MSHTrans: Multi-Scale Hypergraph Transformer with Time-Series Decomposition for Temporal Anomaly Detection

Zhaoliang Chen
Hong Kong Baptist University
Hong Kong SAR, China
chenzl23@outlook.com

Zhihao Wu
Zhejiang University
Hangzhou, China
zhihaowu1999@gmail.com

William K. Cheung
Hong Kong Baptist University
Hong Kong SAR, China
william@comp.hkbu.edu.hk

Hong-Ning Dai
Hong Kong Baptist University
Hong Kong SAR, China
henrydai@comp.hkbu.edu.hk

Byron Choi
Hong Kong Baptist University
Hong Kong SAR, China
bchoi@comp.hkbu.edu.hk

Jiming Liu
Hong Kong Baptist University
Hong Kong SAR, China
jiming@comp.hkbu.edu.hk

Abstract

Time series anomaly detection has garnered significant research attention due to growing demands for temporal data monitoring across diverse domains. Despite the rapid advent of unsupervised anomaly detection models, existing approaches face two critical challenges in understanding the mechanisms of reconstruction-based models when handling diverse temporal dependencies: (1) the insufficient exploration of complex inter-timestamp relationships encompassing both short-term and long-term dependencies, and (2) the lack of integrated frameworks for jointly learning short-term patterns and long-term temporal characteristics. To address these challenges, we propose the novel Multi-Scale Hypergraph Transformer (MSHTrans), which leverages the capacity of hypergraphs for modeling multi-order temporal dependencies. Particularly, our method employs multi-scale downsampling to derive complementary fine-grained and coarse-grained representations, integrated with trainable hypergraph neural networks that can adaptively learn inter-timestamp relationships. The framework further integrates time series decomposition to systematically extract periodic and trend components from multi-granular features, thereby enhancing long-term dependency modeling. Through synergistic integration of learned short-term patterns and long-term temporal structures, the model achieves comprehensive time series reconstruction for effective anomaly detection. Extensive experiments demonstrate that MSHTrans outperforms state-of-the-art competitors with an average performance improvement of 8.21% (without point adjustment) and 3.52% (with point adjustment).

CCS Concepts

• **Computing methodologies** → **Anomaly detection**; • **Mathematics of computing** → **Hypergraphs**; **Time series analysis**.

Keywords

Temporal anomaly detection, hypergraph learning, graph transformer, time-series analysis, multi-scale model.

This work is licensed under a Creative Commons Attribution 4.0 International License.
KDD '25, Toronto, ON, Canada
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737057

ACM Reference Format:
Zhaoliang Chen, Zhihao Wu, William K. Cheung, Hong-Ning Dai, Byron Choi, and Jiming Liu. 2025. MSHTrans: Multi-Scale Hypergraph Transformer with Time-Series Decomposition for Temporal Anomaly Detection. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737057

KDD Availability Link:
The source code of this paper has been made publicly available at https://doi.org/10.5281/zenodo.15502925.

1 Introduction

Time series anomaly detection serves as a fundamental task across various real-world domains, particularly in industrial monitoring systems, healthcare diagnostics, and financial risk management [4, 6, 19]. With the exponential growth of temporal data, modern time series exhibit three inherent characteristics: massive scale, heterogeneous sources, and multimodal patterns, which collectively pose challenges to time series anomaly detection. To address the complexity of large-scale time series data, existing methods predominantly employ unsupervised learning approaches that extract latent patterns from sequential data, enabling efficient anomaly detection [2, 39]. Existing time series anomaly detection models can be roughly divided into two categories: prediction-based methods and reconstruction-based methods. Although *prediction-based* approaches utilize historical observations to generate future estimations, with anomalies detected through quantitative analysis of prediction errors, they exhibit limited adaptability to rapid temporal pattern changes [36, 41]. *Reconstruction-based* approaches have received increasing attention due to their strengths in detecting anomalies by comparing series reconstructed from learned latent representations with original inputs.

Both prediction-based and reconstruction-based approaches require effective modeling of long-term and short-term temporal dependencies under the non-independence assumption of timestamps. Given the implicit nature of temporal correlations, recent studies have adopted Graph Neural Networks (GNNs) [16, 17, 32] to construct similarity-based relationships between timestamps. While

graph-based approaches have demonstrated effectiveness in forecasting tasks, their application to reconstruction-based anomaly detection remains under-explored. Despite some recent studies considering feature-oriented or time-oriented graphs for reconstruction-based models [14, 40], these models generally construct *fixed* graphs based on pair-wise or group-wise similarities, thereby hindering GNN-based models from dynamically learning adjacency relationships in a data-driven manner during the training process. Moreover, due to the highly complex temporal relationships in time series data, a simple adjacency relationship is insufficient to comprehensively describe the compound long-term and short-term correlations between timestamps. In light of this issue, this paper proposes using *trainable* hypergraphs to model the intrinsic temporal relationships. Hypergraphs enable high-order interaction modeling through flexible degree-free hyperedges that connect multiple timestamps simultaneously, surpassing the representational constraints of ordinary graphs. Further, in practical time series applications, sensor measurements at each timestamp often manifest a complex interaction between short-term fluctuations and long-term trends. This multi-scale temporal dependency presents a fundamental limitation for conventional fixed graph-based approaches, as their inherent constraints in topological rigidity prevent effective modeling of temporal patterns across different granularities. Therefore, *how to construct trainable long-term and short-term timestamp correlations with hypergraphs* becomes a critical challenge.

