CAAI Artificial Intelligence Research

Anomaly Detection in Long-term Time Series with Transformer and Variational Autoencoder

Xiankun Shi[1], Yibo Li[1], Ziqi Wang[2], Junqi Zhang[1], Jing Bi[1,3*], Nan Ma[4], Junfei Qiao[3,5]

ABSTRACT

Anomaly detection is crucial for ensuring system reliability and providing early warnings of potential failures. However, long-term time-series anomaly detection remains challenging because anomalous events are rare, anomaly patterns are diverse, and temporal dependencies can extend across multiple scales. Traditional methods and existing deep learning models still struggle to jointly represent fine-grained temporal variation, long-range contextual information, and the distribution of normal patterns. This work presents Transformer with Variational AutoEncoder (TransVAE), a framework that combines a dual-embedding representation pipeline, a Transformer encoder for long-range dependency modeling, and VAE-based latent regularization of encoded features. Specifically, local position-aware embeddings and feature-wise global embeddings are integrated into a shared representation before attention modeling, and the VAE is applied to the Transformer-encoded representation to regularize the latent space of normal patterns. Evaluations on benchmark time-series datasets show that TransVAE outperforms several state-of-the-art baselines, achieving improvements in the area under the receiver operating characteristic by 6.7% and 13.6%, respectively. Ablation and sensitivity analyses further support the contributions of each component of the framework. These results demonstrate that TransVAE provides an effective and robust solution for long-term time-series anomaly detection.

KEYWORDS

Anomaly detection, variational autoencoder, Transformer, self-attention mechanism, temporal embedding

Just Accepted

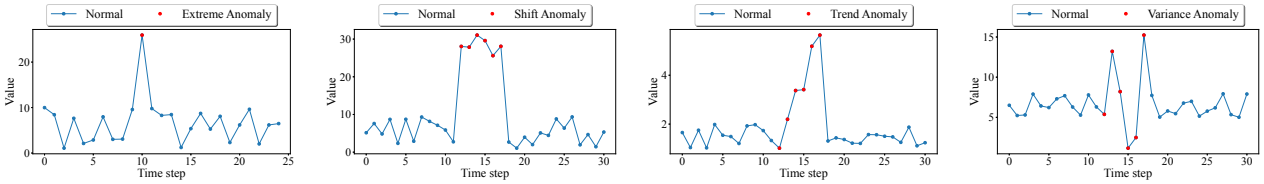
Anomaly detection [1] is essential for maintaining the reliability and operational integrity of complex systems, including industrial facilities, information infrastructures, and environmental monitoring networks. Equipment malfunctions, cyber-attacks, sensor failures, and abrupt changes in operating conditions may produce abnormal variations in key indicators, potentially affecting safety, service availability, and economic performance. Timely identification of such deviations supports early warning, impact assessment, and root-cause analysis, making accurate monitoring an important requirement for modern systems [2].

Traditional monitoring commonly relies on periodic inspection and manual analysis, which are costly and may not respond promptly to rapidly changing conditions. Advances in the Internet of Things (IoT) [3] and sensor technologies have enabled automated monitoring systems to continuously collect large volumes of multivariate time series data [4]. In water quality monitoring, for example, long-term sensor observations provide detailed information about environmental dynamics, but they also contain seasonal variations, periodic fluctuations, trend changes, and nonlinear patterns caused by interacting natural and operational factors [5, 6]. Abnormal observations embedded in these patterns may indicate changes in the monitored environment, sensor malfunctions, or data acquisition errors.

Detecting anomalies in such time series remains challenging for several reasons. First, anomalous events occur much less frequently than normal observations, resulting in highly imbalanced data. Second, anomalies may appear in different forms, including extreme, shift, trend, and variance anomalies, as illustrated in Fig. 1. Third, abnormal behavior may depend not only on an individual observation but also on its relationship with distant time steps. Effective anomaly detection therefore requires a model that can represent fine-grained temporal variation, capture long-range contextual dependencies, and distinguish abnormal deviations from complex normal patterns.

Existing anomaly-detection methods address these challenges from different perspectives. Statistical, distance-based, and density-based methods are effective for relatively simple patterns, but often struggle to model temporal dependencies in complex, high-dimensional time series [7]. Recurrent models such as RNNs, LSTMs, and GRUs can capture sequential context [8, 9, 10], although their sequential computation and gradient-related limitations reduce their effectiveness on long sequences [11]. Transformers use self-attention to directly model dependencies between distant time steps [12, 13], while

1 College of Computer Science, Beijing University of Technology, Beijing 100124, China.
2 School of Software Technology, Zhejiang University, Ningbo 315100, China.
3 Beijing Laboratory of Smart Environmental Protection, Beijing University of Technology, Beijing 100124, China.
4 College of Artificial Intelligence, Beijing University of Technology, Beijing 100124, China.
5 School of Information Science and Technology, Beijing University of Technology, Beijing 100124, China.
* Corresponding author: Jing Bi (bijing@bjut.edu.cn)



(a) Extreme anomaly (b) Shift anomaly (c) Trend anomaly (d) Variance anomaly

Fig. 1 Types of time series anomalies.

Variational Autoencoders (VAEs) learn latent distributions of normal patterns and identify deviations through reconstruction errors [14]. Nevertheless, existing approaches may not simultaneously preserve local temporal details, represent global sequence context, and regularize the distribution of normal patterns, leaving room for a more coordinated framework for long-term time-series anomaly detection.

Due to the challenges posed by long-term time-series anomaly detection, this work presents Transformer with Variational AutoEncoder (TransVAE). Rather than introducing a completely new fusion operator, TransVAE builds a coordinated representation pipeline in which local and global embeddings are first integrated into a shared representation, the Transformer encoder models temporal dependencies on top of that representation, and the VAE regularizes the encoded features in latent space. The main methodological extensions of the framework are summarized as follows.

1) A dual-embedding representation pipeline integrates position-aware local embeddings with feature-wise global embeddings so that both fine-grained temporal variation and sequence-level contextual information are encoded before attention modeling.

2) A Transformer encoder with multi-head self-attention serves as the backbone for modeling long-range temporal dependencies in multivariate time series.

3) A VAE-based latent regularization module operates on the Transformer-encoded representations to shape the latent distribution of normal patterns and improve the sensitivity of reconstruction-based anomaly detection.

These components jointly improve the model's ability to distinguish normal fluctuations from abnormal deviations across multiple temporal scales without changing the reconstruction-based anomaly scoring rule.

The structure of this paper is organized as follows. Section 1 presents an in-depth review of prior research and foundational work in the field. Section 2 provides a comprehensive discussion of the TransVAE framework, detailing its architectural components and theoretical underpinnings. Section 3 delivers the comprehensive analysis of experimental results, including performance metrics and comparative evaluations. Finally, Section 4 summarizes this work, discusses theoretical and practical implications, and suggests possible directions for upcoming investigations.

1 Related Work

1.1 Traditional Time Series Anomaly Detection

Performing anomaly detection on sequential temporal data has garnered significant attention in recent years. Conventional methods for anomaly detection have mainly depended on statistical approaches, including moving averages and the autoregressive integrated moving average (ARIMA) [15]. However, it aims to effectively address the nonlinearity inherent in time series data. Additionally, commonly used techniques include clustering models such as K-Means [16] and distance-based approaches like K-nearest neighbors (KNN) [17]. Both K-Means and KNN are sensitive to noise and outliers, which can lead to misassignments in clusters and adversely affect the accuracy of anomaly detection. The work in [18] proposes an isolation forest algorithm (iForest) for anomaly detection using binary trees. iForest exhibits linear time complexity and requires minimal memory resources. However, it primarily identifies global outliers and has limitations in detecting local outliers. Chen et al. [19] introduces the Local Outlier Factor (LOF) for identification of outliers, which evaluates variations in local density within the dataset. Despite its effectiveness in detecting local outliers, LOF is characterized by high time complexity.

