deep-level representations $\mathbf{F}_{i,t}^{(\mathbb{ED}_q)}$.

mathematically as Eq. (25),

$$\hat{y}_{i,t}^{(\mathbb{P}_p)} = \sigma\left(\mathbf{W}^{(\mathbb{P}_p)} \cdot \text{Flatten}\left(\mathbf{F}_{i,t}^{(\mathbb{E}_p)}\right) + \mathbf{b}^{(\mathbb{P}_p)}\right)$$
$$\hat{y}_{i,t}^{(\mathbb{P}_q)} = \sigma\left(\mathbf{W}^{(\mathbb{P}_q)} \cdot \text{Flatten}\left(\mathbf{F}_{i,t}^{(\mathbb{ED}_q)}\right) + \mathbf{b}^{(\mathbb{P}_q)}\right) \tag{25}$$

where the symbol σ denotes the activation function, and Flatten refers to the flatten function. $\mathbf{W}^{(\mathbb{P}_p)}$ and $\mathbf{W}^{(\mathbb{P}_q)}$ are the kernel parameter matrices in the predictive networks $\mathbb{P}_p$ and $\mathbb{P}_q$, respectively. $\mathbf{b}^{(\mathbb{P}_p)}$ and $\mathbf{b}^{(\mathbb{P}_q)}$ are the bias vectors in the predictive networks $\mathbb{P}_p$ and $\mathbb{P}_q$, respectively. The predicted RUL provided by the predictive network $\mathbb{P}_p$ is denoted by $\hat{y}_{i,t}^{(\mathbb{P}_p)}$, and the predicted RUL provided by the predictive network $\mathbb{P}_q$ is denoted by $\hat{y}_{i,t}^{(\mathbb{P}q)}$. Using these notations, we can rewrite the expectation term in the ELBO listed in Eq. (9) as Eq. (26), where $y_{i,t}$ denotes the true RUL for unit i at time t.

$$\mathcal{L}_{\text{predict}}^{\mathbb{P}_q} = \sum_i \sum_t \left(\hat{y}_{i,t}^{(\mathbb{P}_q)} - y_{i,t}\right)^2 \tag{26}$$

The other introduced predictive network, $\mathbb{P}_p$, also results in a predictive loss that can be written as Eq. (27).

$$\mathcal{L}_{\text{predict}}^{\mathbb{P}_p} = \sum_i \sum_t \left(\hat{y}_{i,t}^{(\mathbb{P}_p)} - y_{i,t}\right)^2 \tag{27}$$

Additionally, by using the reparameterization trick [43,44], we can rewrite the KL-divergence in the ELBO listed in Eq. (9) as Eq. (28), where $\mathscr{d}$ refers to the dimensionality of the learned deep-level representations, and $tr(\cdot)$ denotes the trace of a matrix.

$$\mathcal{L}_{\text{KL}} = \sum_i \sum_t \frac{1}{2}\left(tr\left[\left(\Sigma_{i,t}^{(2)}\right)^{-1} \Sigma_{i,t}^{(1)}\right]\right.$$
$$\left. + \left(\mu_{i,t}^{(2)} - \mu_{i,t}^{(1)}\right)^T \left(\Sigma_{i,t}^{(2)}\right)^{-1} \left(\mu_{i,t}^{(2)} - \mu_{i,t}^{(1)}\right) - \mathscr{d} - \log\left(|\Sigma_{i,t}^{(2)}|/|\Sigma_{i,t}^{(1)}|\right)\right) \tag{28}$$

The overall training loss is the sum of all three losses from Eq. (26) to Eq. (28), and it can be written as Eq. (29).

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{predict}}^{\mathbb{P}_q} + \mathcal{L}_{\text{predict}}^{\mathbb{P}_p} + \mathcal{L}_{\text{KL}} \tag{29}$$

The gradient descent method is employed to update the parameters in the proposed conditional variational transformer network.

2.3. Two-stage training process of conditional variation transformer

Fig. 3 illustrates the two-stage training process and prediction process of the proposed conditional variational transformer for predicting the RUL. In the first training stage, the entire conditional variational transformer is trained using the overall training loss $\mathcal{L}_{\text{overall}}$. The KL-divergence loss is backpropagated within both the transformer encoder ($\mathbb{E}_p$) and the transformer encoder–decoder network ($\mathbb{E}_q$ and $\mathbb{D}_q$). Similarly, the prediction loss $\mathcal{L}_{\text{predict}}^{\mathbb{P}_p}$ is backpropagated within both the predictive network ($\mathbb{P}_p$) and the transformer encoder $\mathbb{E}_p$. The prediction loss $\mathcal{L}_{\text{predict}}^{\mathbb{P}_q}$ is also backpropagated within both the predictive network $\mathbb{P}_q$ and the transformer encoder–decoder network ($\mathbb{E}_q$ and $\mathbb{D}_q$). In the second training stage, the trained transformer encoder $\mathbb{E}_p$ obtained from the first training stage is connected to the predictive network $\mathbb{P}_p$, and the parameters in $\mathbb{E}_p$ are frozen. Only the predictive network $\mathbb{P}_p$ is retrained by backpropagating the prediction loss $\mathcal{L}_{\text{predict}}^{\mathbb{P}_p}$. To predict the RUL, the trained $\mathbb{E}_p$ and the retrained $\mathbb{P}_p$ are used. Specifically, the encoder $\mathbb{E}_p$ is used to extract the features of the input sequence, and the predictive network $\mathbb{P}_p$ is adopted to make the RUL prediction based upon the extracted features.

Table 1 provides further details about the first and second training stages used to train the proposed conditional variational transformer. Specifically, the first training stage involves both the feedforward and backpropagation processes. In the feedforward process, $\mathbf{X}_{i,t}$ with the positional encoding $PE_{i,t}$ for all i and t are input to the generative network $\mathbb{E}_p$ to obtain $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$, while both $\mathbf{X}_{i,t}$ and $\mathbf{y}_{i,t}$ with the positional encoding $PE_{i,t}$ for all i and t are input to the generative network $\mathbb{E}_q$ and $\mathbb{D}_q$ to obtain $\mathbf{F}_{i,t}^{(\mathbb{ED}_q)}$. The obtained $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$ and $\mathbf{F}_{i,t}^{(\mathbb{ED}_q)}$ are then input to the predictive networks $\mathbb{P}_p$ and $\mathbb{P}_q$ to obtain the RUL predictions $\hat{y}_{i,t}^{(\mathbb{P}_p)}$ and $\hat{y}_{i,t}^{(\mathbb{P}_q)}$, respectively. In the backpropagation process, the overall training loss $\mathcal{L}_{\text{overall}}$ is used to update the parameters within the entire conditional variational transformer network, including $\mathbb{E}_p$, $\mathbb{E}_q$, $\mathbb{D}_q$, $\mathbb{P}_p$, and $\mathbb{P}_q$. Similarly, the second training stage involves both the feedforward and backpropagation processes. In the feedforward process, $\mathbf{X}_{i,t}$ with the positional encoding $PE_{i,t}$ for all i and t are input to the generative network $\mathbb{E}_p$ to obtain $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$. The obtained $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$ is then input to the predictive network $\mathbb{P}_p$ to obtain the RUL predictions $\hat{y}_{i,t}^{(\mathbb{P}_p)}$. The training loss $\mathcal{L}_{\text{predict}}^{\mathbb{P}_p}$ is employed to update parameters within the predictive network $\mathbb{P}_p$ only.