In time series data reconstruction, short-term relationships help the model propagate local key features from nearby timestamps, while long-term relationships provide support for sequence reconstruction from a global perspective. Theoretically, coarse-grained long-term information can assist the model in learning the overall periodicity and trend characteristics of the time series, which also mitigates the impact of local time series noises or fluctuations. Apart from the global information brought by long-distance associations in hypergraphs, decomposing time series can also introduce seasonal and trend signals for global coarse-grained model reconstruction. Seasonality and trends can reveal the periodic patterns of time series at a certain scale. The proposed model is architected with a dual-phase reconstruction mechanism: (1) Coarse-grained reconstruction leveraging hypergraph-derived long-term temporal dependencies as well as the inherent seasonality and trends of the series; (2) Fine-grained reconstruction through hypergraph-derived short-term temporal dependencies, enabling localized refinement through residual pattern analysis. To this end, *how to effectively integrate and synergistically utilize both short-term temporal variations and long-term temporal dynamics for multi-scale time series reconstruction*, and *how such hierarchical reconstruction mechanisms can intrinsically improve anomaly detection performance*, are also challenges that this paper seeks to tackle.

Solutions: To address the aforementioned challenges, we propose a novel multivariate time series anomaly detection model dubbed Multi-Scale Hypergraph Transformer (MSHTrans), which can effectively reconstruct time series from hypergraph-oriented correlations and periodical signals. MSHTrans reorganizes the input window as multi-scale inputs through downsampling, thereby simultaneously exploring coarse-grained and fine-grained timestamp correlations via multi-channel encoders. The proposed model

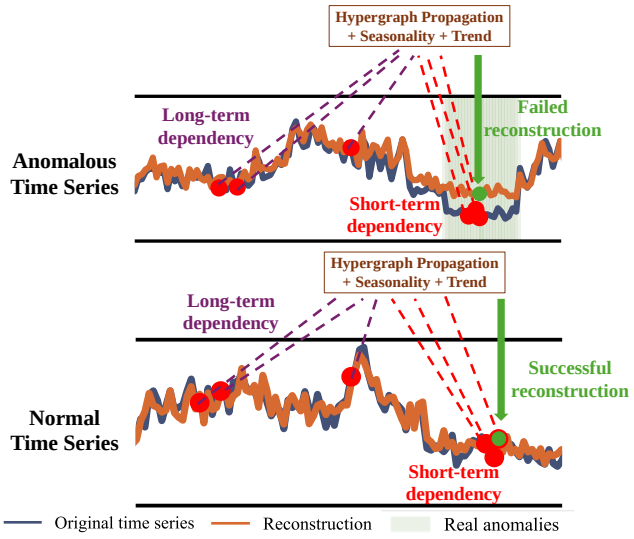

Figure 1: Reconstruction of time series with long-term and short-term dependencies for the proposed MSHTrans.

can jointly learn both short-term and long-term correlations between timestamps by trainable hypergraphs and time series decomposition analysis. Figure 1 illustrates the mechanism of the reconstruction-based MSHTrans model for anomaly detection. As shown in the figure, MSHTrans is able to conduct series reconstruction with trainable hypergraph propagation, seasonality learning, and trend learning. Generally, a failed sequence reconstruction (the 1st row in Figure 1) only indicates potential anomalies, whereas a successful sequence reconstruction (the 2nd row in Figure 1) suggests that the current temporal variables can be well restored by learning correlations between timestamps and other global implicit information. When the time series exhibit normal temporal patterns, the model first utilizes the long-term dependencies learned through hypergraph representations, combined with seasonality and trend components, to perform coarse-grained reconstruction. Subsequently, the hypergraph-encoded short-term dependencies are systematically integrated to achieve fine-grained reconstruction refinement. When an anomaly happens in the time series, the anomalous data will be propagated to the target timestamp through the learned short-term relationships of hypergraphs, disrupting the intermediate feature representation. This mechanism leads to the failure of fine-grained sequence reconstruction because the model can only conduct coarse-grained reconstruction with long-term dependencies, global seasonality, and trend components. We further elaborate on this mechanism with experiments in Section 5.4.2. In summary, the main contributions of this paper include:

- To capture the relationships between different timestamps within the window, we propose a trainable hypergraph learning schema, which adaptively explores long-term and short-term dependencies among time series to reconstruct the variables.
- To extract the seasonal and trend information embedded within the latent features, we construct a time series decomposition module and a feature fusion module, which aim

to capture and integrate the periodic information and the variable trends from the latent features.
- To leverage both short-term and long-term signals, we propose a transformer-based framework for time series anomaly detection, which comprises multiple hypergraph learning networks, time series decomposition modules, and signal fusion processes. The framework aims to uncover the hidden correlations and intrinsic periodicity among timestamps, thereby facilitating efficient reconstruction.
- We demonstrate the superiority of MSHTrans over state-of-the-art methods through comprehensive experiments. We also explore how the proposed hypergraph-based model detects anomalies through series reconstruction.

2 Related Work

2.1 Time Series Anomaly Detection

2.1.1 Prediction-based Models. The core concept behind prediction-based anomaly detection models is to build a predictive model to estimate the future values of time series data and then evaluate the deviations between predicted values and the actual observations. A large number of predictive techniques have been utilized for this purpose, such as Convolutional Neural Networks (CNN) [24, 33], Recurrent Neural Networks (RNN) [9, 26] and transformers [8, 30]. For instance, LSTM networks [15, 23] have been employed to predict the future values of time series data, with the prediction loss used as the anomaly score. CNN was combined with a spectral residual model to detect visual anomalies [24]. However, prediction-based models generally struggle with rapidly and continuously changing time series, thereby limiting their ability to forecast short-term series and overcome local noises.