Traditional models are often inadequate for effectively addressing nonlinear problems in time series analysis. To overcome this challenge, the work in [20] introduces a Deep Belief Network integrated with a One-Class Support Vector Machine model for unsupervised anomaly detection, demonstrating precision in processing high-dimensional data. However, this approach may exhibit reduced efficiency when handling large-scale datasets. The Gaussian mixture model-based detection approach for hyperspectral imaging anomaly detection, as proposed by Qu et al. [21], relies on the premise that the data conform to a mixture of Gaussian distributions. If a significant deviation from this assumption occurs, the model may face difficulties in effectively identifying the fundamental patterns within the data distributions, potentially compromising its overall performance. In summary, traditional time series anomaly detection models are limited to identifying a single anomalous point and are unable to detect the distribution of multiple types of abnormal time series segments.

1.2 Deep Learning-based Time Series Anomaly Detection

Neural network architectures excel at capturing the nonlinear characteristics present in temporal data. Fan et al. [22] developed a model based on neural network architecture that employs temporal convolutional networks (TCNs). However, TCNs encounter difficulties in leveraging long-term temporal dependencies due to the limitations associated with dilated convolutions. Numerous researchers widely adopt various variants

Not Accepted

of RNNs [23, 24, 25] due to their effectiveness in capturing the characteristics of time series data. Consequently, these networks are utilized across multiple domains, including natural language processing and sequential data prediction. Architectures such as LSTM networks [26, 27] effectively address the challenge of vanishing gradients inherent in conventional RNNs. Although these models excel at extracting temporal features, they often struggle to accurately estimate the hidden distribution of time series data from the latent space representation.

The widespread application of Generative Adversarial Networks (GANs) [28] in anomaly detection is primarily due to their ability to identify abnormalities through adversarial learning of sample representations. Li et al. [29] integrate LSTM, RNN, and GAN frameworks to explore time-dependent relationships within the distributions of sequential data. However, training GANs can be complex and unstable, as the competitive dynamics between the generator and discriminator may lead to training instability. VAE [30] represents another powerful class of generative models that can learn latent representations and reconstruct data. The work in [31] employs an LSTM and VAE-based (LSTM-VAE) detector that utilizes a score for anomaly detection derived from reconstruction, combined with a state-dependent threshold for anomaly detection. VAE offers advantages in probabilistic modeling of data distributions, making it particularly suited for anomaly detection tasks where understanding normal patterns is crucial. However, LSTM-VAE incurs a high computational cost, especially when processing long sequences, which can result in decreased efficiency during both training and inference.

Transformer models [32, 33] demonstrate robust capabilities in feature modeling through their self-attention mechanisms. The study in [34] introduces the Informer model, which employs Kullback-Leibler (KL) [35] divergence to improve the computational efficiency of self-attention. However, the Informer model is unable to decompose the periodic and trend components of time series features. In contrast, the research in [36] innovatively integrates the Transformer architecture into a decomposition forecasting framework known as Autoformer. This architecture effectively separates long-term trend information.

Unlike the Autoformer model, reconstruction-based methods [37] involve training a model to reconstruct normal samples, with anomalies identified as instances that the trained model cannot adequately reconstruct. The study in [38] introduces the Anomaly Transformer, a reconstruction-based method that features a novel anomaly attention mechanism designed to compute association discrepancies. Although this approach does not directly incorporate details regarding the position of the input sequence, it employs positional embeddings to represent the relative positions within the sequence, which may inadequately address temporal dependencies. In summary, temporal anomaly detection models utilizing Transformers offer significant advantages over other deep learning models. However, the position encoding of standard Transformer architectures limits their effectiveness in capturing temporal relationships across various time scales simultaneously.

Compared with Anomaly Transformer, TransVAE introduces a local–global dual-embedding pipeline before attention modeling, so that both fine-grained temporal variation and sequence-level contextual information are encoded in the shared representation. Compared with LSTM-VAE, TransVAE uses a Transformer encoder to model long-range dependencies and then applies VAE-based latent regularization to the encoded representations rather than relying on recurrent feature extraction. Compared with standard reconstruction-based Transformer models, TransVAE combines reconstruction learning with latent distribution regularization instead of relying on reconstruction alone. In this sense, the Transformer with multi-head self-attention serves as the temporal modeling backbone, whereas the main methodological extensions of the framework lie in the dual-embedding representation pipeline and the VAE-based regularization of the encoded features.

Therefore, TransVAE combines the two models so that temporal dependencies and normal-data distributions can be learned within a unified framework.

Table 1 Main notations

Notation	Definition
$X \in \mathbb{R}^{T\times\mathcal{F}}$	Time series data
$\mathcal{F}$	Number of features
T	Number of time steps
$X_{t,:}$	Feature vector at time step t
$X_{:,n}$	Time series of feature n
ψ	Position of input sequence
$x_t \in \mathbb{R}^{\mathcal{F}}$	Indicator value at time step t
D	Dimension of the model
τ	Attention head number
H	Output of multi-head self-attention
$\bar{H}$	Output of residual network
z	Latent space vector
D_z	Dimension of latent space
$\hat{X}$	Reconstructed value
$\hat{Y}$	Binary detection result
η	Predefined threshold

2 Model Framework

This section presents the proposed TransVAE for time-series anomaly detection. As illustrated in Fig. 2, the framework follows an explicit information-flow path: local and global embeddings are integrated into a shared representation, the Transformer encoder models temporal dependencies on that representation, and the VAE regularizes the encoded features before final reconstruction-based anomaly scoring. After introducing the notation and problem statement, we detail the embedding design, the Transformer backbone, the VAE-based latent regularization module, and the anomaly detection strategy.

2.1 Notations and Problem Statement

A multivariate time series is represented by a matrix $X \in \mathbb{R}^{T\times\mathcal{F}}$, where T is the number of time steps and $\mathcal{F}$ is the number of features. The vector of features at a given time step t, corresponding to the t-th row of the matrix, is denoted by $X_{t,:}$. For simplicity in notation, we also represent this vector as $x_t \in \mathbb{R}^{\mathcal{F}}$. Conversely, the complete time series for the n-th feature is denoted by $X_{:,n}$, which is the n-th column of X. The objective is to detect whether the time series data X exhibits

Just Accepted

abnormal behavior, with $\hat{Y} \in \{0, 1\}$ denoting the binary detection result. If X is an abnormal sequence, $\hat{Y} = 1$; otherwise, $\hat{Y} = 0$. Additionally, $z_t \in \mathbb{R}^{D_z}$ represents a latent space vector at time step t. Table 1 summarizes the main notations applied throughout this paper.

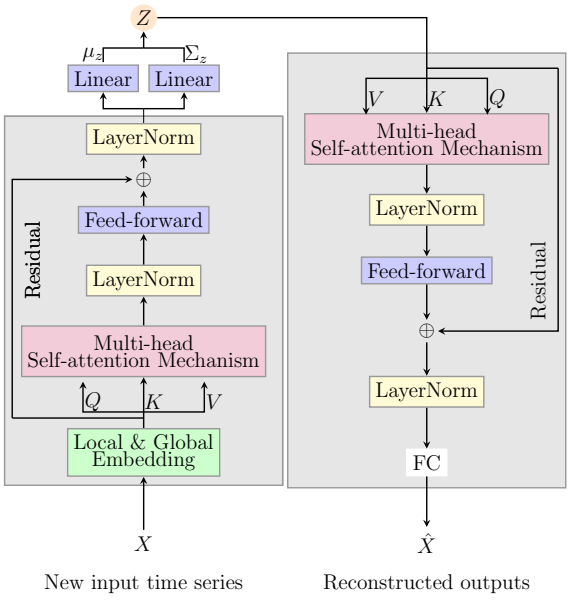


Fig. 2 Structure of TransVAE.

2.2 Transformer Mechanism

This work implements a Transformer-based architecture for anomaly detection in time series data. The architecture follows an encoder-decoder structure, with each block comprising two primary components: a multi-head self-attention module that identifies dependencies across different time steps, and a position-specific feed-forward neural network (FNN) to further refine these relationships. Layer normalization and residual connections connect these components, enhancing both training stability and overall model performance. This design effectively enables the model to discern complex temporal relationships in time series data, which is crucial for identifying various types of anomalies.

2.2.1 Time Series Local Embedding

A fundamental limitation of standard self-attention mechanisms is their permutation-invariant nature, which sharply contrasts with recurrent architectures like LSTM that inherently process data sequentially. In contrast to LSTM, self-attention lacks the intrinsic ability to incorporate information about temporal steps, treating each element in a sequence independently and without regard for their chronological order. This limitation presents a significant challenge when analyzing time series data, where understanding temporal relationships is crucial for accurate representation and anomaly detection.