3. Case study I

3.1. Data description

In this case study, we demonstrated the effectiveness of the conditional variational transformer using the FEMTO bearing dataset [45]. The dataset was collected from the PRONOSTIA platform, which is designed to accelerate the wear and tear of rolling bearings, enabling the detection of faults within hours. The platform consists of a gearbox attached to a rotating motor, a pneumatic jack, and a regulator that controls pressure using digital electro-pneumatic technology, which are used to manage the speed and load-up pressure of the bearings. The run-to-failure experiments were conducted with the platform, and were discontinued if the measured vibration exceeded 20 g-forces. Fig. 4 displays the PRONOSTIA platform, the normal bearings before the experiment, and the degraded bearings after the experiment. Table 2 shows the operating conditions used to collect the condition monitoring data in this dataset, as well as the bearing indices associated with the distinct operating conditions. In this case study, a seven-fold cross-validation was adopted to evaluate the prediction performance of the proposed conditional variational transformer on Bearing1_1 to Bearing1_7, demonstrating its ability to predict the RUL of bearings under a constant condition. A five-fold cross-validation was also used to evaluate the prediction performance of the proposed method on

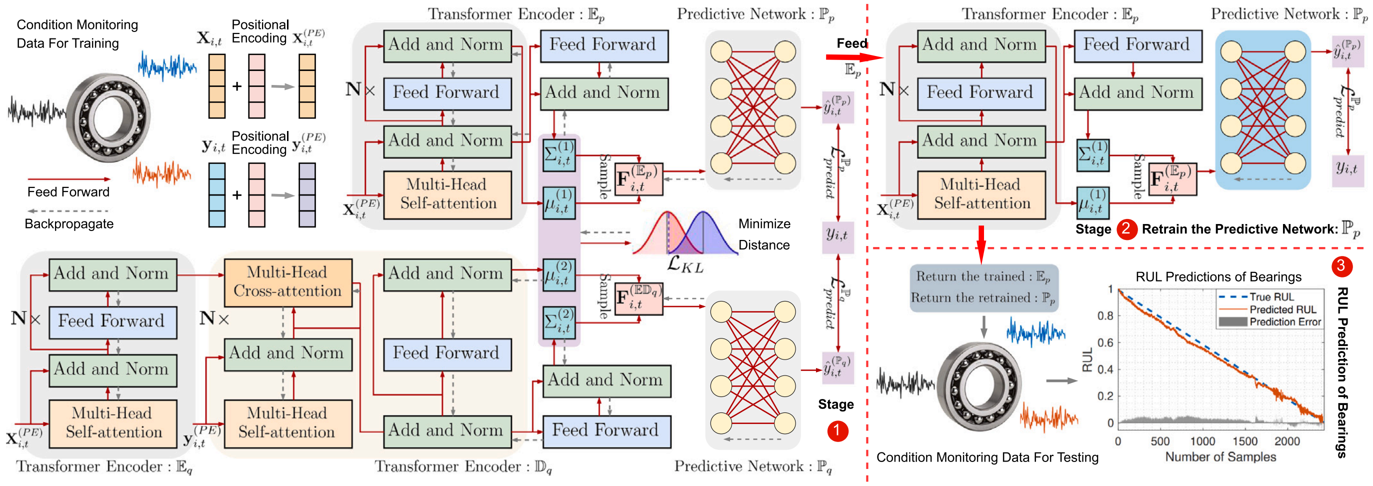

Condition Monitoring Data For Training
Positional Encoding
$\mathbf{X}_{i,t}$
$\mathbf{X}_{i,t}^{(PE)}$
$\mathbf{y}_{i,t}$
$\mathbf{y}_{i,t}^{(PE)}$
Feed Forward
Backpropagate
Transformer Encoder : $\mathbb{E}_p$
Add and Norm
Feed Forward
Multi-Head Self-attention
N×
$\Sigma_{i,t}^{(1)}$
$\mu_{i,t}^{(1)}$
Sample
$\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$
Predictive Network : $\mathbb{P}_p$
$\hat{y}_{i,t}^{(\mathbb{P}_p)}$
$\mathcal{L}_{predict}^{\mathbb{P}_p}$
$y_{i,t}$
Minimize Distance
$\mathcal{L}_{KL}$
$\mu_{i,t}^{(2)}$
$\Sigma_{i,t}^{(2)}$
$\mathbf{F}_{i,t}^{(\mathbb{ED}_q)}$
$\mathcal{L}_{predict}^{\mathbb{P}_q}$
$\hat{y}_{i,t}^{(\mathbb{P}_q)}$
Stage 1
Predictive Network : $\mathbb{P}_q$
Transformer Encoder : $\mathbb{E}_q$
Transformer Encoder : $\mathbb{D}_q$
Multi-Head Cross-attention
Feed $\mathbb{E}_p$
Stage 2 Retrain the Predictive Network: $\mathbb{P}_p$
Return the trained : $\mathbb{E}_p$
Return the retrained : $\mathbb{P}_p$
Condition Monitoring Data For Testing
RUL Predictions of Bearings
True RUL
Predicted RUL
Prediction Error
RUL
Number of Samples
0
500
1000
1500
2000
0.2
0.4
0.6
0.8
1
3 RUL Prediction of Bearings

Fig. 3. The two-stage training process of the proposed conditional variational transformer, and the test process of RUL predictions of bearings.

Table 1
The first and second training stages that are used to train the proposed conditional variational transformer.

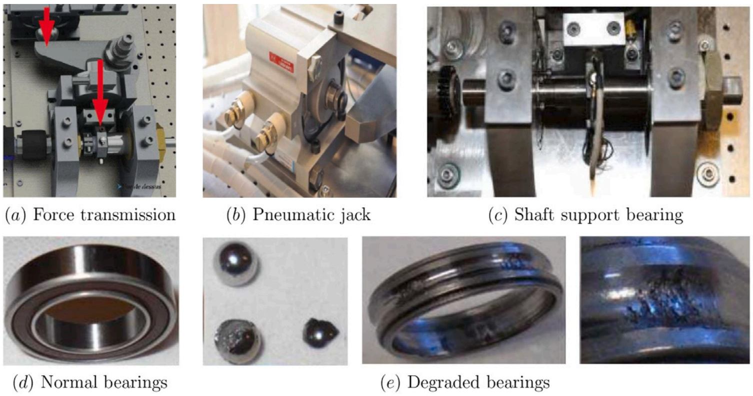
```
◇ First Training Stage
1. Extract features in both time-domain and frequency-domain, and the number of features is denoted as $\mathscr{F}$
2. Sample features and true RUL with a moving window to obtain feature matrix $\mathbf{X}_{i,t}$ and true RUL vector $\mathbf{y}_{i,t}$
3. Utilize a positional encoding function to obtain $PE_{i,t}$, and obtain $\mathbf{X}_{i,t}^{(PE)}$ and $\mathbf{y}_{i,t}^{(PE)}$
4. For iteration=1, ..., $I$ do
    4.1. For encoder layer = 1, ..., $N$, use $\mathbf{X}_{i,t}^{(PE)}$ and do within the generative network $\mathbb{E}_p$
        Generate attention matrix $\mathbf{A}_{i,t}^{(h,\mathbb{E}_p)}$, obtain $\mathbf{O}_{i,t}^{(h,\mathbb{E}_p)}$ in each head, then obtain the resulting tensor $\mathbf{O}_{i,t}^{(\mathbb{E}_p)}$
        Perform the residual connection and normalization to update $\mathbf{O}_{i,t}^{(\mathbb{E}_p)}$
        If encoder layer is $N$, use two feedforward and normalization layers to generate $\mu_{i,t}^{(1)}$ and $\Sigma_{i,t}^{(1)}$, and sample $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$
        If encoder layer is less than $N$, use one feedforward and normalization layer to obtain the resulting tensor
    4.2. End return $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$
    4.3. For encoder layer = 1, ..., $N$, use $\mathbf{X}_{i,t}^{(PE)}$ and do within the generative network $\mathbb{E}_q$
        Use the multi-head self-attention mechanism and follows by a normalization layer
        Use a feed forward layer and follows by a normalization layer to obtain $\mathbf{O}_{i,t}^{(\mathbb{E}_q)}$
    4.4. End return $\mathbf{O}_{i,t}^{(\mathbb{E}_q)}$
    4.5. For decoder layer=1, ..., $N$, use $\mathbf{y}_{i,t}^{(PE)}$ and $\mathbf{O}_{i,t}^{(\mathbb{E}_q)}$ and do within the generative network $\mathbb{D}_q$
        Use the multi-head self-attention mechanism to obtain $\mathbf{y}'_{i,t}$, follows by a normalization layer
        Use $\mathbf{O}_{i,t}^{(\mathbb{E}_q)}$ to generate pairs of key and value, and use $\mathbf{y}'_{i,t}$ generate query
        Use the multi-head cross-attention mechanism to obtain $\mathbf{O}_{i,t}^{(\mathbb{D}_q)}$, follows by a normalization layer
        If encoder layer is $N$, use two feedforward and normalization layers to generate $\mu_{i,t}^{(2)}$ and $\Sigma_{i,t}^{(2)}$, and sample $\mathbf{F}_{i,t}^{(\mathbb{ED}_q)}$
        If encoder layer is less than $N$, use one feedforward and normalization layer to obtain the resulting tensor
    4.6. End return $\mathbf{F}_{i,t}^{(\mathbb{ED}_q)}$
    4.7. Feed $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$ and $\mathbf{F}_{i,t}^{(\mathbb{ED}_q)}$ into the predictive network $\mathbb{P}_p$ and $\mathbb{P}_q$ to obtain $\hat{y}_{i,t}^{(\mathbb{P}_p)}$ and $\hat{y}_{i,t}^{(\mathbb{P}_q)}$, respectively
    4.8. Obtained the training loss $\mathcal{L}_{\text{overall}} \leftarrow \mathcal{L}_{\text{predict}}^{\mathbb{P}_q} + \mathcal{L}_{\text{predict}}^{\mathbb{P}_p} + \mathcal{L}_{\text{KL}}$
    4.9. Use the overall training loss $\mathcal{L}_{\text{overall}}$ to backpropagate update the parameters within $\mathbb{E}_p, \mathbb{E}_q, \mathbb{D}_q, \mathbb{P}_p$, and $\mathbb{P}_q$
5. End and return the trained generative network $\mathbb{E}_p$

◇ Second Training Stage
1. For iteration=1, ..., $I$ do
    1.1. Use the trained $\mathbb{E}_p$ from the first training stage to obtained the learned deep-level representations $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$
    1.2. Feed $\mathbf{F}_{i,t}^{(\mathbb{E}_p)}$ into the predictive network $\mathbb{E}_p$ to obtain the predicted RUL $\hat{y}_{i,t}^{(\mathbb{P}_p)}$
    1.3. Use the prediction loss $\mathcal{L}_{\text{predict}}^{\mathbb{P}_p}$ to backpropagate update the parameters within $\mathbb{P}_p$
2. End and return the trained $\mathbb{E}_p$ from the first training stage and the trained $\mathbb{P}_p$ from the second training stage
```