2.1.2 Reconstruction-based Models. Reconstruction-based models aim to learn the underlying features of time series to reconstruct the input time series, and then evaluate the reconstruction quality. Most reconstruction-based models have similar structures to autoencoders [1, 3, 11, 18] or transformers [4, 20, 27, 37]. As a typical example, the Deep Autoencoding Gaussian Mixture Model (DAGMM) [43] predicted the probability of series samples through the Gaussian mixture prior to the latent space. TranAD [29] incorporated adversarial training with transformers to conduct 2-phase time series reconstruction. Belonging to reconstruction-based models, the proposed MSHTrans recovers time series with trainable hypergraph relationships and series analysis.

2.2 Hypergraph Neural Networks

Graph neural network (GNN) can model the relationships between different objects and has shown encouraging performance in various tasks [10, 34, 42]. Recent research efforts have been made to the investigation of time series anomaly detection with GNNs. As a successful attempt of GNN-based methods, MTAD-GAT [40] utilized Graph Attention Network (GAT) to construct both feature-level and time-level relationships, which considered both prediction-based losses and reconstruction-based losses to estimate anomalies. Due to its ability to model complex node relationships, Hypergraph Neural Network (HGNN) [12, 13] has become a research hotspot in recent years. Although some recent studies have adopted HGNNs to node-level, graph-level, or link-level anomaly detection [21, 22], the application of HGNNs in time-series anomaly detection is still under-explored. Two recent frameworks [25, 35] also attempted to utilize multi-scale hypergraphs in time series prediction tasks by building group-wise correlations between timestamps. However, *the mechanism of hypergraph learning under the reconstruction model in time series anomaly detection tasks has not yet been fully explored.* These HGNN models also lack effective supervision of hyperedge learning or seldom consider global characteristics like seasonality and trend signals. Therefore, this paper proposes a novel framework to jointly investigate supervised hypergraph construction methodologies while systematically integrating the resultant dependency relationships with global temporal characteristics.

3 Preliminary

The multivariate time-series data is defined by $\mathbf{X} \in \mathbb{R}^{T\times D}$, where T is the length of the time series and D is the number of variables. In order to capture the trend of time series from historical data, we can generate contextual window $\mathbf{W}^{(t)} \in \mathbb{R}^{S\times D}$ at the t-th time point with $\mathbf{W}^{(t)} = [\mathbf{X}^{(t-S+1)}; \cdots; \mathbf{X}^{(t)}]$ for $t \geq S$, where S is the predefined maximum window size. The proposed model aims to reconstruct input window $\mathbf{W}$, based on which generates the anomaly scores $\mathcal{S}$. The anomaly detection indicator $\mathcal{Y} = \{\mathcal{Y}_1, \cdots, \mathcal{Y}_T\}$ is computed with the anomaly scores and threshold, where $\mathcal{Y}_t \in \{0, 1\}$ and $\mathcal{Y}_t = 1$ indicates the anomalous timestamp.

4 The Proposed Model

4.1 Overview

The proposed model is illustrated in Figure 2. As a multi-scale model based on an encoder-decoder architecture, MSHTrans adopts multi-scale encoders, which are responsible for encoding the input data with trainable hypergraphs. The learnable seasonal features and trend information at different scales are aggregated eventually. Moreover, MSHTrans leverages the decoder to reconstruct the original time series based on the hypergraph relationships and underlying features learned by encoders. Finally, the model calculates anomaly scores for each time point based on the reconstruction quality of the decoder, thereby predicting anomalous timestamps. We next elaborate on the detailed design of the proposed MSHTrans framework as follows.

4.2 Multi-Scale Window Generator

To capture latent features of L different granularities, a multi-scale window generator is proposed to construct window $\mathbf{W}_{(s)}^{(t)} \in \mathbb{R}^{N_s\times D}$ (simplified as $\mathbf{W}_{(s)}$) at scale s. With $\mathbf{W}_{(0)}^{\text{conv}} = \mathbf{W}_{(0)}^{\text{sample}} = \mathbf{W}$, the generator is defined as

$$\mathbf{W}_{(s)}^{\text{conv}} = \text{Conv1D}\left(\mathbf{W}_{(s-1)}^{\text{conv}} | \Theta_{(s-1)}, \mathcal{K}\right), \tag{1}$$

$$\mathbf{W}_{(s)}^{\text{sample}} = \text{DownSample}\left(\mathbf{W}_{(s-1)}^{\text{sample}} | \mathcal{K}\right), \tag{2}$$

$$\mathbf{W}_{(s)} = [\mathbf{W}_{(s)}^{\text{conv}}, \mathbf{W}_{(s)}^{\text{sample}}], \tag{3}$$

where $[\cdot,\cdot]$ is the concatenation operation, Conv1D$(\cdot)$ is the 1-D convolution operator parameterized by trainable $\Theta_{(s-1)}$ and $\mathcal{K}$ is the kernel size. DownSample$(\cdot)$ indicates the downsampling