To address this limitation, we incorporate positional encoding [34] into the input embeddings to integrate temporal information. This method allows the model to distinguish tokens based on their position within the sequence, thereby preserving essential temporal dependencies. The positional encoding operates as follows.

$$\begin{cases} P_{\psi,2i} = \sin\left(\dfrac{\psi}{10000^{2i/D}}\right) \\ P_{\psi,2i+1} = \cos\left(\dfrac{\psi}{10000^{2i/D}}\right) \end{cases} \tag{1}$$

where $P_{\psi,2i}$ and $P_{\psi,2i+1}$ represent the $2i$-th and $(2i+1)$-th components of the positional encoding vector at position ψ, respectively. The parameter ψ denotes the sequential position within the input data, ranging from 0 to the sequence length minus one.

Sinusoidal encoding distinguishes time steps and enables relative positions to be inferred without introducing learned positional parameters.

The input feature vectors are first projected into the model's D-dimensional space and then enhanced by incorporating positional encoding, resulting in a representation that simultaneously captures both feature information and temporal position. Then,

$$\mathcal{X}_l = \text{Linear}(X) + P_\psi \tag{2}$$

where $X \in \mathbb{R}^{T \times \mathcal{F}}$ represents an original feature vector, and $\text{Linear}(\cdot)$ is a learnable embedding layer that projects the input to the dimension $\mathbb{R}^{T \times \mathcal{F} \times D}$. The positional encoding P_ψ is broadcasted to this shape before addition. This approach effectively transforms the position-agnostic self-attention mechanism into one that is position-aware, enabling the model to recognize and utilize temporal patterns in time series data.

2.2.2 Time Series Global Embedding

While local positional encoding effectively addresses the immediate challenge of incorporating temporal orders, it remains inadequate for capturing the global contextual patterns that span the entire time series. Most Transformer-based models apply positional encoding exclusively to individual time steps within a sequence, neglecting to represent the overall contextual position of the sequence within a broader temporal framework. As a result, tokens generated from individual time steps may struggle to capture global sequence information essential for accurate anomaly detection.

This limitation significantly constrains the Transformer's ability to learn comprehensive representations of time series data, particularly when addressing complex anomaly patterns that emerge over extended periods. The lack of global context diminishes the model's generalization capabilities across various time series datasets and types of anomalies, including extreme, shift, trend, and variance anomalies.

To solve this limitation, this work employs a global embedding mechanism that encodes the entire time series of each feature into a unified variate token, improving the model's capability to capture long-term relationships. The global embedding for each feature n is formulated as follows:

$$\mathcal{X}_{g,n} = \text{MLP}(X_{:,n}) \tag{3}$$

where $X_{:,n} \in \mathbb{R}^T$ denotes the complete time series of the n-th feature across all T time steps, capturing the feature's overall temporal pattern. The multi-layer perceptron (MLP) transforms the entire feature sequence from length T into a compact D-dimensional representation as follows. Then,

Just Accepted

$$\mathrm{MLP}(X_{:,n})=W_2 \cdot \mathrm{ReLU}(W_1 \cdot X_{:,n}+b_1)+b_2 \tag{4}$$

where $W_1 \in \mathbb{R}^{D_h \times T}$ and $W_2 \in \mathbb{R}^{D \times D_h}$ are learnable weight matrices, and $b_1 \in \mathbb{R}^{D_h}$ and $b_2 \in \mathbb{R}^{D}$ are bias vectors. The parameter D_h represents the hidden layer dimension, which determines the intermediate representation size. ReLU[39] introduces nonlinear characteristics into the model, enabling the network to capture intricate temporal patterns across the entire sequence. This architecture effectively compresses the time series from its original length T to a fixed dimension D, creating a representation $\mathcal{X}_{g,n} \in \mathbb{R}^D$ that encapsulates the comprehensive contextual information of feature n throughout the entire time series.

The complete global embedding matrix $\mathcal{X}_g \in \mathbb{R}^{\mathcal{F} \times D}$ is formed by stacking the global embeddings of all $\mathcal{F}$ features, followed by a transpose operation to ensure proper dimensional alignment. Then,

$$\mathcal{X}_g=[\mathcal{X}_{g,1}, \mathcal{X}_{g,2}, \ldots, \mathcal{X}_{g,\mathcal{F}}]^T \tag{5}$$

2.2.3 Integration of Local and Global Embeddings

In TransVAE, the integration stage is designed as a coordinated representation step rather than a completely new fusion operator. The local embedding $\mathcal{X}_l \in \mathbb{R}^{T \times \mathcal{F} \times D}$ captures position-aware temporal variation, whereas the global embedding $\mathcal{X}_g \in \mathbb{R}^{\mathcal{F} \times D}$ summarizes feature-wise sequence context across the full time horizon. To make these two sources of information interact in the same feature space, the global embedding is first expanded along the time dimension to form $\mathcal{X}_g^e \in \mathbb{R}^{T \times \mathcal{F} \times D}$. The expanded global embedding is then added to the local embedding and normalized, yielding the shared representation $\mathcal{X}$ that is passed to the Transformer encoder. Concretely, the local embedding and the expanded global embedding are first integrated into the shared representation $\mathcal{X}$, which is then fed into the Transformer encoder; the VAE is subsequently applied to the Transformer-encoded representation rather than directly to the raw input. In this way, the integration mechanism should be understood as an explicit information-flow coupling step that prepares a unified representation for downstream temporal modeling and latent regularization.

To perform this integration, we need to first align the dimensions of the local and global embeddings. The local embedding $\mathcal{X}_l \in \mathbb{R}^{T \times \mathcal{F} \times D}$ encodes position-aware features across time steps, while the global embedding $\mathcal{X}_g \in \mathbb{R}^{\mathcal{F} \times D}$ captures sequence-level patterns for each feature. To ensure compatibility, the global embedding is expanded along the time dimension to create a new tensor $\mathcal{X}_g^e \in \mathbb{R}^{T \times \mathcal{F} \times D}$. The components of this expanded tensor are defined by replicating the values of $\mathcal{X}_g$ for each time step:

$$(\mathcal{X}_g^e)_{t,f,d} = (\mathcal{X}_g)_{f,d} \tag{6}$$

where $t \in \{1, \ldots, T\}$, $f \in \{1, \ldots, \mathcal{F}\}$, and $d \in \{1, \ldots, D\}$. This operation ensures that each time step has access to the same global feature representations, preserving the feature-specific global context while making it compatible with the time step dimension of the local embedding.

The integration of these complementary embeddings is achieved through a feature-wise addition followed by layer normalization [40] to stabilize the combined representation:

$$\mathcal{X}=\mathrm{LayerNorm}(\mathcal{X}_l+\mathcal{X}_g^e) \tag{7}$$

where $\mathrm{LayerNorm}(\cdot)$ denotes layer normalization, which stabilizes the training process and accelerates convergence by normalizing the inputs across features.

This additive integration approach offers several key advantages. The method maintains dimensional consistency between local and global representations, ensuring proper alignment of feature spaces throughout the network. It facilitates efficient gradient flow through both embedding paths during backpropagation, which enhances training stability and convergence. Furthermore, this approach enables balanced contribution from each representation to the final embedded representation $\mathcal{X} \in \mathbb{R}^{T \times \mathcal{F} \times D}$, allowing the model to capture both local temporal patterns and overarching contextual insights in detecting anomalies. The combined embedding $\mathcal{X}$ is subsequently input into the multi-head self-attention module, which further processes these enriched representations to effectively model the temporal dependencies within the time series.

2.2.4 Multi-head Self-attention Module

Following the application of the integrated embedding approach, the complete sequence data $\mathcal{X}$ undergoes linear projection to generate the value (V), key (K), and query (Q) matrices. These matrices are fed into the self-attention module, which enables effective modeling of complex temporal relationships within the data. The mechanism is mathematically formulated as:

$$\mathrm{Attention}(Q, K, V)=\mathrm{Softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{8}$$

where $V \in \mathbb{R}^{T \times D}$, $K \in \mathbb{R}^{T \times D}$, and $Q \in \mathbb{R}^{T \times D}$ denote the value, key, and query matrices, respectively. The parameter D denotes the model dimension, and T represents the sequence length. The term d_k denotes the dimensionality of the key vectors and acts as a scaling factor to avoid overly large dot product values, which might lead to numerical instability. Attention scores are normalized with the softmax function, which ensures they sum to one across all time steps.