Bearing2_1 to Bearing3_3, showing its ability to predict the RUL of bearings under varying operating conditions. Furthermore, 20 features and their cumulative features were extracted from the signals collected in both the horizontal and vertical directions, resulting in a total of 80 extracted features. Additional details about these features can be found in [46–48].

3.2. Piece-wise RUL prediction and hyperparameters

Previous studies have shown that bearing degradation processes typically experience multiple degradation stages [9,49], and estimating the RUL of bearings based on these different stages can enhance prediction performance [50]. Therefore, in this case study, we used an abrupt change point detection method [51] to detect different degradation stages, so that the RUL of bearings can be estimated in a piece-wise order. More specifically, the root-mean-square (RMS) was used for change point detection. With respect to each detected change point, the average RMS of condition monitoring data before and after the detected change point was compared. The detected change point was considered a true change point if the average of condition monitoring data after the detected change point was more than twice the average of condition monitoring data before the detected change point, so


(*a*) Force transmission (*b*) Pneumatic jack (*c*) Shaft support bearing

(*d*) Normal bearings (*e*) Degraded bearings

Fig. 4. The PRONOSTIA platform, the normal bearings before the experiment, and the degraded bearings after the experiments [45].

Table 2
The operating conditions used to collect the condition monitoring data in the FEMTO bearing dataset, as well as the bearing indices associated with the different operating conditions.

Condition	Angular velocity (rpm)	Radial force (kN)	Twisting force (N m)	Bearing indices
Condition 1	1800	4.0	1.326	Bearing1_1 to Bearing1_7
Condition 2	1650	4.2	1.447	Bearing2_1 to Bearing2_7
Condition 3	1500	5.0	1.591	Bearing3_1 to Bearing3_3

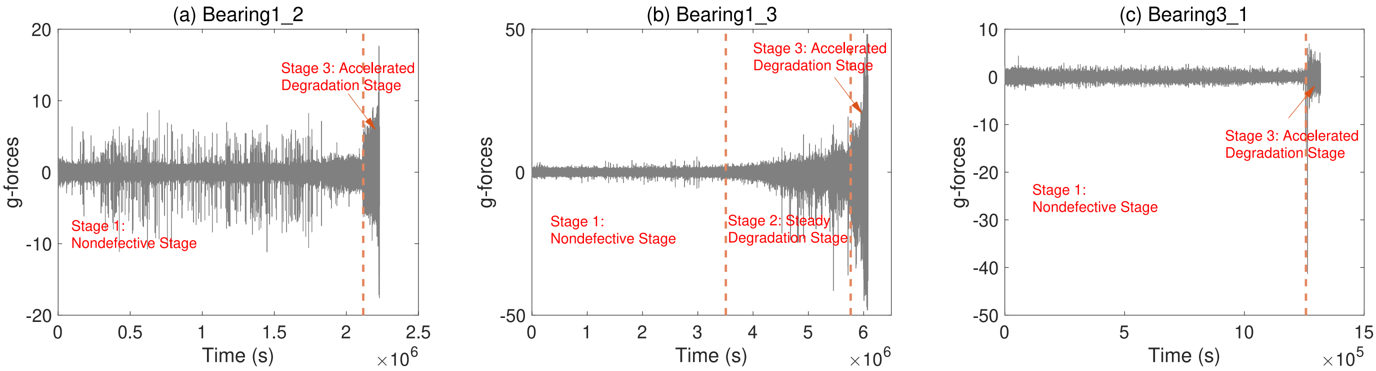


Fig. 5. The vibration signals for IEEE Bearing1_2, IEEE Bearing1_3, and IEEE Bearing3_1, and the results of the stage detection.

that the degradation stages can be effectively detected. More details about the change point detection method can be found in [9,52]. Fig. 5 displays the vibration signals for various bearing units and the results of the degradation stage detection. The number of detected stages differs among the bearing units. For instance, Bearing1_3 has three degradation stages, including a non-defective stage, a steady degradation stage, and an accelerated degradation stage. In contrast, Bearing1_2 and Bearing3_1 only have two degradation stages, involving a non-defective stage and an accelerated degradation stage. As some bearings do not involve a steady degradation stage, we trained one predictive model to estimate the RUL for both the non-defective and steady degradation stages, and another predictive model to estimate the RUL for the accelerated degradation stage.

The hyperparameters of the presented conditional variational transformer are set as follows: The batch size is 100, and $d_{\text{model}} = \mathscr{F}$ is set to 80. The quantity of encoder layers in the generative network $\mathbb{E}_p$ is set to 3, the quantity of encoder layers in $\mathbb{E}_q$ is 3, and the quantity of decoder layers in the generative network $\mathbb{D}_q$ is also set to 3. The quantity of hidden nodes in the feedforward layers is set to 200 in both generative networks. The number of heads H is determined as one, and the number of feedforward layers in the predictive networks $\mathbb{P}_p$ and $\mathbb{P}_q$ are set to 2. The activation function ReLU is utilized in all hidden layers, and the linear activation function is used for all last layers. The learning rate in the first training stage is set to 10^{-3}, and the learning rate in the second training stage decreases to 10^{-4} for fine-tuning the predictive network $\mathbb{P}_p$. Moreover, both the window size $\mathscr{S}$ and the prediction starting points are set to 20 samples. Each sample includes 2560 data points in this dataset, which means that the RUL prediction initiates when 20×2560 data points have been observed. There are two primary reasons why the prediction starting point is set at 20 samples. First, 20 samples ensure that there is sufficient condition monitoring data to accurately initiate the RUL prediction process. Second, using 20 samples will not significantly increase the size of the training data, resulting in an acceptable training time.

3.3. Prediction results

Fig. 6 shows the RUL prediction results for some of the bearing units, and this figure includes the true RUL, predicted RUL, and the prediction error. The prediction error refers to the true RUL subtracts the predicted RUL. In this work, the RUL refers to the percentage of remaining lifetime of bearings. More specifically, the RUL of a specific bearing unit i at time t is defined as $(T_i - t)/T_i$, where T_i denotes the total lifetime of the bearing unit i. From this figure, a preliminary conclusion can be drawn that the conditional variational transformer can predict the RUL of bearings with relatively high precision, as the predicted RUL trajectory is close to the true RUL trajectory of bearings. For instance, for Bearing1_1, the predicted RUL is 0.959 while the true RUL is 0.993. Similarly, for Bearing2_1, the predicted RUL is 0.273 while the true RUL is 0.288.