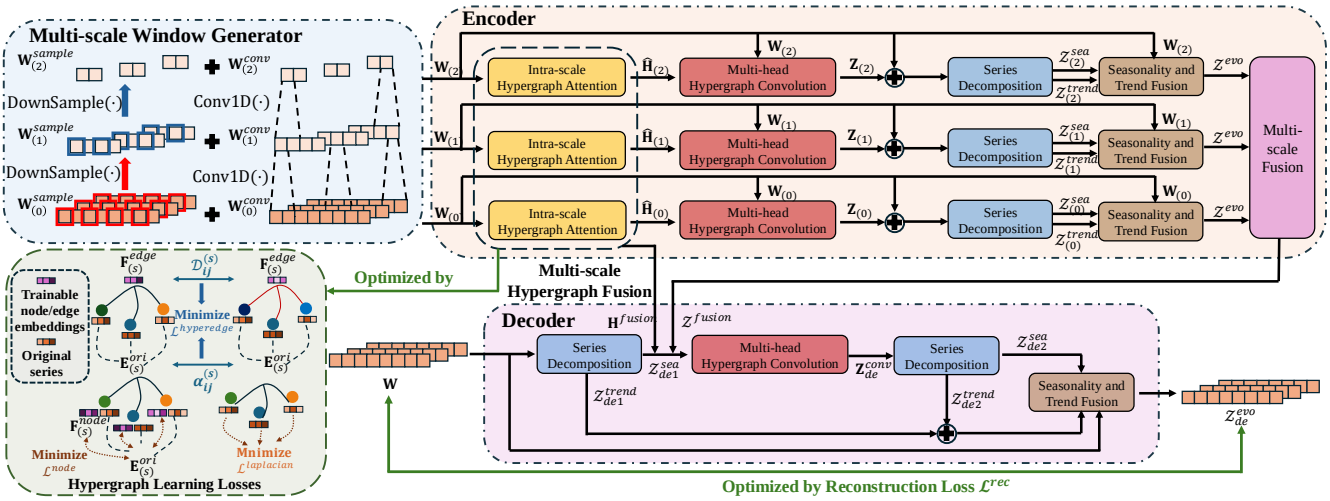

Figure 2: The framework of the proposed MSHTrans.

operation that directly selects the sequence data from scale $s-1$ at intervals of $\mathcal{K}$. Thus, we have $N_s = \lfloor \frac{N_{s-1}}{\mathcal{K}} \rfloor$. Specifically, the maximum size $N_0 = S$, i.e., the window size. Eqs. (1), (2) and (3) indicate that window at the higher scale level is calculated from a lower scale level. The point-wise relationships at scale s correspond to the group-wise relationships at scale $s-1$ when $s \geq 1$. When $s = 0$, the point-wise relationships indicate the fine-grained connections between each timestamp. Therefore, the proposed MSHTrans can explore fine-grained time relations with low-scale windows and coarse-grained time relations with high-scale windows.

4.3 Hypergraph Transformer

To capture the complicated short-term and long-term relations between different timestamps, we build a hypergraph transformer for a hypergraph $\mathcal{G}_{(s)} = \{\mathcal{V}_{(s)}, \mathcal{E}_{(s)}\}$ at scale s. Herein, $\mathcal{V}_{(s)} = \{v_1, \cdots, v_{N_s}\}$ is the node set, which corresponds to different timestamps within the window at scale s, $\mathcal{E}_{(s)} = \{e_1, \cdots, e_{M_s}\}$ indicates the hyperedge set containing rich connections among nodes. The topology of the hypergraph at scale s is denoted by an incidence matrix $\mathbf{H}_{(s)} \in \mathbb{R}^{N_s \times M_s}$, which is defined by

$$[\mathbf{H}_{(s)}]_{nm} = \begin{cases} 1 & \text{if } v_n \in e_m, \\ 0 & \text{if } v_n \notin e_m. \end{cases} \tag{4}$$

At different timestamps, we hypothesize that the corresponding windows share similar long-term and short-term dependencies across temporal nodes. Consequently, we construct a window-level hypergraph within the same scale, thereby enabling the hypergraph to learn generalized temporal dependencies that maintain invariance across varying temporal windows.

4.3.1 Trainable Hypergraphs. To get a learnable hypergraph describing intra-scale relationships with trainable incidence matrix, we compute the similarity-based $\mathbf{H}_{(s)}$ at scale s by

$$\mathbf{H}_{(s)} = \sigma(\text{ReLU}(\mathbf{F}^{\text{node}}_{(s)}(\mathbf{F}^{\text{edge}}_{(s)})^T)), \tag{5}$$

where $\mathbf{F}^{\text{node}}_{(s)} \in \mathbb{R}^{N_s \times d}$ and $\mathbf{F}^{\text{edge}}_{(s)} \in \mathbb{R}^{M_s \times d}$ are trainable node and hyperedge embeddings, respectively. Herein, $\sigma(\cdot)$ is the activation function, $\text{ReLU}(\cdot)$ is the rectified linear unit to keep the non-negativity and sparsity of embeddings.

At each epoch, the hypergraph incidence matrix is updated with an updating rate τ. Namely, $\mathbf{H}_{(s)} = (1-\tau)\mathbf{H}_{(s)} + \tau\tilde{\mathbf{H}}_{(s)}$, where $\tilde{\mathbf{H}}_{(s)}$ is the hypergraph incidence matrix at the last epoch. Specifically, the initial incidence matrix at scale s is generated by

$$[\mathbf{H}_{(s)}]_{nm} = \begin{cases} 1 & \text{if } N_s - M_s - k + m \leq n \leq N_s - M_s + m, \\ 0 & \text{otherwise}, \end{cases} \tag{6}$$

where k is the predefined hyperparameter to determine each hyperedge's initial connected nodes. Eq. (6) indicates that the initial hypergraph tends to construct adjacency relationships over short-term timestamps; this design agrees with the intuition of time series reconstruction. Namely, the value at the current timestamp is highly correlated with recent data. Moreover, the proposed model updates the incidence matrix during training to discover either unknown long-term or other short-term dependencies.