The self-attention module acts by enabling each query to extract pertinent information from all key-value pairs, resulting in a contextually enriched representation. This process enables the model to selectively emphasize the most relevant elements of the input sequence during prediction or reconstruction, thereby effectively capturing both short-term and long-range relationships within the time series data. To enhance the model's representational capacity, we implement a multi-head attention module that can simultaneously focus on information from different representational subspaces. This approach is defined as:

$$\mathrm{MultiHead}(Q, K, V)=\mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_\tau)W_O \tag{9}$$

where $\mathrm{head}_j=\mathrm{Attention}(Q_j, K_j, V_j)$, which denotes the output produced by the j-th attention head, τ is the count of attention heads, and W_O is a learnable parameter matrix that projects the concatenated outputs to the desired dimensionality. Each attention head can emphasize different aspects of the temporal

Just Accepted

relationships, collectively providing a comprehensive representation of the input sequences.

The multi-head attention module greatly improves the model's capacity to capture diverse patterns, especially when addressing complex anomalies that may manifest differently across various time scales and features.

2.2.5 Feed-forward and Residual Network

To further enhance the model's representational ability, we incorporate a feed-forward neural network (FFN), expressed as:

$$\text{FFN}(H)=\text{ReLU}(HW_1+b_1)W_2+b_2 \tag{10}$$

where H represents the output generated by the multi-head self-attention mechanism. The matrices $W_1 \in \mathbb{R}^{D_m \times D_f}$ and $W_2 \in \mathbb{R}^{D_f \times D_m}$, along with the bias terms $b_1 \in \mathbb{R}^{D_f}$ and $b_2 \in \mathbb{R}^{D_m}$, are learnable parameters. D_m and D_f denote the model dimension and the feed-forward network dimension, respectively. The ReLU activation function introduces nonlinearity, allowing the network to model intricate patterns within the input sequence. To mitigate the difficulties in training deep networks, particularly the problem of vanishing gradients that can affect the detection of long-term anomalies, we implement a residual network architecture. This approach facilitates the flow of gradients throughout the network by establishing a direct pathway for information propagation. The residual network is defined as:

$$\bar{H}=\text{LayerNorm}(\text{FFN}(H)+\mathcal{X}) \tag{11}$$

where $\mathcal{X}$ represents the original input sequence after embedding, and $\bar{H}=\{\bar{h}_1, \dots, \bar{h}_T\}$ is the output of the residual network. This residual connection enables the model to preserve attention on the original temporal patterns while simultaneously learning more abstract representations, thereby enhancing its ability in anomaly detection on time series.

The TransVAE model's integration of local and global embeddings, together with the Transformer backbone and residual connections, forms a coherent architecture for anomaly detection in time series. By jointly modeling fine-grained temporal patterns and broader contextual relationships, the framework is better aligned with the challenges posed by anomalies that emerge across different temporal scales.

2.3 Variational AutoEncoder

Following the Transformer encoder, the VAE module receives the encoded representation $\bar{H}$ rather than the raw input sequence. In the proposed framework, temporal dependency modeling is primarily handled by the Transformer encoder, while the role of the VAE is to regularize the latent space of the Transformer-encoded features and to model the distributional structure of normal patterns during training. This design choice does not imply that VAE is universally superior to all alternative generative models; rather, VAE is adopted here because it provides a tractable probabilistic regularization mechanism that aligns naturally with the Transformer-based representation learning pipeline. The KL constraint shapes a smoother latent manifold for normal behaviors, and the reconstruction term encourages the encoded features to preserve the structure needed for reconstructing normal sequences. As a result, abnormal deviations are more likely to yield larger reconstruction discrepancies after training, even though the anomaly score itself remains reconstruction-based.

To enhance the model's robustness, we introduce Gaussian noise to this input, defined as $\tilde{h}_t=\bar{h}_t+\epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_{noise})$, resulting in a noise-augmented representation $\tilde{H}=\{\tilde{h}_1, \dots, \tilde{h}_t, \dots, \tilde{h}_T\}$. This regularization technique prevents overfitting while preserving essential structural information. The noise parameter $\sigma_{noise}=0.01$ is empirically determined through cross-validation.

The VAE then maps the noise-augmented representation $\tilde{h}_t$ to a probability distribution in the latent space. Using $\tilde{h}_t$ as input, we derive the approximate posterior distribution $\tilde{q}_\phi(z_t|\tilde{h}_t)$ by estimating its mean μ_{z_t} and covariance Σ_{z_t} through linear projections. Crucially, the model is trained to reconstruct the original, clean data $\bar{h}_t$ from the latent representation z_t. This objective is formulated by maximizing the variational lower bound of the log-likelihood:

$$\mathcal{L}_{vae}=-D_{KL}(\tilde{q}_\phi(z_t|\tilde{h}_t)||p_\theta(z_t))+\mathbb{E}_{\tilde{q}_\phi(z_t|\tilde{h}_t)}[\log p_\theta(\bar{h}_t|z_t)] \tag{12}$$

where the first term is the KL divergence that regularizes the latent space, and the second is the reconstruction term. This second term quantifies the model's ability to reconstruct the original data $\bar{h}_t$ from a latent space conditioned on the noisy input $\tilde{h}_t$.

The prior distribution $p_\theta(z_t)$ is modeled as a Gaussian $\mathcal{N}(\mu_p, I)$. The KL divergence term is then computed as:

$$D_{KL}(\tilde{q}_\phi(z_t|\tilde{h}_t)||p_\theta(z_t)) \approx D_{KL}(\mathcal{N}(\mu_{z_t}, \Sigma_{z_t})||\mathcal{N}(\mu_p, I)) \tag{13}$$

$$=\frac{1}{2}((\mu_p-\mu_{z_t})^{\mathrm{T}}(\mu_p-\mu_{z_t})+\text{tr}(\Sigma_{z_t})-\log|\Sigma_{z_t}|-D_z)$$

where D_z is the dimension of the latent variable z_t.

For the reconstruction term, the conditional distribution $p_\theta(\bar{h}_t|z_t)$ is modeled as a multivariate normal distribution $\mathcal{N}(\mu_{\bar{h}_t}, \Sigma_{\bar{h}_t})$. The expectation is then given by:

$$\mathbb{E}_{\tilde{q}_\phi(z_t|\tilde{h}_t)}[\log p_\theta(\bar{h}_t|z_t)] \tag{14}$$

$$\approx -\frac{1}{2}(\log(|\Sigma_{\bar{h}_t}|)+(\bar{h}_t-\mu_{\bar{h}_t})^{\mathrm{T}}\Sigma_{\bar{h}_t}^{-1}(\bar{h}_t-\mu_{\bar{h}_t})+D\log(2\pi))$$

where D is the model dimension of the hidden state $\bar{h}_t$, which is distinct from the latent dimension D_z. The expectation is approximated by sampling z_t from the posterior using the reparameterization trick.

The VAE's decoder transforms a sampled latent variable z_t back into the hidden space, generating reconstructed hidden states $\hat{H}=\{\hat{h}_1, \dots, \hat{h}_t, \dots, \hat{h}_T\}$. Each $\hat{h}_t$ here represents the reconstruction of the original Transformer encoder output $\bar{h}_t$. Finally, a linear output layer projects these reconstructed hidden states back to the original data dimension, generating the final reconstructed time series $\hat{X}$:

$$\hat{X} = W_x\hat{H} + b_x \tag{15}$$

where W_x and b_x are the learnable weight and bias parameters of the output layer. The model can then compare the original input X with its final reconstruction $\hat{X}$.

Just Accepted

2.4 Training Procedure

The training procedure of TransVAE optimizes a dual objective function that balances reconstruction accuracy with distribution modeling. The detailed steps of this process are given in Algorithm 1, which illustrates how the model integrates embedding generation, attention computation, latent variable sampling, and loss optimization during training.