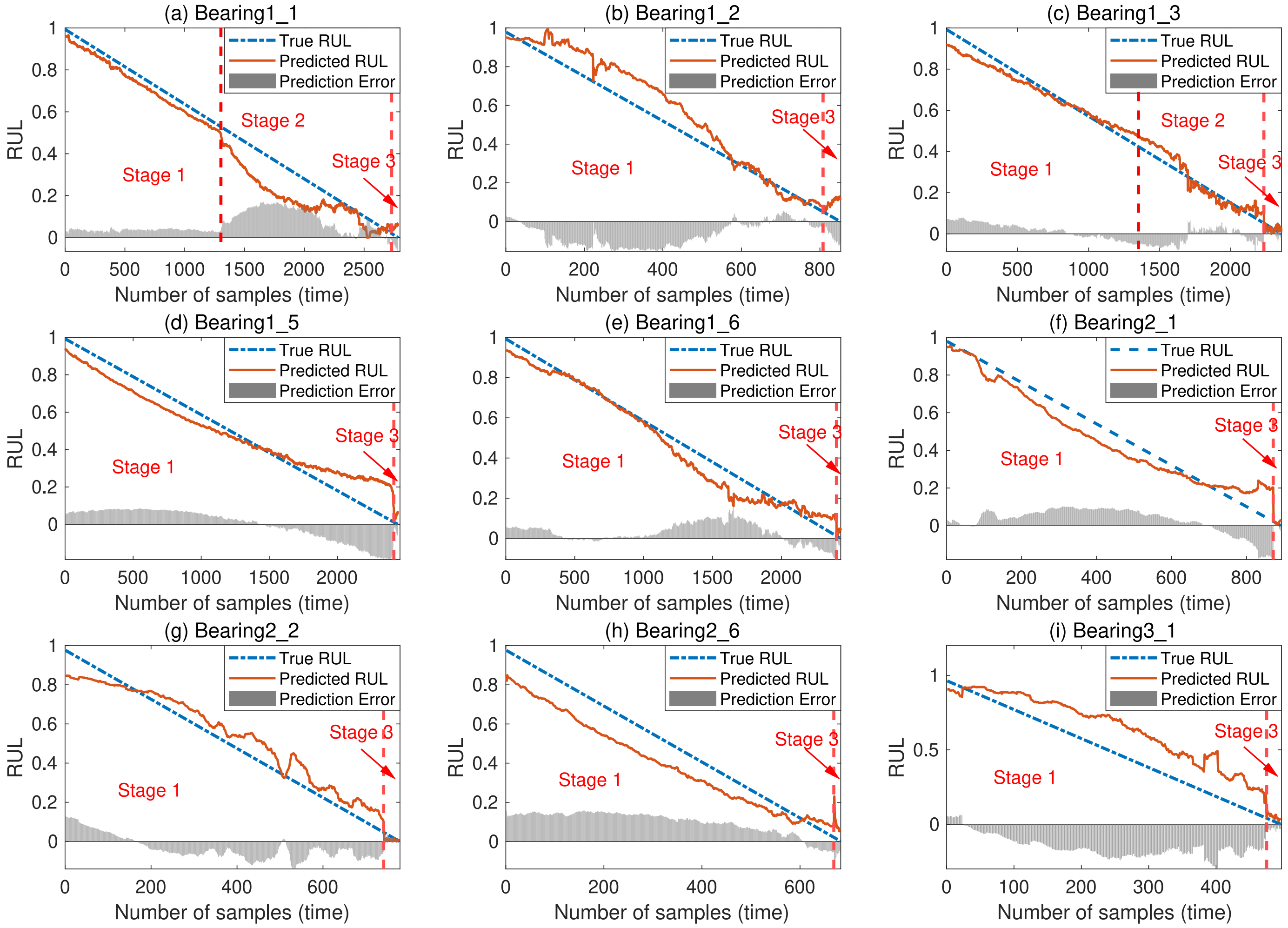


Fig. 6. The RUL prediction results for a selection of bearing units from the FEMTO bearing dataset.

3.4. Ablation study

In order to provide additional evidence supporting the efficiency of the proposed conditional variational transformer, we carried out an ablation study. Firstly, we removed the fine-tuning mechanism in the second training stage to show the impact of the proposed two-stage training process. Then, we removed the transformer encoder $\mathbb{E}_q$ and transformer decoder $\mathbb{D}_q$ and used only the transformer encoder $\mathbb{E}_p$ to make RUL predictions to show the effectiveness of the conditional variational inference. Table 3 displays the prediction results of the ablation study in terms of root-mean-squared error (RMSE) and mean absolute error (MAE) for all bearing units in this case study. In this table, CVT-FT denotes the proposed method with the fine-tuning mechanism used in the two-stage training process, CVT denotes the proposed conditional variational transformer without the fine-tuning mechanism in the two-stage training process, and TENC refers to the transformer encoder. Based on this table, we can demonstrate that the proposed CVT-FT can predict the RUL of bearings with high accuracy and can improve the prediction performance. As an example, for Bearing3_1, the prediction RMSE of the proposed CVT-FT is 0.080, while the prediction RMSE of the ablation study for CVT and TENC are 0.126 and 0.125, respectively. In terms of the average prediction error, the average prediction RMSE of the proposed method is 0.134 and the average prediction MAE of the proposed method is 0.113. In contrast, the average prediction RMSE of TENC is 0.161 and the average prediction MAE of TENC is 0.137.

Fig. 7 displays the spider plot of five evaluation metrics used to assess the prediction performance of the methods used in this ablation study, namely RMSE (root-mean-squared-errors), MAE (mean-absolute error), MSE (mean-squared-errors), MedAE (median absolute error), and R2 Score (r-squared score). From Fig. 7, we conclude that both the proposed conditional variational transformer and the fine-tuning mechanism in the proposed two-stage training process can improve the prediction performance across all evaluation metrics. For instance, for all bearing units operated under the third operating condition, the proposed CVT-FT achieves a prediction MedAE of 0.095. In contrast, the prediction MedAE of CVT and TENC are 0.100 and 0.168, respectively. Moreover, for all bearing units operated under the first operating condition, the R2 Score of the proposed CVT-FT is 0.913, while the R2 Score of CVT and TENC are 0.866 and 0.840, respectively.

3.5. Comparative study

To conduct a comprehensive evaluation of the proposed CVT-FT, we also conducted a comparative study with various deep learning approaches documented in the existing literature. Table 4 presents the average prediction RMSE for bearings operated under three distinct operating conditions using CVT-FT (proposed), CVT, TENC, bi-channel hierarchical vision transformer (BCVHiT), convolutional long short term memory (CLSTM), convolutional neural network (CNN), deep adversarial network (DAN), generative adversarial network (GAN), and transferable bidirectional GRU (TGRU). From this table, we can observe that the proposed CVT-FT outperforms many of the deep learning methods reported in the literature, regardless of the operating conditions. For instance, with respect to all bearing units operated under the first

Table 3
The RMSE and MAE of RUL predictions for all bearings in the FEMTO bearing dataset.

Condition	Bearing index	RMSE			MAE		
		CVT-FT	CVT	TENC	CVT-FT	CVT	TENC
Condition 1	Bearing1_1	0.080	0.126	0.125	0.064	0.101	0.092
	Bearing1_2	0.091	0.105	0.171	0.075	0.094	0.140
	Bearing1_3	0.040	0.045	0.049	0.034	0.035	0.034
	Bearing1_4	0.124	0.131	0.141	0.111	0.118	0.127
	Bearing1_5	0.082	0.089	0.075	0.070	0.081	0.069
	Bearing1_6	0.056	0.082	0.054	0.044	0.060	0.046
	Bearing1_7	0.088	0.124	0.123	0.069	0.098	0.095
Condition 2	Bearing2_1	0.072	0.098	0.102	0.061	0.087	0.082
	Bearing2_2	0.069	0.117	0.095	0.061	0.100	0.079
	Bearing2_3	0.239	0.264	0.241	0.206	0.217	0.190
	Bearing2_4	0.239	0.239	0.225	0.199	0.209	0.180
	Bearing2_5	0.298	0.322	0.336	0.278	0.292	0.299
	Bearing2_6	0.113	0.172	0.176	0.105	0.165	0.165
	Bearing2_7	0.242	0.264	0.277	0.195	0.231	0.245
Condition 3	Bearing3_1	0.164	0.194	0.253	0.150	0.181	0.236
	Bearing3_2	0.101	0.084	0.078	0.075	0.069	0.059
	Bearing3_3	0.177	0.170	0.216	0.124	0.114	0.195
	Average	**0.134**	0.155	0.161	**0.113**	0.132	0.137