With a sparse $\mathbf{H}_{(s)}$, we further select the top-k significant connections in $\mathbf{H}_{(s)}$ to limit the maximum number of connected nodes and reduce computation cost, i.e.,

$$[\mathbf{H}_{(s)}]_{nm} = \begin{cases} 1 & \text{if } [\mathbf{H}_{(s)}]_{nm} \in \text{TopK}([\mathbf{H}_{(s)}]_n), \\ 0 & \text{if } [\mathbf{H}_{(s)}]_{nm} \notin \text{TopK}([\mathbf{H}_{(s)}]_n), \end{cases} \tag{7}$$

where $\text{TopK}(\cdot)$ is the top-k selection operator.

4.3.2 Intra-scale Hypergraph Attention. First, we denote the resampled variables of the n-th node of training window by $[\mathbf{W}_{(s)}]_n$. To realize the intra-scale hypergraph transformer module, we first adopt the graph attention mechanism [25, 31] to aggregate the embeddings at each node with hypergraph at scale s, defined as

$$[\hat{\mathbf{H}}_{(s)}]_{nm} = \frac{\exp(\sigma([[\mathbf{W}_{(s)}]_n, [\mathbf{E}_{(s)}]_m]\Theta^{Att}_{(s)}))}{\sum_{\mathcal{E}^k_{(s)} \in \mathcal{N}(\mathcal{V}^n_{(s)})} \exp(\sigma([[\mathbf{W}_{(s)}]_n, [\mathbf{E}_{(s)}]_k]\Theta^{Att}_{(s)}))}, \tag{8}$$

Algorithm 1: Intra-scale Hypergraph Attention Module IntraHAtt $\left(\mathbf{W}_{(s)}, \mathcal{G}_{(s)}\right)$

Input : Resampled window data $\mathbf{W}_{(s)}$ and trainable hypergraph $\mathcal{G}_{(s)}$.

Initialize the similarity-based hypergraph incidence matrix $\mathbf{H}_{(s)}$ with Eq. (5);

Generate sparse hypergraph incidence matrix $\mathbf{H}_{(s)}$ via top-k selection with Eq. (7);

Get $\hat{\mathbf{H}}_{(s)}$ with hypergraph attention defined by Eq. (8);

Return Hypergraph incidence matrix $\hat{\mathbf{H}}_{(s)}$.

Algorithm 2: Multi-head Hypergraph Convolution Module MHConv $\left(\hat{\mathbf{H}}_{(s)}, \mathbf{W}_{(s)}\right)$

Input : Window data $\mathbf{W}_{(s)}$, sparse hypergraph incidence matrix $\hat{\mathbf{H}}_{(s)}$ and head number H.

for $h = 1 \rightarrow H$ **do**

  Conduct hypergraph convolution with Eq. (11);

Conduct multi-head hypergraph convolution fusion with Eq. (10) to get $\mathbf{Z}_{(s)}$;

Return Multi-head hypergraph convolution result $\mathbf{Z}_{(s)}$.

where $\Theta_{(s)}^{Att}$ is the trainable weight, and $\mathcal{N}(\mathcal{V}_{(s)}^{n})$ indicates the neighborhood hyperedges connecting node n according to hypergraph $\mathcal{G}_{(s)}$. $[\mathbf{E}_{(s)}]_m$ is the edge features computed by the aggregation of connected node variables obtained by Eq. (3), i.e.,

$$[\mathbf{E}_{(s)}]_m = \mathrm{Agg}\left(\sum_{\mathcal{V}_{(s)}^{k} \in \mathcal{N}(\mathcal{E}_{(s)}^{m})} [\mathbf{W}_{(s)}]_k\right), \tag{9}$$

where $\mathcal{V}_{(s)}^{k} \in \mathcal{N}(\mathcal{E}_{(s)}^{m})$ represents the variables of timestamps connected by edge m. We denote the intra-scale hypergraph attention module by $\hat{\mathbf{H}}_{(s)} = \mathrm{IntraHAtt}\left(\mathbf{W}_{(s)}, \mathcal{G}_{(s)}\right)$. Algorithm 1 depicts its construction process.

4.3.3 Multi-Head Hypergraph Convolutions. With $\hat{\mathbf{H}}_{(s)}$, we can conduct the multi-head hypergraph convolution [12] with

$$\mathbf{Z}_{(s)} = [\mathbf{Z}_{(s)}^{1}, \cdots, \mathbf{Z}_{(s)}^{H}], \tag{10}$$

where H is the number of heads and the h-th head's output is

$$\mathbf{Z}_{(s)}^{h} = \sigma\left((\mathbf{D}_{(s)}^{v})^{-\frac{1}{2}} \hat{\mathbf{H}}_{(s)}^{h} \Phi_{(s)}^{h} (\mathbf{D}_{(s)}^{e})^{-1} (\hat{\mathbf{H}}_{(s)}^{h})^{T} (\mathbf{D}_{(s)}^{v})^{-\frac{1}{2}} \mathbf{W}_{(s)} \Theta_{(s)}^{h}\right), \tag{11}$$

where $\Phi_{(s)}^{h} = \mathrm{diag}(\phi_1^h, \cdots, \phi_{N_s}^h)$ is the trainable weight for hyperedges, and $\Theta_{(s)}^{h}$ is the trainable weight to capture underlying features. $\mathbf{D}_{(s)}^{v} \in \mathbb{R}^{N_s \times N_s}$ and $\mathbf{D}_{(s)}^{e} \in \mathbb{R}^{M_s \times M_s}$ are diagonal node degree and hyperedge degree matrices, where $[\mathbf{D}_{(s)}^{v}]_{nn} = \sum_{m=1}^{M_s} [\mathbf{H}_{(s)}]_{nm}$ and $[\mathbf{D}_{(s)}^{e}]_{mm} = \sum_{n=1}^{N_s} [\mathbf{H}_{(s)}]_{nm}$. The multi-head hypergraph convolution module is denoted by $\mathbf{Z} = \mathrm{MHConv}\left(\hat{\mathbf{H}}_{(s)}, \mathbf{W}_{(s)}\right)$. Algorithm 2 depicts its construction process.