The total loss combines VAE loss $\mathcal{L}_{vae}$ from (12) with MSE reconstruction loss $\mathcal{L}_{recon}$ from (16) [41], i.e.,

$$\mathcal{L}_{recon}=\frac{1}{N}\left\|X-\hat{X}\right\|_2^2 \tag{16}$$

MSE is particularly appropriate for time series data as it effectively captures deviations across multiple measurement scales while maintaining differentiability for gradient-based optimization. The total objective combines reconstruction learning with latent distribution regularization. In this formulation, $\mathcal{L}_{recon}$ preserves fidelity to the observed time series, whereas $\mathcal{L}_{vae}$ regularizes the encoded representations through the KL term and the reconstruction of clean hidden states from latent variables. This training strategy improves the separation between normal fluctuations and abnormal deviations in the learned representation space, while keeping the final anomaly score consistent with the reconstruction MSE used at inference time.

$$\mathcal{L}_{total}=\alpha\mathcal{L}_{recon}+(1-\alpha)\mathcal{L}_{vae} \tag{17}$$

where α represents a hyperparameter that regulates the trade-off between reconstruction accuracy and distribution learning. This formulation enables the network to simultaneously optimize for accurate time series reconstruction and effective modeling of the underlying probability distribution. The optimization of model parameters employs the Adam optimizer [42, 43], which incorporates adaptive learning rates with momentum. This optimization approach is selected for its effectiveness in training deep architectures with complex loss landscapes, facilitating more rapid convergence and enhanced training stability compared to conventional gradient descent methods.

Algorithm 1 outlines the training procedure for TransVAE. The core of the training involves a forward pass, where the model generates a reconstruction of the input, followed by a backward pass to update the weights. In the forward pass, the model integrates multi-scale temporal features, processes them through its attention-based architecture, and reconstructs the data via the VAE module. In the backward pass, gradients are computed using the standard backpropagation algorithm and applied using the Adam optimizer. Once training is complete, the resulting model can be used for anomaly detection as described in the next section.

Algorithm 1 Training Process of TransVAE

Input: Time series data $X=\{x_1,\dots,x_T\}$
Output: Trained TransVAE model parameters Θ

```
Initialize model parameters Θ
for each epoch do
    Compute integrated embedding 𝒳 by combining local and global embeddings with (7).
    Process 𝒳 through the Transformer Encoder to generate the hidden representation H̄.
    Estimate posterior parameters (μ_z, Σ_z) from H̄.
    Sample latent variable z ~ 𝒩(μ_z, Σ_z) via the reparameterization trick.
    Reconstruct the final time series X̂ using the VAE decoder and the output layer with (15).
    Calculate the total loss ℒ_total(X, X̂) with (17).
    Compute gradients ∇_Θ ℒ_total via backpropagation.
    Use Adam optimizer to minimize ℒ_total.
end for
return Trained model parameters Θ.
```

2.5 Anomaly Detection

The anomaly detection methodology in TransVAE leverages the model's reconstruction capabilities. During inference, the model processes an input time series X and generates its reconstruction $\hat{X}$. The discrepancy between the input and its reconstruction is quantified using an anomaly score. To maintain consistency with the training objective, this score is defined as MSE:

$$\text{AnomalyScore}(X)=\frac{1}{N}\left\|X-\hat{X}\right\|_2^2 \tag{18}$$

where N is the total number of elements in X.

A high anomaly score signifies that the model struggled to reconstruct the input, which suggests that the input's underlying patterns deviate significantly from the normal patterns learned during training. A time series segment is classified as anomalous if its score exceeds a predefined threshold η:

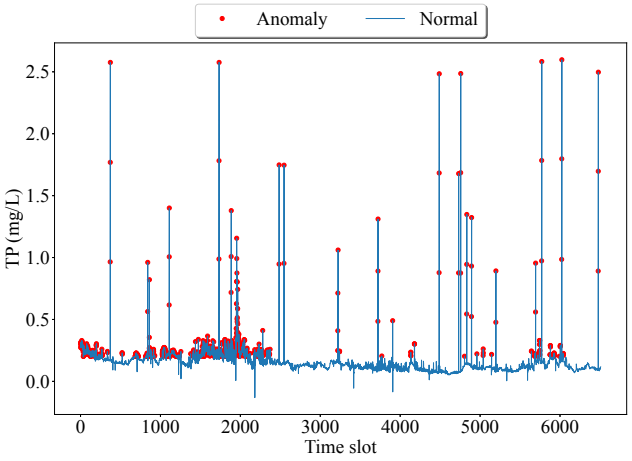
$$\hat{Y}=\begin{cases}1:\text{anomaly}, & \text{if AnomalyScore}(X)>\eta\\ 0:\text{normal}, & \text{otherwise}\end{cases} \tag{19}$$

The threshold η is determined empirically from a validation set composed entirely of normal data. Specifically, after computing the mean μ and standard deviation σ of these scores, the threshold can be defined as $\eta=\mu+k\sigma$. In our experiments, we found that setting the threshold to $\eta=0.15$ for the Wucun dataset and $\eta=0.18$ for the Gubeikou dataset yielded a good balance between detection sensitivity and false alarm rates.

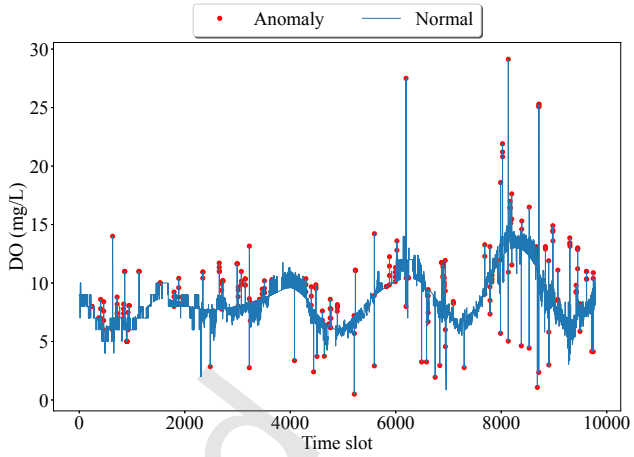
Table 4 Statistics of Datasets

Parameter	Datasets	
	Wucun	Gubeikou
Time span	Aug. 31, 2018–Dec. 12, 2021	Apr. 2, 2014–Sept. 17, 2018
Time interval	4 hours	4 hours
Data length	6,510	9,778
Water indicator	TP	DO
Anomaly ratio	1.7%	1.8%

Just Accepted



(a) Water quality dataset of TP in Wucun area



(b) Water quality dataset of DO in Gubeikou area

Fig. 3 Water quality datasets

Table 2 Comparison among TransVAE and other baseline models

Models	TP in Wucun area			DO in Gubeikou area		
	Accuracy	F1-score	AUROC	Accuracy	F1-score	AUROC
LOF	0.768	0.353	0.598	0.715	0.473	0.602
KNN	0.742	0.338	0.583	0.674	0.407	0.618
Autoencoder	0.793	0.374	0.613	0.799	0.594	0.711
K-Means	0.244	0.317	0.385	0.348	0.311	0.385
LSTM-VAE	0.684	0.493	0.654	0.539	0.467	0.682
Random Forest	0.739	0.416	0.629	0.610	0.479	0.589
Anomaly Transformer	0.689	0.536	0.689	0.719	0.605	0.696
TransVAE	0.793	0.606	0.735	0.782	0.739	0.808

Table 3 Ablation experiments with different configurations of modules.

Datasets	Models	Accuracy	F1-score	AUROC
TP in Wucun area	TransVAE-novae	0.785	0.458	0.645
	TransVAE-noembedding	0.747	0.588	0.729
	TransVAE-nores	0.651	0.503	0.661
	TransVAE	0.793	0.606	0.735
DO in Gubeikou area	TransVAE-novae	0.712	0.613	0.700
	TransVAE-noembedding	0.698	0.645	0.720
	TransVAE-nores	0.699	0.643	0.719
	TransVAE	0.782	0.739	0.808

3 Experimental Evaluation

3.1 Experimental Setup

The implementation of TransVAE, along with several benchmark models, is conducted with the PyTorch library. We perform the experiments on a server equipped with an NVIDIA RTX 3090 GPU and an Intel Xeon 6248R processor, evaluating multiple baseline models to identify the optimal parameter settings for each configuration. The effectiveness of TransVAE is demonstrated through the analysis of two real-world water quality datasets collected from the Gubeikou and Wucun regions. The dataset for Gubeikou water quality consists of 9,778 recorded dissolved oxygen (DO) samples collected from the Chaohe River in Beijing, China, between April 2, 2014, and September 17, 2018. Additionally, the water quality dataset includes 6,510 total phosphorus (TP) data points from the Wucun area in China, collected from August 31, 2018, to December 12, 2021. Water quality time series data are recorded every four hours, with further details provided in Table 4 and Fig. 3. The dataset is partitioned into training, testing, and validation subsets in a ratio of 7:2:1, respectively. The split is performed

Just Accepted

chronologically rather than randomly to preserve temporal order and prevent future observations from leaking into earlier subsets. These datasets are solely for method validation purposes. The detection of anomalies does not necessarily indicate water quality issues, and the results have not been compared against the relevant national standard (GB 3838-2002 Environmental quality standards for surface water).