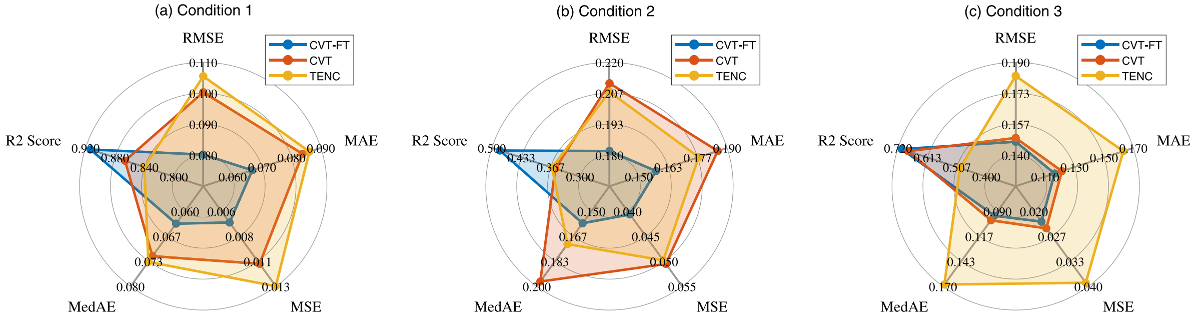


Fig. 7. The spider plot of five evaluation metrics are used to evaluate the prediction performance of the methods used in this ablation study.

Table 4
The average prediction RMSE of the proposed CVT-FT and other deep learning methods reported in the literature.

Condition	CVT-FT	CVT	TENC	BCVHiT [53]	CLSTM [54]	CNN [55]	DAN [56]	GAN [23]	TGRU [57]
Condition 1	**0.080**	0.100	0.106	0.133	0.159	0.189	0.206	0.105	0.230
Condition 2	**0.182**	0.211	0.207	0.175	0.173	0.260	0.206	0.187	0.170
Condition 3	**0.147**	0.149	0.183	0.189	0.152	0.290	0.366	–	0.150

condition, the average prediction RMSE of the proposed CVT-FT is 0.080, while the average prediction RMSE of other methods ranges from 0.100 to 0.230. Similarly, for all bearing units operated under the third operating condition, the average prediction RMSE of the proposed CVT-FT is 0.147, while the average prediction RMSE of the other methods ranges from 0.149 to 0.366.

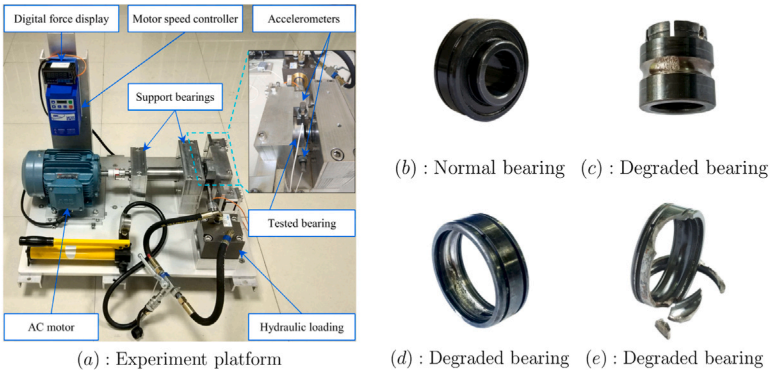
4. Case study II

4.1. Data description

The effectiveness of the conditional variational transformer was evaluated on the XJTU-SY bearing dataset [58] in this case study. This dataset consists of condition monitoring data from 15 LDK UER204 bearings that were operated under three different conditions. Run-to-failure experiments were performed, and if the measured vibration exceeded 20 g-forces, the experiments were discontinued. This criterion is similar to that used in the FEMTO bearing dataset. Additional details regarding the experimental setup can be found in [58]. Fig. 8 displays the platform utilized to gather the condition monitoring data, as well as the normal bearings before the experiment and the degraded bearings after the experiment. Table 5 outlines the combinations of different operating conditions, and in this case study, bearings operated under conditions 1 and 2 were used. Furthermore, a five-fold cross-validation approach was used to assess the prediction performance of the proposed conditional variational transformer for bearings that were operated under different conditions.

4.2. Piece-wise RUL prediction and hyperparameters

Similarly, in this case study, we utilized the change point detection method to detect various degradation stages of bearings so that RUL prediction could be performed on a piece-wise basis. Fig. 9 demonstrates some of the results of the degradation stage detection for bearings in the XJTU-SY bearing dataset. It is evident that the number of detected stages varies among the bearing units. For instance, Bearing1_1 and Bearing2_5 have three degradation stages, including a non-defective stage, a steady degradation stage, and an accelerated degradation stage. Conversely, Bearing1_4 only has two degradation stages, consisting of a non-defective stage and an accelerated degradation stage. Since some bearings do not have a steady degradation stage, we trained one predictive model to predict the RUL for both the non-defective and steady degradation stages, and another predictive model to forecast the RUL for the accelerated degradation stage. Furthermore, most of the hyperparameters utilized in this case study were identical to those used in the first case study, except for the quantity of hidden



(a) : Experiment platform

(b) : Normal bearing

(c) : Degraded bearing

(d) : Degraded bearing

(e) : Degraded bearing

Fig. 8. (a) The experiment platform utilized to collect the condition monitoring data; (b) Normal bearings; (c) (d) (e) Degraded bearings [58].

Table 5
The details about the combination of different operating conditions.

Condition	Angular velocity (rpm)	Radial force (kN)	Bearing index
Condition 1	2100	12	Bearing1_1 to Bearing1_5
Condition 2	2250	11	Bearing2_1 to Bearing2_5
Condition 3	2400	10	Bearing3_1 to Bearing3_5

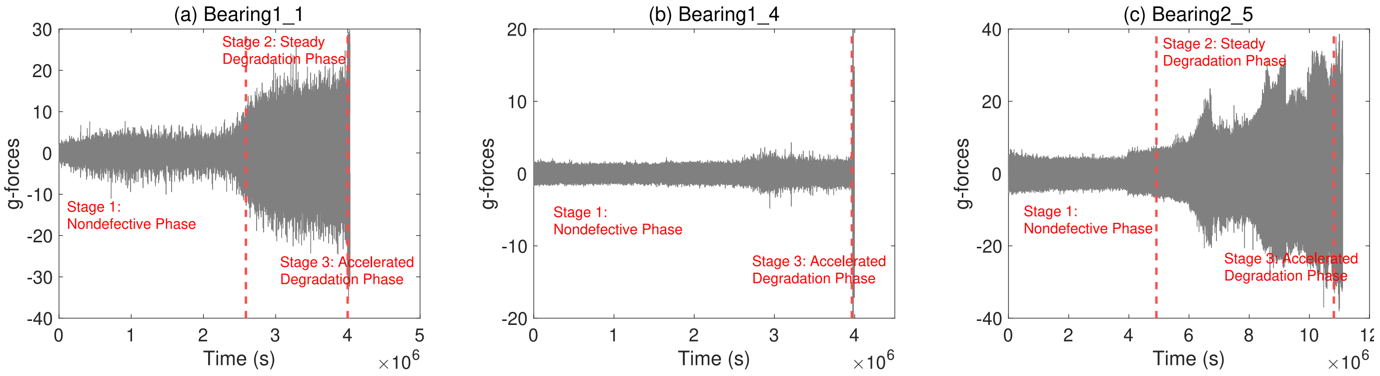


Fig. 9. The results of the degradation stage detection for bearings in the XJTU-SY bearing dataset.

nodes in the predictive networks $\mathbb{P}_p$ and the learning rate. The learning rate was set to 10^{-4} in the first training stage and 10^{-3} in the second training stage. Concerning bearings operated under the first operating condition, the quantity of hidden nodes in the first hidden layer of $\mathbb{P}_p$ was set to 4000. Regarding bearings operated under the second operating condition, the quantity of hidden nodes in the first hidden layer of $\mathbb{P}_p$ was set to 500.