4.3.4 Hypergraph Learning Constraints. Next, we elaborate on the hypergraph losses applied in MSHTrans, as described in Figure 2.

Inter-scale Hypergraph Consistency Constraint. First, the relationships among timestamps, i.e., the adjacency matrix $\mathbf{A}_{(s)} \in \mathbb{R}^{N_s \times N_s}$ of timestamps at different scales can be obtained by $\mathbf{A}_{(s)} = \mathbf{H}_{(s)} \mathbf{H}_{(s)}^{T}$. Herein, we hypothesize that the learned hypergraphs should follow Laplacian constraints, ensuring cross-scale feature consistency among timestamp nodes connected through hyperedges. To learn such consistency, we adopt the Laplacian constraint on all adjacency matrices to minimize the similarities between connected nodes. Namely, we have

$$\begin{aligned} &\min_{\mathbf{A}_{(0)}, \cdots, \mathbf{A}_{(L-1)}} \mathcal{L}_{(s)}^{\mathrm{Laplacian}}\left(\mathbf{W}_{(s)}^{\mathrm{sample}}, \mathbf{L}_{(s)}\right) \\ &\qquad = \sum_{s=0}^{L-1} \mathrm{Tr}\left((\mathbf{W}_{(s)}^{\mathrm{sample}})^{T} \mathbf{L}_{(s)} \mathbf{W}_{(s)}^{\mathrm{sample}}\right), \end{aligned} \tag{12}$$

where $\mathbf{L}_{(s)}$ is the Laplacian matrix computed by $\mathbf{L}_{(s)} = \mathbf{D}_{(s)} - \mathbf{A}_{(s)}$, and $\mathbf{D}_{(s)} = \mathrm{Diag}(\mathbf{A}_{(s)})$ is the diagonal matrix. Since $\{\mathbf{W}_{(s)}^{\mathrm{sample}}\}_{s=1}^{L-1}$ are downsampled from the same window $\mathbf{W}$, the minimization problem in Eq. (12) attempts to learn the consistency of multi-scale hypergraphs within the window.

Similarity-based Hypergraph Constraints: We hope that connected nodes of the learned hyperedge are as similar as possible. Hence, we calculate the similarity-based weights at each scale, i.e.,

$$\alpha_{ij}^{(s)} = \frac{\left[\mathbf{E}_{(s)}^{\mathrm{ori}}\right]_i \left[\mathbf{E}_{(s)}^{\mathrm{ori}}\right]_j^{T}}{\left\|\left[\mathbf{E}_{(s)}^{\mathrm{ori}}\right]_i\right\|_2 \left\|\left[\mathbf{E}_{(s)}^{\mathrm{ori}}\right]_j\right\|_2}, \tag{13}$$

where $\left[\mathbf{E}_{(s)}^{\mathrm{ori}}\right]_i$ indicates original features of the i-th hyperedge, which is computed from the connected nodes. Namely,

$$\left[\mathbf{E}_{(s)}^{\mathrm{ori}}\right]_i = \mathrm{Agg}\left(\sum_{\mathcal{V}_{(s)}^{k} \in \mathcal{N}\left(\mathcal{E}_{(s)}^{i}\right)} \left[\mathbf{W}_{(s)}^{\mathrm{sample}}\right]_k\right). \tag{14}$$

Since we hope to minimize the deviations between learned edge embeddings and original edge features, we define the hyperedge constraint as:

$$\begin{aligned} &\min_{\mathbf{F}_{(s)}^{\mathrm{edge}}} \mathcal{L}_{(s)}^{\mathrm{hyperedge}}\left(\mathbf{F}_{(s)}^{\mathrm{edge}}, \mathbf{E}_{(s)}^{\mathrm{ori}}\right) \\ &= \frac{1}{(M_s)^2} \sum_{i=1}^{M_s} \sum_{j=1}^{M_s} \left(\alpha_{ij}^{(s)} \mathcal{D}_{ij} + (1 - \alpha_{ij}^{(s)}) \max(\gamma - \mathcal{D}_{ij}, 0)\right), \end{aligned} \tag{15}$$

where γ is the hyperparameter and the distance between learned hyperedge embeddings are calculated as $\mathcal{D}_{ij} = \|[\mathbf{F}_{(s)}^{\mathrm{edge}}]_i - [\mathbf{F}_{(s)}^{\mathrm{edge}}]_j\|_2$. Because a higher $\alpha_{ij}^{(s)}$ indicates a stronger correlation between hyperedges, this objective will minimize the embedding distance $\mathcal{D}_{ij}$ between the learned hyperedges. Conversely, a lower $\alpha_{ij}^{(s)}$ suggests

Algorithm 3: Time Series Decomposition Module TSDecom ($\mathcal{Z}$)

Input : Input series $\mathcal{Z}$.
Conduct DFT operation for $\mathcal{Z}$ with Eq. (17);
Obtain seasonality embeddings $\mathcal{Z}^{\text{sea}}$ with Eq. (18);
Obtain trend embeddings $\mathcal{Z}^{\text{trend}}$ with Eq. (19);
Return Seasonality embeddings $\mathcal{Z}^{\text{sea}}$ and trend embeddings $\mathcal{Z}^{\text{trend}}$.

that the model should learn hyperedge embeddings that have remote distance. In a word, the minimization problem defined in Eq. (15) promotes the consistency between the learned hyperedge embeddings and the original hyperedge features, thereby improving the quality of the hypergraph learned from $\mathbf{F}^{\text{edge}}_{(s)}$.