3.2 Evaluation Metrics

We employ three metrics to assess the effectiveness of the TransVAE model in anomaly detection, which includes Accuracy, F1-Score, and area under the receiver operating characteristic (AUROC).

$$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{20}$$

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}} \tag{21}$$

$$\text{F1}-\text{score}=\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}\times 2 \tag{22}$$

$$\text{Accuracy}=\frac{\text{TP}+\text{TN}}{\text{TS}} \tag{23}$$

where TP denotes the number of positive samples correctly classified as positive, while FP refers to negative instances that are mistakenly categorized as positive. Conversely, FN represents positive samples incorrectly assigned as negative, and TN captures negative instances accurately recognized as negative, demonstrating correct rejections. TS is the total number of samples, and TS=TP+FP+FN+TN. Accuracy is measured as the proportion of correctly identified anomalies and normal instances among all samples classified by the model. AUROC measures the model's ability to distinguish between normal and anomalous samples across different threshold values, with higher values indicating better discriminative capability.

3.3 Hyperparameter Setting

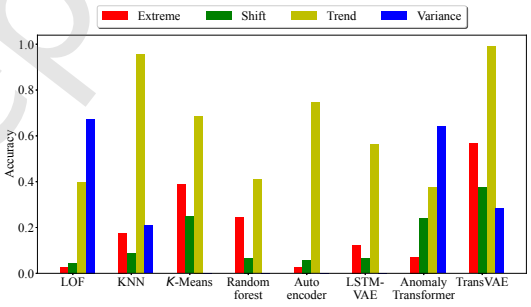
TransVAE has multiple hyperparameters that need to be configured. The batch size is fixed at 32, while the number of time steps T is selected from the set $\{10, 20, 30, 40\}$. The hidden size D of TransVAE is selected from the options $\{16, 32, 64, 128, 256\}$. The number of heads τ is selected from $\{2, 4, 6, 8, 10\}$. During the hyperparameter tuning phase, the optimizer starts with a learning rate of 0.01, which is subsequently reduced by 1×10^{-6} after every 20 epochs.

The final configurations used in the reported experiments are $(T, D, \tau)=(40, 32, 2)$ for the Wucun dataset and $(T, D, \tau)=(40, 64, 2)$ for the Gubeikou dataset.

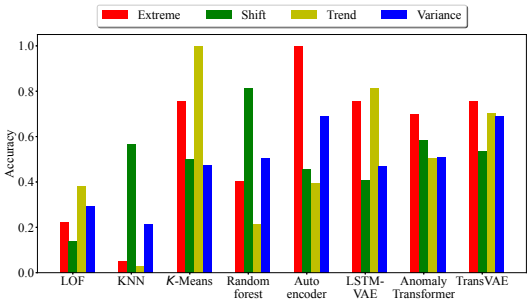
3.4 Benchmark Methods

TransVAE is benchmarked against seven representative anomaly-detection models: the traditional models LOF, KNN, K-Means, and Random Forest, and the deep learning models Autoencoder, LSTM-VAE, and Anomaly Transformer.

1) LOF [19]. LOF identifies outliers by local density deviations from neighboring points.
2) KNN [17]. KNN is a supervised learning classifier that classifies data based on K nearest neighbors.
3) K-Means [16]. K-Means is a clustering method that divides data into K separate clusters by iteratively reducing intra-cluster variance.
4) Random forest [44]. Random forest is a meta-estimator, utilizing multiple decision tree classifiers trained on diverse data subsets. It employs averaging to enhance prediction accuracy while managing the risk of overfitting.
5) Autoencoder [45]. Autoencoder detects anomalies by reconstruction with a standard autoencoder.
6) LSTM-VAE [46]. LSTM-VAE adopts LSTM to generate local features and uses VAE to estimate the long-term correlations of the sequence.
7) Anomaly Transformer [38]. A reconstruction-based method with a novel anomaly attention mechanism is designed to compute association discrepancies.



(a) Accuracy for the TP dataset in the Wucun area



(b) Accuracy for the DO dataset in the Gubeikou area

Fig. 4 Accuracy comparison of different anomaly types

3.5 Quantitative Comparison

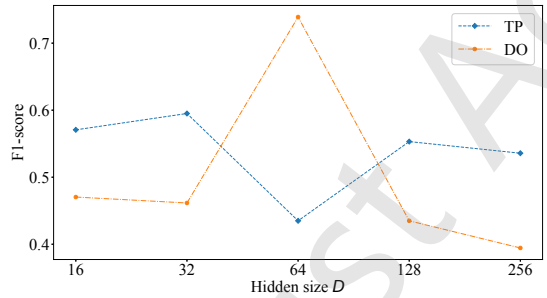
As presented in Table 2, TransVAE demonstrates stronger overall anomaly-detection performance than the baseline models across multiple evaluation metrics. This improvement should be interpreted in connection with the framework design rather than as an isolated numerical gain. Compared with purely reconstruction-based baselines, the proposed framework combines shared local–global representation learning with latent regularization of the encoded features. Compared

Just Accepted

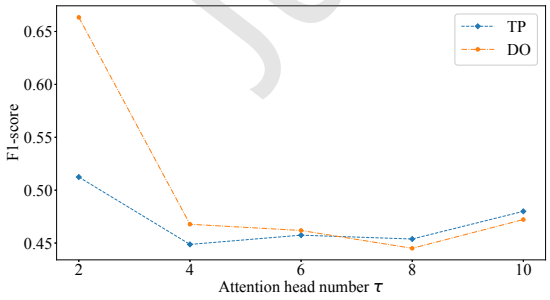
with LSTM-VAE, the Transformer backbone is better suited to modeling long-range dependencies in long sequences, while the VAE continues to regularize the representation space of normal patterns. Accordingly, the improvements in F1-score and AUROC are consistent with the goal of distinguishing abnormal deviations from normal fluctuations in imbalanced anomaly-detection settings.

On the Gubeikou dataset, TransVAE obtains a slightly lower Accuracy than the Autoencoder baseline (0.782 versus 0.799), but substantially higher F1-score (0.739 versus 0.594) and AUROC (0.808 versus 0.711). This result suggests that the small reduction in overall classification accuracy is accompanied by a better precision–recall balance and stronger discrimination between normal and anomalous samples across different thresholds. The three metrics therefore provide complementary perspectives on the anomaly-detection performance of the models.

The accuracy of TransVAE in comparison to benchmark models across various types of anomalies is shown in Fig. 4. The anomalies present in the Wucun and Gubeikou datasets include extreme, shift, trend, and variance anomalies. Although TransVAE demonstrates lower accuracy than some benchmark models for extreme and trend anomalies, its overall anomaly detection accuracy remains higher than that of the other benchmark models in both datasets. The relatively lower performance on extreme anomalies is consistent with the smoothing effect introduced by reconstruction-oriented modeling, whereas the challenge of trend anomalies lies in balancing sensitivity to gradual shifts with robustness against normal variations.