4.3. Prediction results

Fig. 10 shows the RUL prediction results for some bearing units, and this figure includes the true RUL, predicted RUL, and the prediction error. The prediction error refers to the true RUL subtracts the predicted RUL. From this figure, it is clear that the proposed conditional variational transformer demonstrates a considerable level of precision in predicting the RUL, as the predicted degradation trajectory closely aligns with the true degradation trajectory. For example, concerning Bearing1_1, the predicted RUL of the bearing is 0.553 when the true RUL is 0.516. Regarding Bearing2_5, the predicted RUL of the bearing is 0.955 when the true RUL is 0.944.

4.4. Ablation study

To further demonstrate the efficiency of the proposed conditional variational transformer, an ablation study was conducted in this case study, identical to the one conducted in Case Study I. Table 6 presents the prediction results of the ablation study in terms of RMSE and MAE for all bearing units in the XJTU-SY bearing dataset. Based on this table, it is evident that the proposed CVT-FT can predict the RUL of bearings with high accuracy and is capable of enhancing the prediction performance. For example, concerning Bearing1_1, the prediction RMSE of the proposed CVT-FT is 0.115, while the prediction RMSE of the ablation study for CVT and TENC are 0.118 and 0.152, respectively. With regard to the average prediction error, the average prediction RMSE of the proposed method is 0.194, and the average prediction MAE of the proposed method is 0.165. In contrast, the average prediction RMSE of TENC is 0.301, and the average prediction MAE of TENC is 0.244.

Fig. 11 shows the box plot of prediction RMSE and MAE for all bearing units operated under both the first and second operating conditions. This figure verifies that the presented conditional variational transformer can improve the prediction accuracy. For instance, the average prediction RMSE of bearings operated under the first condition using the proposed CVT-FT is 0.172, while the average prediction RMSE of bearings operated under the first condition using TENC is 0.234. Moreover, the average prediction MAE of bearings operated under the second condition using the proposed CVT-FT is 0.151, whereas the average prediction RMSE of bearings operated under the second condition using CVT is 0.271.

Fig. 12 shows a spider plot of the five evaluation metrics used to assess the prediction performance of the methods used in this ablation study. From Fig. 12, we observe that both the presented conditional

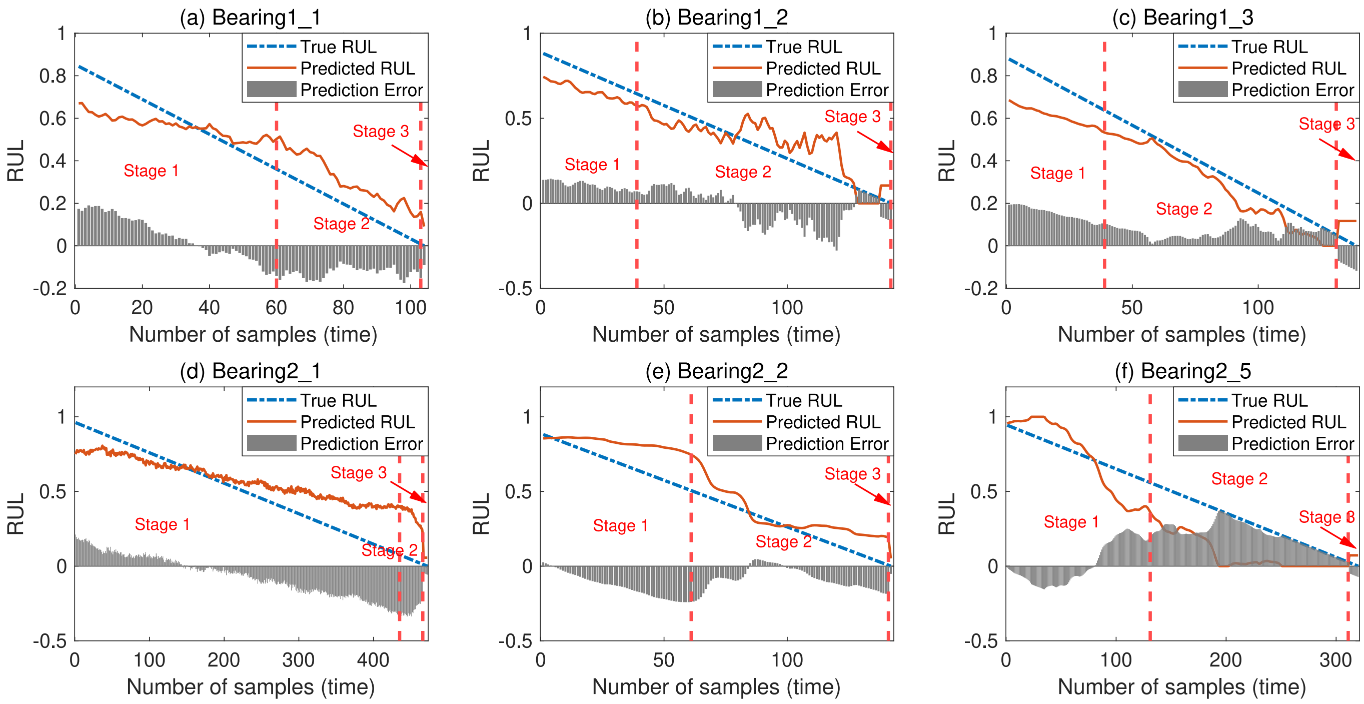


Fig. 10. The RUL prediction results for a selection of bearing units from the XJTU-SY bearing dataset.

Table 6
The RMSE and MAE of RUL predictions for all bearings in the XJTU-SY bearing dataset.

Condition	Bearing index	RMSE			MAE		
		CVT-FT	CVT	TENC	CVT-FT	CVT	TENC
Condition 1	Bearing1_1	0.115	0.118	0.152	0.103	0.061	0.114
	Bearing1_2	0.110	0.248	0.383	0.097	0.174	0.293
	Bearing1_3	0.102	0.204	0.097	0.090	0.156	0.075
	Bearing1_4	0.356	0.481	0.293	0.304	0.444	0.233
	Bearing1_5	0.176	0.622	0.247	0.156	0.521	0.219
Condition 2	Bearing2_1	0.160	0.193	0.310	0.132	0.145	0.257
	Bearing2_2	0.131	0.143	0.429	0.111	0.105	0.370
	Bearing2_3	0.241	0.233	0.413	0.186	0.199	0.333
	Bearing2_4	0.356	0.723	0.413	0.300	0.696	0.342
	Bearing2_5	0.193	0.225	0.270	0.171	0.174	0.203
	Average	**0.194**	0.319	0.301	**0.165**	0.267	0.244

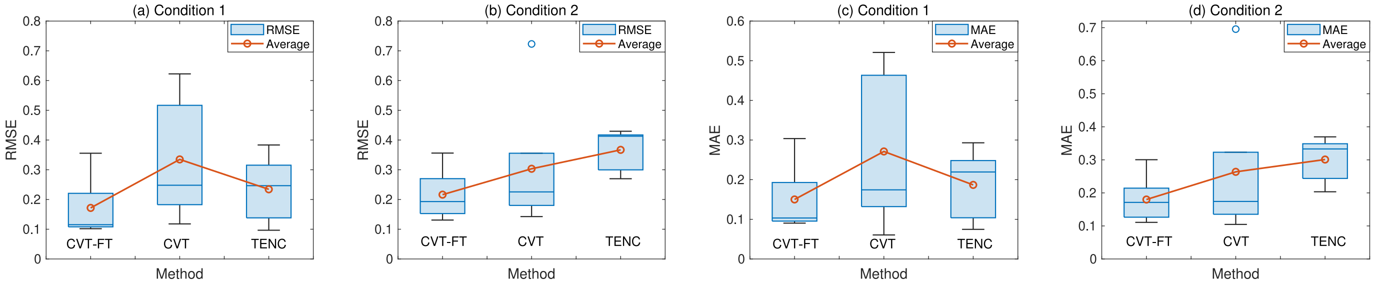


Fig. 11. The box plot of prediction RMSE and MAE for all bearing units operated under both the first and second operating conditions.

variational transformer and the fine-tuning mechanism in the two-stage training process can enhance the prediction performance across different evaluation metrics. For instance, for all bearing units operated under the second operating condition, the prediction MedAE of the proposed CVT-FT is 0.164, while the prediction MedAE of CVT and TENC are 0.256 and 0.254, respectively. Similarly, for all bearing units operated under the first operating condition, the MSE of the proposed CVT-FT is 0.039, compared to the MSE of 0.147 and 0.065 for CVT and TENC, respectively.