Meanwhile, we also hope that the learned node embeddings $\mathbf{F}^{\text{node}}_{(s)}$ are consistent with the original hyperedge features. In detail, we define the node constraint as

$$\begin{aligned}&\min_{\mathbf{F}^{\text{node}}_{(s)}} \mathcal{L}^{\text{node}}_{(s)}\left(\mathbf{F}^{\text{node}}_{(s)}, \mathbf{E}^{\text{ori}}_{(s)}\right)\\ =&\frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{\mathcal{V}^i_{(s)}\in\mathcal{N}\left(\mathcal{E}^j_{(s)}\right)} \text{Abs}\left(\text{Proj}\left(\left[\mathbf{F}^{\text{node}}_{(s)}\right]_i\right) - \left[\mathbf{E}^{\text{ori}}_{(s)}\right]_j\right),\end{aligned} \tag{16}$$

where $\text{Proj}(\cdot)$ is the projection function to maintain the consistency of the dimensions of learned node embeddings and original hyperedge features.

4.4 Time Series Decomposition Module

The time series decomposition module aims to factorize the input window to get the seasonality and trend. First, we adopt discern Fourier Transform operator $\text{DFT}(\cdot)$ [5] to transform the input series $\mathcal{Z}$ from the temporal domain into the frequency domain. Namely,

$$\{f_1, \cdots, f_k\}, \mathbf{A}, \Gamma = \text{TopK}(\text{DFT}(\mathcal{Z})), \tag{17}$$

where $\{f_1, \cdots, f_k\}$ are the top-k significant frequencies, $\mathbf{A}$ is the amplitude, and Γ is the phase.

With these top-k frequencies, the periodic signals $\mathcal{Z}^{\text{sea}}$ can be calculated by the inverse DFT operator $\text{IDFT}(\cdot)$, i.e.,

$$\mathcal{Z}^{\text{sea}} = \text{IDFT}\left(\{f_1, \cdots, f_k\}, \mathbf{A}, \Gamma\right). \tag{18}$$

As for the trend of the window, we utilize K kernels of average pooling for moving averages to capture trend patterns. Namely,

$$\mathcal{Z}^{\text{trend}} = \sum_{i=1}^{K} \alpha_i \text{AvgPool}_i(\mathcal{Z}), \tag{19}$$

where α_i is the weight for the i-th kernel and $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_K]$ is normalized by the Softmax function. In summary, the whole time series decomposition module is denoted by $\mathcal{Z}^{\text{sea}}, \mathcal{Z}^{\text{trend}} = \text{TSDecom}(\mathcal{Z})$, as elaborated in Algorithm 3.

With the seasonality and trend series, we can obtain an evolutionary time-series by

$$\mathcal{Z}^{\text{evo}} = \text{FeedForward}([\mathbf{W}, \mathcal{Z}^{\text{sea}}\Theta^{\text{evo}}, \mathcal{Z}^{\text{trend}}]), \tag{20}$$

Algorithm 4: Seasonality and Trend Fusion Module $\text{STFusion}\left(\mathcal{Z}, \mathcal{Z}^{\text{sea}}, \mathcal{Z}^{\text{trend}}\right)$

Input : Complete window embeddings $\mathcal{Z}$, seasonality embeddings $\mathcal{Z}^{\text{sea}}$ and trend embeddings $\mathcal{Z}^{\text{trend}}$.
Initialize trainable weights Θ^{evo};
Compute evolutionary embeddings with Eq. (20);
Return Evolutionary series features $\mathcal{Z}^{\text{evo}}$.

where Θ^{evo} is the trainable weight and $\text{FeedForward}(\cdot)$ is the feedforward layer. We denote $\mathcal{Z}^{\text{evo}} = \text{STFusion}\left(\mathbf{W}, \mathcal{Z}^{\text{sea}}, \mathcal{Z}^{\text{trend}}\right)$ by the seasonality and trend fusion module, described in Algorithm 4.

4.5 Multi-scale Hypergraph Fusion Process

To integrate the learned multi-scale window features from multi-channel encoders, we first upsample the learned $\mathcal{Z}^{\text{evo}}_{(s)}$ at the s scale for $s > 0$, so that outputs from all scales share the same dimension. Thus, we can directly add these upsampled outputs to get a fused representation. In detail, this process can be formulated as

$$\mathcal{Z}^{\text{evo}}_{(s)} = \mathcal{Z}^{\text{evo}}_{(s)} + \text{Padding}\left(\text{ConvTranspose1D}(\mathcal{Z}^{\text{evo}}_{(s+1)})\right), \tag{21}$$

for $1 \le s \le L-1$, where L is the number of scales and the 1-D transpose convolution layer is denoted by $\text{ConvTranspose1D}(\cdot)$. Eventually, we adopt a feedforward layer to the learned features at the initial scale, i.e., $\mathcal{Z}^{\text{fusion}} = \text{FeedForward}(\mathcal{Z}^{\text{evo}}_{(0)})$.

We also need to integrate the learned multi-scale hypergraphs $\left\{\mathbf{H}_{(s)}\right\}_{s=0}^{L-1}$, which will be utilized in the message passing of the decoder. Because the hyperedge connections at a higher scale indicate the group-wise connections at the low scale, we can upsample the hyperedge connections at the high scale via

$$[\tilde{\mathbf{H}}_{(s)}]_n = [\mathbf{H}_{(s+1)}]_{\lfloor n/\mathcal{K} \rfloor}, \tag{22}$$

where $\mathcal{K}$ is the stride for the data sampling and $\tilde{\mathbf{H}}_{(0)} = \mathbf{H}_{(0)}$. Eventually, the multi-scale hyperedge fusion is conducted by

$$\mathbf{H}^{\text{fusion}} = [\tilde{\mathbf{H}}_{(0)}; \tilde{\mathbf{H}}_{(1)}; \cdots; \tilde{\mathbf{H}}_{(L-1)}] \in \mathbb{R}^{N \times (\sum_{s=0}^{L-1} M_s)}. \tag{23}$$

Thus, $\mathbf{H}^{\text{fusion}}$ includes both point-wise and group-wise interaction information within the window.