(a) F1-score with respect to D



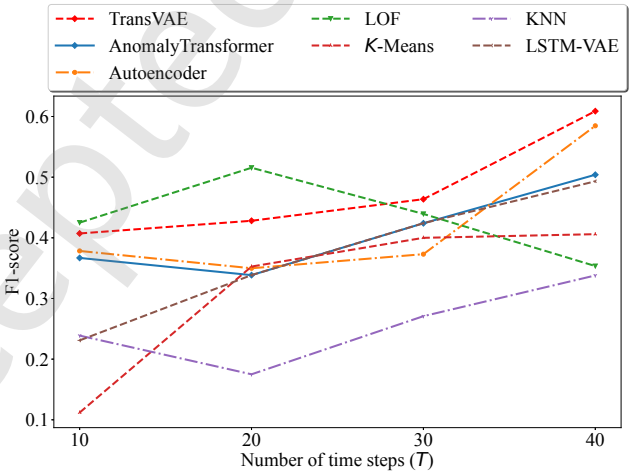
(b) F1-score with respect to τ

Fig. 5 Performance comparison for different hyperparameters

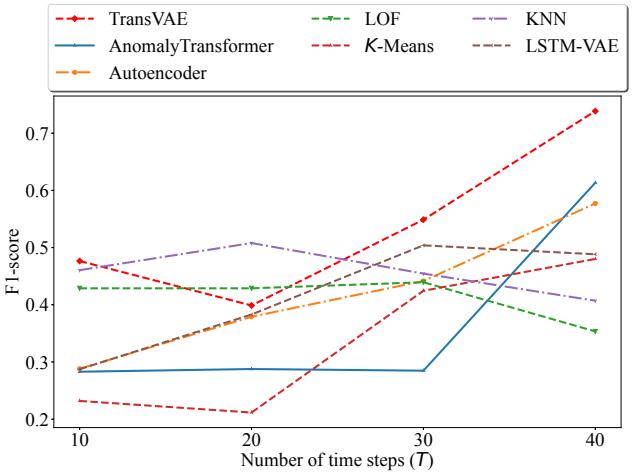
K-Means and random forest algorithms are effective in detecting extreme anomalies, but they do not perform as well as TransVAE and LSTM-VAE in identifying variance-type anomalies, where abnormalities manifest as unusual fluctuation patterns rather than isolated extreme values. This difference is consistent with the advantage of representation-learning methods in modeling nonlinear temporal structure. In particular, TransVAE outperforms both Autoencoder and LSTM-VAE, which is consistent with the combined effect of the local–global representation pipeline and latent regularization described in the framework analysis.

3.6 Model Analysis: Hyperparameters and Components

To comprehensively evaluate TransVAE, we conduct two sets of experiments: (1) a sensitivity analysis evaluating how key hyperparameters affect performance, and (2) an ablation study investigating the contribution of each architectural component.



(a) F1-score performance relative to T for the Wucun area dataset



(b) F1-score performance relative to T for the Gubeikou area dataset

Fig. 6 Performance comparison for different T

3.6.1 Hyperparameter Sensitivity Analysis

In our hyperparameter analysis, we systematically vary the time step count T, latent space dimensionality D, and attention head count τ while keeping other parameters fixed. Fig. 5 presents the F1-scores for different hyperparameters across the Gubeikou and Wucun datasets. Given T=40 for both datasets,

Just Accepted

Fig. 5a illustrates the performance variations for D. It indicates that TransVAE achieves the optimal performance on the Gubeikou dataset when D=64, and on the Wucun dataset when D=32. This difference in the optimal dimensionality reflects the varying complexity of the underlying patterns in each dataset, with DO measurements in Gubeikou potentially exhibiting more complex temporal dependencies that require higher-dimensional representations. Fig. 5b shows that an attention head count of 2 (τ=2) provides the best balance between computational complexity and representational capacity for both datasets, allowing the model to simultaneously attend to multiple temporal patterns at different scales.

Fig. 6 illustrates how the performance of TransVAE varies with adjustments to the number of time steps. As T increases, the F1-score also increases. In anomaly detection tasks involving short-term sequences (T<20), the F1-score of alternative methods is relatively close to that of TransVAE. However, as T increases to 40, TransVAE demonstrates significantly better performance compared to other models. This result supports the advantage of the framework on longer sequences, which is consistent with the use of a Transformer backbone for long-range dependency modeling and the support provided by the global embedding and residual pathways.

3.6.2 Component Analysis

To isolate the effects of each key architectural component, we design ablation experiments comparing TransVAE with its three variants: TransVAE-novae (without VAE component), TransVAE-noembedding (without global embedding module), and TransVAE-nores (without residual network module). Each variant retains all other architectural components of TransVAE.

As shown in Table 3, the component analysis provides an evidence-to-mechanism mapping for the proposed framework. The comparison with TransVAE-noembedding shows that removing the global embedding module weakens performance, supporting the role of the local–global dual-embedding pipeline in supplying both fine-grained and sequence-level context. The comparison with TransVAE-novae shows that removing the VAE degrades performance, supporting the role of latent regularization in shaping the representation of normal patterns and improving the sensitivity of reconstruction-based detection. The comparison with TransVAE-nores shows that removing the residual connection reduces performance, which is consistent with its contribution to representation stability and optimization behavior for long sequences. Together with the time-step and head-number sensitivity analyses, these results show that the empirical gains are aligned with the architectural choices of the framework: the Transformer with multi-head self-attention serves as the temporal modeling backbone, while the dual-embedding design, latent regularization, and residual connection support robust anomaly detection across different temporal scales.

In summary, the results demonstrate that the global embedding, VAE-based latent regularization, and residual network modules all contribute to the effectiveness of TransVAE, while the sensitivity analyses support the framework's behavior on longer sequences and the use of multi-head self-attention as the temporal modeling backbone.

4 Conclusion

This work presents Transformer with Variational AutoEncoder (TransVAE) for anomaly detection in long-term time series. The proposed framework is centered on a local–global dual-embedding representation pipeline and VAE-based latent regularization of Transformer-encoded features, while using a Transformer with multi-head self-attention as the backbone for modeling long-range temporal dependencies. Experimental results on two real-world datasets, together with ablation and sensitivity analyses, support the roles of the global embedding design, latent regularization, and stable long-sequence representation learning in the observed performance gains. These findings indicate that TransVAE provides an effective and robust reconstruction-based solution for long-term time-series anomaly detection.

The future research will focus on two primary areas. First, we plan to utilize patch-based Transformer models to enhance the model's ability to capture nonlinear features. Second, we aim to implement trend decomposition methods to identify trends and intervals characterized by long-range dependence within the time series.

5 Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant 62473014, the National Key Research and Development Program of China under Grant 2025YFE0203700, the Beijing Natural Science Foundation under Grant L233005, and the Fundamental Research Funds for Beijing Municipal Universities under Grant 312000546325001.

References

[1] A. Luo, G. Wen, Z. Chen, and X. Liu. Glasnet: A global–local anomaly generator and salient channel feature selector-based network for accurate few-shot industrial anomaly detection. IEEE Sensors Journal, 25(15):29929–29939, Jun 2025.

[2] Q. Zhao, Y. Wang, B. Wang, J. Lin, S. Yan, W. Song, A. Liotta, J. Yu, S. Gao, and W. Zhang. Msc-ad: A multiscene unsupervised anomaly detection dataset for small defect detection of casting surface. IEEE Transactions on Industrial Informatics, 20(4):6041–6052, Dec 2024.

[3] J. Bi, Z. Wang, H. Yuan, and J. Zhang. Cost-minimized partial computation offloading in cloud-assisted mobile edge computing systems. In 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pages 5052–5057, Oct 2023.

[4] G. Zeng, Y. Yang, K. Lu, G. Geng, and J. Weng. Evolutionary adversarial autoencoder for unsupervised anomaly detection of industrial internet of things. IEEE Transactions on Reliability, 74(3):3454–3468, Jan 2025.

[5] J. Bi, Y. Li, H. Yuan, M. Wang, Z. Wang, J. Zhang, and M. Zhou. Hybrid water quality prediction with multimodal low-rank fusion and localized attention. IEEE Internet of Things Journal, 12(12):21158–21169, Mar 2025.

[6] J. Bi, X. Wu, H. Yuan, Z. Wang, D. Wei, R. Wu, J. Zhang, J. Qiao, and R. Buyya. Stmf: A spatio-temporal multimodal fusion model for long-term water quality forecasting. IEEE Internet of Things Journal, pages 1–1, Jun 2025.

[7] I. Ahmed, A. Dagnino, and Y. Ding. Unsupervised anomaly detection based on minimum spanning tree approximated distance measures

Just Accepted

and its application to hydropower turbines. IEEE Transactions on Automation Science and Engineering, 16(2):654–667, Apr 2019.

[8] J. Lu, R. Lian, D. Jiang, Y. Song, Z. Su, V. J. Wei, and L. Yang. Pretraining enhanced rnn transducer. CAAI Artificial Intelligence Research, 3:9150039, 2024.