4.5. Comparative study

To further demonstrate the efficacy of the proposed CVT-FT, it was compared with other deep learning methods documented in the literature. Table 7 displays the average prediction RMSE for bearing units operated under different operating conditions using CVT-FT (proposed), CVT, TENC, multiscale CNN (MCNN), deep adversarial network (DAN), LSTM, and graph convolutional network with self-attention mechanism (GCN-SA). Based on this table, we can conclude that the

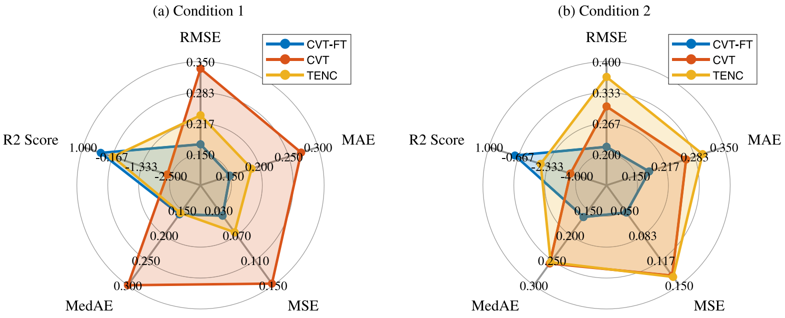


Fig. 12. The spider plot of five evaluation metrics are used to evaluate the prediction performance of the methods used in the ablation study.

Table 7
The average prediction RMSE of the proposed CVT-FT and other deep learning methods reported in the literature.

Condition	CVT-FT	CVT	TENC	MCNN	DAN [56]	LSTM [59]	GCN-SA [46]
Condition 1	**0.172**	0.335	0.234	0.248	0.297	0.264	0.175
Condition 2	**0.216**	0.303	0.367	0.230	0.240	0.346	0.218

proposed CVT-FT outperforms some of the deep learning methods reported in the literature for all operating conditions. For example, the average prediction RMSE of the proposed CVT-FT for all bearing units operated under the first condition is 0.172, while the average prediction RMSE of other methods ranges from 0.175 to 0.335.

5. Conclusion and future work

A novel conditional variational transformer architecture was developed to address the limitations of the self-attention mechanism in the conventional transformer model. The proposed architecture consists of four networks: two generative networks and two predictive networks. The first generative network used a transformer encoder–decoder architecture with a cross-attention mechanism to learn deep-level representations in one feature space of the condition monitoring data. The true RUL data were used in the cross-attention mechanism, allowing the attention matrix to select the most important features of the condition monitoring data that are highly correlated with the true RUL to make RUL predictions. The second generative network used a transformer encoder to learn deep-level representations in another feature space of the condition monitoring data with the condition monitoring data only as input. Two separate predictive networks used the learned deep-level representations in different feature spaces to predict the RUL of bearings. Because the true RUL data are not available during testing, a KL divergence was introduced to minimize the distance between two feature spaces so that the feature space extracted from the second generative network can approximate the feature space extracted from the first generative network. Therefore, without the true RUL data, the feature space extracted from the first generative network can be taken into account when the feature space extracted from the second generative network is used for testing. Additionally, we introduced a two-stage training process to train the proposed conditional variational transformer. In the first training stage, we trained both generative networks and both predictive networks by minimizing two prediction losses and the KL-divergence loss simultaneously. In the second training stage, we implemented a fine-tuning mechanism to specifically tune the parameters in the second predictive network by minimizing a single prediction loss only. The first training stage aims to minimize the distance between two feature spaces generated by two generative networks; the second training stage aims to minimize the prediction loss to achieve the optimal prediction performance. The proposed method was demonstrated on two publicly available bearing datasets, including the FEMTO bearing dataset and the XJTU-SY bearing dataset. The experimental results have shown that the proposed method achieved an average RMSE of 0.134 for the FEMTO bearing dataset and an average RMSE of 0.194 for the XJTU-SY bearing dataset. The proposed method outperforms existing data-driven methods reported in the literature. In future work, we will investigate the effectiveness of the proposed method on other datasets.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Mingyue Yu, Yi Zhang, Chunxue Yang, Rolling bearing faults identification based on multiscale singular value, Adv. Eng. Inform. 57 (2023) 102040.

[2] Yaguo Lei, Jing Lin, Zhengjia He, Ming J. Zuo, A review on empirical mode decomposition in fault diagnosis of rotating machinery, Mech. Syst. Signal Process. 35 (1–2) (2013) 108–126.

[3] Fernando Porté-Agel, Majid Bastankhah, Sina Shamsoddin, Wind-turbine and wind-farm flows: A review, Bound.-Layer Meteorol. 174 (2020) 1–59.

[4] Duy-Tang Hoang, Hee-Jun Kang, A survey on deep learning based bearing fault diagnosis, Neurocomputing 335 (2019) 327–335.

[5] Zepeng Liu, Long Zhang, A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings, Measurement 149 (2020) 107002.

[6] Dhiraj Neupane, Jongwon Seok, Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review, IEEE Access 8 (2020) 93155–93178.

[7] M.M. Manjurul Islam, Alexander E. Prosvirin, Jong-Myon Kim, Data-driven prognostic scheme for rolling-element bearings using a new health index and variants of least-square support vector machines, Mech. Syst. Signal Process. 160 (2021) 107853.

[8] Jianghong Zhou, Yi Qin, Dingliang Chen, Fuqiang Liu, Quan Qian, Remaining useful life prediction of bearings by a new reinforced memory GRU network, Adv. Eng. Inform. 53 (2022) 101682.

[9] Junchuan Shi, Tianyu Yu, Kai Goebel, Dazhong Wu, Remaining useful life prediction of bearings using ensemble learning: The impact of diversity in base learners and features, J. Comput. Inf. Sci. Eng. 21 (2) (2021).

[10] Teng Wang, Zheng Liu, Nezih Mrad, A probabilistic framework for remaining useful life prediction of bearings, IEEE Trans. Instrum. Meas. 70 (2020) 1–12.

[11] Yuxiong Li, Xianzhen Huang, Chengying Zhao, Pengfei Ding, A novel remaining useful life prediction method based on multi-support vector regression fusion and adaptive weight updating, ISA Trans. 131 (2022) 444–459.

[12] Zong Meng, Jing Li, Na Yin, Zuozhou Pan, Remaining useful life prediction of rolling bearing using fractal theory, Measurement 156 (2020) 107572.

[13] Gang Wang, Hui Li, Feng Zhang, Zhangjun Wu, Feature fusion based ensemble method for remaining useful life prediction of machinery, Appl. Soft Comput. 129 (2022) 109604.

[14] Chenyang Wang, Wanlu Jiang, Xukang Yang, Shuqing Zhang, RUL prediction of rolling bearings based on a DCAE and CNN, Appl. Sci. 11 (23) (2021) 11516.

[15] Gang Wang, Jiawei Xiang, Remain useful life prediction of rolling bearings based on exponential model optimized by gradient method, Measurement 176 (2021) 109161.

[16] Weiyang Xu, Quansheng Jiang, Yehu Shen, Fengyu Xu, Qixin Zhu, RUL prediction for rolling bearings based on convolutional autoencoder and status degradation model, Appl. Soft Comput. 130 (2022) 109686.

[17] Yupeng Wei, Dazhong Wu, State of health and remaining useful life prediction of lithium-ion batteries with conditional graph convolutional network, Expert Syst. Appl. (2023) 122041.

[18] Yupeng Wei, Dazhong Wu, Model-based real-time prediction of surface roughness in fused deposition modeling with graph convolutional network-based error correction, J. Manuf. Syst. 71 (2023) 286–297.

[19] Yupeng Wei, Dazhong Wu, Janis Terpenny, Learning the health index of complex systems using dynamic conditional variational autoencoders, Reliab. Eng. Syst. Saf. 216 (2021) 108004.

[20] Hao Lu, Vahid Barzegar, Venkat Pavan Nemani, Chao Hu, Simon Laflamme, Andrew Todd Zimmerman, Joint training of a predictor network and a generative adversarial network for time series forecasting: A case study of bearing prognostics, Expert Syst. Appl. 203 (2022) 117415.

[21] Jun Zhu, Nan Chen, Weiwen Peng, Estimation of bearing remaining useful life based on multiscale convolutional neural network, IEEE Trans. Ind. Electron. 66 (4) (2018) 3208–3216.

[22] Xiaoyu Yang, Ying Zheng, Yong Zhang, David Shan-Hill Wong, Weidong Yang, Bearing remaining useful life prediction based on regression shapalet and graph neural network, IEEE Trans. Instrum. Meas. 71 (2022) 1–12.

[23] Sungho Suh, Paul Lukowicz, Yong Oh Lee, Generalized multiscale feature extraction for remaining useful life prediction of bearings with generative adversarial networks, Knowl.-Based Syst. 237 (2022) 107866.

[24] Lu Liu, Xiao Song, Kai Chen, Baocun Hou, Xudong Chai, Huansheng Ning, An enhanced encoder–decoder framework for bearing remaining useful life prediction, Measurement 170 (2021) 108753.

[25] Bin Zhang, Shaohui Zhang, Weihua Li, Bearing performance degradation assessment using long short-term memory recurrent network, Comput. Ind. 106 (2019) 14–29.

[26] Yongmeng Zhu, Jiechang Wu, Xing Liu, Jun Wu, Kai Chai, Gang Hao, Shuyong Liu, Hybrid scheme through read-first-LSTM encoder-decoder and broad learning system for bearings degradation monitoring and remaining useful life estimation, Adv. Eng. Inform. 56 (2023) 102014.

[27] Qing Ni, J.C. Ji, Ke Feng, Data-driven prognostic scheme for bearings based on a novel health indicator and gated recurrent unit network, IEEE Trans. Ind. Inform. 19 (2) (2022) 1301–1311.

[28] Jiahang Luo, Xu Zhang, Convolutional neural network based on attention mechanism and Bi-LSTM for bearing remaining life prediction, Appl. Intell. (2022) 1–16.

[29] Meng Ma, Zhu Mao, Deep-convolution-based LSTM network for remaining useful life prediction, IEEE Trans. Ind. Inform. 17 (3) (2020) 1658–1667.

[30] Jiusi Zhang, Jilun Tian, Minglei Li, Jose Ignaclo Leon, Leopoldo García Franquelo, Hao Luo, Shen Yin, A parallel hybrid neural network with integration of spatial and temporal features for remaining useful life prediction in prognostics, IEEE Trans. Instrum. Meas. 72 (2022) 1–12.

[31] Tian Han, Jiachen Pang, Andy C.C. Tan, Remaining useful life prediction of bearing based on stacked autoencoder and recurrent neural network, J. Manuf. Syst. 61 (2021) 576–591.

[32] Peter Shaw, Jakob Uszkoreit, Ashish Vaswani, Self-attention with relative position representations, 2018, arXiv preprint arXiv:1803.02155.

[33] Yudong Cao, Yifei Ding, Minping Jia, Rushuai Tian, A novel temporal convolutional network with residual self-attention mechanism for remaining useful life prediction of rolling bearings, Reliab. Eng. Syst. Saf. 215 (2021) 107813.

[34] Xuanyuan Su, Hongmei Liu, Laifa Tao, Chen Lu, Mingliang Suo, An end-to-end framework for remaining useful life prediction of rolling bearing based on feature pre-extraction mechanism and deep adaptive transformer model, Comput. Ind. Eng. 161 (2021) 107531.

[35] Yifei Ding, Minping Jia, Convolutional transformer: An enhanced attention mechanism architecture for remaining useful life estimation of bearings, IEEE Trans. Instrum. Meas. 71 (2022) 1–10.

[36] Jiusi Zhang, Xiang Li, Jilun Tian, Hao Luo, Shen Yin, An integrated multi-head dual sparse self-attention network for remaining useful life prediction, Reliab. Eng. Syst. Saf. 233 (2023) 109096.

[37] Li Jiang, Tianao Zhang, Wei Lei, Kejia Zhuang, Yibing Li, A new convolutional dual-channel transformer network with time window concatenation for remaining useful life prediction of rolling bearings, Adv. Eng. Inform. 56 (2023) 101966.

[38] Yupeng Wei, Dazhong Wu, Janis Terpenny, Constructing robust and reliable health indices and improving the accuracy of remaining useful life prediction, J. Nondestruct. Eval. Diagn. Progn. Eng. Syst. 5 (2) (2022) 021009.

[39] Jinwon An, Sungzoon Cho, Variational autoencoder based anomaly detection using reconstruction probability, Special Lect. IE 2 (1) (2015) 1–18.

[40] Kurt Hornik, Maxwell Stinchcombe, Halbert White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, Illia Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[42] Yupeng Wei, Dazhong Wu, Material removal rate prediction in chemical mechanical planarization with conditional probabilistic autoencoder and stacking ensemble learning, J. Intell. Manuf. (2022) 1–13.

[43] Diederik P. Kingma, Max Welling, Auto-encoding variational bayes, 2013, arXiv preprint arXiv:1312.6114.

[44] Yongzhen Huang, Zifeng Wu, Liang Wang, Tieniu Tan, Feature coding in image classification: A comprehensive study, IEEE Trans. Pattern Anal. Mach. Intell. 36 (3) (2013) 493–506.

[45] Patrick Nectoux, Rafael Gouriveau, Kamal Medjaher, Emmanuel Ramasso, Brigitte Chebel-Morello, Noureddine Zerhouni, Christophe Varnier, PRONOSTIA: An experimental platform for bearings accelerated degradation tests, in: IEEE International Conference on Prognostics and Health Management, PHM'12, IEEE Catalog Number: CPF12PHM-CDR, 2012, pp. 1–8.

[46] Yupeng Wei, Dazhong Wu, Janis Terpenny, Bearing remaining useful life prediction using self-adaptive graph convolutional networks with self-attention mechanism, Mech. Syst. Signal Process. 188 (2023) 110010.

[47] Kamran Javed, Rafael Gouriveau, Noureddine Zerhouni, Patrick Nectoux, A feature extraction procedure based on trigonometric functions and cumulative descriptors to enhance prognostics modeling, in: 2013 IEEE Conference on Prognostics and Health Management, Phm, IEEE, 2013, pp. 1–7.

[48] Yupeng Wei, Dazhong Wu, Remaining useful life prediction of bearings with attention-awared graph convolutional network, Adv. Eng. Inform. 58 (2023) 102143.

[49] Yupeng Wei, Dazhong Wu, Janis Terpenny, Robust incipient fault detection of complex systems using data fusion, IEEE Trans. Instrum. Meas. 69 (12) (2020) 9526–9534.

[50] Haobo Qiu, Yingchun Niu, Jie Shang, Liang Gao, Danyang Xu, A piecewise method for bearing remaining useful life estimation using temporal convolutional networks, J. Manuf. Syst. 68 (2023) 227–241.

[51] Rebecca Killick, Paul Fearnhead, Idris A. Eckley, Optimal detection of changepoints with a linear computational cost, J. Amer. Statist. Assoc. 107 (500) (2012) 1590–1598.

[52] Marc Lavielle, Using penalized contrasts for the change-point problem, Signal Process. 85 (8) (2005) 1501–1510.

[53] Wei Hao, Zhixuan Li, Guohao Qin, Kun Ding, Xuwei Lai, Kai Zhang, A novel prediction method based on bi-channel hierarchical vision transformer for rolling bearings' remaining useful life, Processes 11 (4) (2023) 1153.

[54] Shaoke Wan, Xiaohu Li, Yanfei Zhang, Shijie Liu, Jun Hong, Dongfeng Wang, Bearing remaining useful life prediction with convolutional long short-term memory fusion networks, Reliab. Eng. Syst. Saf. 224 (2022) 108528.

[55] Xiang Li, Wei Zhang, Qian Ding, Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction, Reliab. Eng. Syst. Saf. 182 (2019) 208–218.

[56] Xiang Li, Wei Zhang, Hui Ma, Zhong Luo, Xu Li, Data alignments in machinery remaining useful life prediction using deep adversarial neural networks, Knowl.-Based Syst. 197 (2020) 105843.

[57] Yudong Cao, Minping Jia, Peng Ding, Yifei Ding, Transfer learning for remaining useful life prediction of multi-conditions bearings based on bidirectional-GRU network, Measurement 178 (2021) 109287.

[58] Biao Wang, Yaguo Lei, Naipeng Li, Ningbo Li, A hybrid prognostics approach for estimating remaining useful life of rolling element bearings, IEEE Trans. Reliab. 69 (1) (2018) 401–412.

[59] Dongdong Zhao, Liu Feng, A two-stage machine-learning-based prognostic approach for bearing remaining useful prediction problem, IAENG Int. J. Comput. Sci. 48 (4) (2021).