[9] J. Bi, Z. Wang, H. Yuan, X. Wu, R. Wu, J. Zhang, and M. Zhou. Long-term water quality prediction with transformer-based spatial-temporal graph fusion. IEEE Transactions on Automation Science and Engineering, 22:11392–11404, Jan 2025.

[10] Z. Li, J. Ma, J. Wu, P. K. Wong, X. Wang, and X. Li. A gated recurrent generative transfer learning network for fault diagnostics considering imbalanced data and variable working conditions. IEEE Transactions on Neural Networks and Learning Systems, 36(8):13782–13793, Feb 2025.

[11] S. Yang, Y. Tian, C. He, X. Zhang, K. C. Tan, and Y. Jin. A gradient-guided evolutionary approach to training deep neural networks. IEEE Transactions on Neural Networks and Learning Systems, 33(9):4861–4875, Sept 2022.

[12] Y. Lin, J. Qiao, J. Bi, H. Yuan, M. Wang, J. Zhang, and M. Zhou. Transformer-based water quality forecasting with dual patch and trend decomposition. IEEE Internet of Things Journal, 12(8):10987–10997, Apr 2025.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30, 2017.

[14] H. He, Q. Zhang, K. Yi, K. Shi, Z. Niu, and L. Cao. Distributional drift adaptation with temporal conditional variational autoencoder for multivariate time series forecasting. IEEE Transactions on Neural Networks and Learning Systems, 36(4):7287–7301, Apr 2025.

[15] L. Kong, G. Li, W. Rafique, S. Shen, Q. He, M. R. Khosravi, R. Wang, and L. Qi. Time-aware missing healthcare data prediction based on arima model. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 21(4):1042–1050, Jul-Aug 2024.

[16] W. Ma, M. Ma, Z. Zhang, J. Ma, R. Zhang, and J. Wang. Anomaly detection of mountain photovoltaic power plant based on spectral clustering. IEEE Journal of Photovoltaics, 13(4):621–631, Jul 2023.

[17] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang. Efficient knn classification with different numbers of nearest neighbors. IEEE Transactions on Neural Networks and Learning Systems, 29(5):1774–1785, May 2018.

[18] S. Hariri, M. C. Kind, and R. J. Brunner. Extended isolation forest. IEEE Transactions on Knowledge and Data Engineering, 33(4):1479–1489, Apr 2021.

[19] L. Chen, W. Wang, and Y. Yang. Celof: Effective and fast memory efficient local outlier detection in high-dimensional data streams. Applied Soft Computing, 102:107079, 2021.

[20] Y. Qiao, K. Wu, and P. Jin. Efficient anomaly detection for high-dimensional sensing data with one-class support vector machine. IEEE Transactions on Knowledge and Data Engineering, 35(1):404–417, Jan 2023.

[21] J. Qu, Q. Du, Y. Li, L. Tian, and H. Xia. Anomaly detection in hyperspectral imagery based on gaussian mixture model. IEEE Transactions on Geoscience and Remote Sensing, 59(11):9504–9517, Nov 2021.

[22] J. Fan, Z. Liu, H. Wu, J. Wu, Z. Si, P. Hao, and T. H. Luan. Luad: A lightweight unsupervised anomaly detection scheme for multivariate time series data. Neurocomputing, 557:126644, 2023.

[23] Y. Xu, Y. Lu, C. Ji, and Q. Zhang. Adaptive graph fusion convolutional recurrent network for traffic forecasting. IEEE Internet of Things Journal, 10(13):11465–11475, Jul 2023.

[24] J. Bi, Y. Lin, Q. Dong, H. Yuan, and M. Zhou. Large-scale water quality prediction with integrated deep neural network. Information Sciences, 571:191–205, 2021.

[25] J. Bi, L. Zhang, H. Yuan, and J. Zhang. Multi-indicator water quality prediction with attention-assisted bidirectional lstm and encoder-decoder. Information Sciences, 625:65–80, 2023.

[26] L. Ren, J. Dong, X. Wang, Z. Meng, L. Zhao, and M. J. Deen. A data-driven auto-cnn-lstm prediction model for lithium-ion battery remaining useful life. IEEE Transactions on Industrial Informatics, 17(5):3478–3487, May 2021.

[27] C. Wang, T. Li, D. Xin, Q. Wang, R. Chen, and C. Cao. Pan evaporation prediction using lstm models based on pca factor reduction and firefly optimization algorithm. IEEE Journal on Miniaturization for Air and Space Systems, 4(4):416–422, Dec 2023.

[28] X. Xia, X. Pan, N. Li, X. He, L. Ma, X. Zhang, and N. Ding. Gan-based anomaly detection: A review. Neurocomputing, 493:497–535, 2022.

[29] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S. Ng. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In International Conference on Artificial Neural Networks, pages 703–716, 2019.

[30] R. Yang, D. Wang, and J. Qiao. Policy gradient adaptive critic design with dynamic prioritized experience replay for wastewater treatment process control. IEEE Transactions on Industrial Informatics, 18(5):3150–3158, May 2022.

[31] A. Terbuch, P. O'Leary, N. Khalili-Motlagh-Kasmaei, P. Auer, A. Zöhrer, and V. Winter. Detecting anomalous multivariate time-series via hybrid machine learning. IEEE Transactions on Instrumentation and Measurement, 72:1–11, 2023.

[32] Z. Chen, M. Ma, T. Li, H. Wang, and C. Li. Long sequence time-series forecasting with deep learning: A survey. Information Fusion, 97:101819, 2023.

[33] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting, 37(4):1748–1764, 2021.

[34] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI conference on artificial intelligence, volume 35, pages 11106–11115, 2021.

[35] Y. Bu, S. Zou, Y. Liang, and V. V. Veeravalli. Estimation of kl divergence: Optimal minimax rate. IEEE Transactions on Information Theory, 64(4):2648–2674, Apr 2018.

[36] H. Wu, J. Xu, J. Wang, and M. Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In Advances in Neural Information Processing Systems, volume 34, pages 22419–22430, 2021.

[37] W. Sakong, J. Kwon, K. Min, S. Wang, and W. Kim. Anomaly transformer ensemble model for cloud data anomaly detection. IEEE Transactions on Cloud Computing, 12(4):1305–1313, Oct-Dec 2024.

[38] L. Chen, Z. You, N. Zhang, J. Xi, and X. Le. Utrad: Anomaly detection and localization with u-transformer. Neural Networks, 147:53–62, 2022.

[39] K. Eckle and J. Schmidt-Hieber. A comparison of deep networks with relu activation function and linear spline-type methods. Neural Networks, 110:232–242, 2019.

[40] B. Rodrawangpai and W. Daungjaiboon. Improving text classification with transformers and layer normalization. Machine Learning With Applications, 10:100403, 2022.

[41] M. Abdolahnejad and P. X. Liu. A deep autoencoder with novel adaptive resolution reconstruction loss for disentanglement of concepts in face images. IEEE Transactions on Instrumentation and Measurement, 71:1–13, 2022.

Just Accepted

[42] Q. Wang, Y. Ma, K. Zhao, and Y. Tian. A comprehensive survey of loss functions in machine learning. Annals of Data Science, pages 1–26, 2020.

[43] A. H. Khan, X. Cao, S. Li, V. N. Katsikis, and L. Liao. Bas-adam: An adam based approach to improve the performance of beetle antennae search optimizer. IEEE/CAA Journal of Automatica Sinica, 7(2):461–471, Mar 2020.

[44] W. Zhang, J. Wang, G. Han, S. Huang, Y. Feng, and L. Shu. A data set accuracy weighted random forest algorithm for iot fault detection based on edge computing and blockchain. IEEE Internet of Things Journal, 8(4):2354–2363, Feb 2021.

[45] X. Allka, P. Ferrer-Cid, J. M. Barcelo-Ordinas, and J. Garcia-Vidal. Leveraging spatiotemporal correlations with recurrent autoencoders for sensor anomaly detection. IEEE Internet of Things Journal, 11(19):31144–31152, Oct 2024.

[46] R. Chen, G. Shi, W. Zhao, and C. Liang. A joint model for it operation series prediction and anomaly detection. Neurocomputing, 448:130–139, Aug 2021.

Just Accepted