4.6 Model Training

In summary, MSHTrans is a multi-channel encoder-decoder structure to reconstruct window data from multi-scale inputs, as shown in Figure 2. In MSHTrans, the short-term dependencies are discovered by the trainable hypergraphs, and the long-term dependencies are captured by both hyperedge connections and time series decomposition analysis. With the aforementioned modules, **Appendix A.1** depicts the detailed training algorithm of MSHTrans. The training loss consists of Laplacian loss, hyperedge loss, node loss, and reconstruction loss, i.e.,

$$\mathcal{L}^{\text{all}} = \sum_{s=0}^{L-1}\left(\mathcal{L}^{\text{Laplacian}}_{(s)} + \mathcal{L}^{\text{hyperedge}}_{(s)} + \mathcal{L}^{\text{node}}_{(s)}\right) + \mathcal{L}^{\text{rec}}, \tag{24}$$

where $\mathcal{L}^{\text{rec}}$ is the reconstruction loss computed from the deviations between the reconstructed window $\mathcal{Z}^{\text{evo}}_{\text{de}}$ in the decoder and the

Table 1: Experimental results (F1 scores) of compared models, where the best performance is highlighted in orange and the second-best performance is highlighted in blue.

Datasets	SWaT		WADI		MSL		SMAP		SMD		Average	
Evaluation strategy	w/o PA	with PA	w/o PA	with PA	w/o PA	with PA	w/o PA	with PA	w/o PA	with PA	w/o PA	with PA
DAGMM [43]	0.750	0.853	0.121	0.209	0.199	0.701	0.233	0.712	0.238	0.723	0.308	0.640
LSTM-VAE [23]	0.705	0.805	0.227	0.380	0.212	0.854	0.235	0.756	0.375	0.808	0.351	0.721
MSCRED [38]	0.757	0.807	0.146	0.374	0.204	0.782	0.211	0.772	0.382	0.841	0.340	0.715
OmniAnomaly [28]	**0.762**	0.856	0.223	0.417	0.227	0.871	0.228	0.841	0.384	0.842	0.365	0.769
MTAD-GAT [40]	0.743	0.848	**0.332**	**0.552**	0.235	0.878	0.276	**0.854**	0.366	0.848	**0.390**	**0.796**
THOC [26]	0.612	0.851	0.199	0.506	0.241	0.886	0.244	0.845	0.168	0.843	0.293	0.786
TranAD [29]	0.669	0.815	**0.311**	0.495	0.251	**0.901**	0.247	0.841	0.310	**0.889**	0.358	0.788
IMDiffusion [7]	0.721	0.862	0.249	0.523	**0.269**	0.881	**0.299**	0.852	**0.388**	**0.862**	0.385	**0.796**
TimesNet [33]	0.687	**0.912**	0.119	0.504	0.183	0.862	0.198	0.734	0.245	0.843	0.286	0.771
MSHTrans	**0.776**	**0.936**	0.272	**0.568**	**0.364**	**0.904**	**0.305**	**0.861**	**0.395**	0.851	**0.422**	**0.824**

original window $\mathbf{W}$, i.e.,

$$\mathcal{L}^{\text{rec}}\left(\mathbf{W}, \mathcal{Z}_{\text{de}}^{\text{evo}}\right) = \frac{1}{SD}\left\|\mathbf{W} - \mathcal{Z}_{\text{de}}^{\text{evo}}\right\|_F^2. \tag{25}$$

As for the anomaly detection tasks, the anomaly score at the t-th timestamp is computed by

$$\mathcal{S}_t = \frac{1}{D}\left\|[\mathbf{W}^{(t)}]_{-1}, [\mathcal{Z}_{\text{de}}^{\text{evo}\,(t)}]_{-1}\right\|_F^2, \tag{26}$$

where $\text{MSE}(\cdot,\cdot)$ is the mean squared error function. The anomaly label is given by $\mathcal{Y}_t = \mathbf{1}(\mathcal{S}_t > \xi)$, where ξ is the threshold.

4.7 Computational Complexity of MSHTrans

In this subsection, we analyze the computational complexity of the primary modules in MSHTrans. The intra-scale hypergraph attention module has a complexity of $O(Nkd_{in}d_{out} + Mkd_{in})$, where M is the number of hyperedges, k is the average number of nodes connected per hyperedge, and d_{in}, d_{out} denote the input/output feature dimensions. The multi-head hypergraph convolution module requires $O(H(Mk^2 + Nd_{in}d_{out}))$, where H denotes the number of attention heads. In the time series decomposition module, the DFT and IDFT operations each require $O(N \log N)$. The average pooling operation for trend extraction has the complexity of $O(\mathcal{K}Nd_{in})$, where $\mathcal{K}$ is the pooling stride length. Thus, the overall complexity of the time series decomposition module is $O(N \log N + \mathcal{K}Nd_{in})$. Both the seasonality-trend fusion module and multi-scale fusion process have the complexity of $O(Nd_{in}d_{out})$.

5 Experimental Analyses

5.1 Datasets

In our experiments, several benchmark time series datasets are adopted to verify the effectiveness of the proposed model, including SWaT, WADI, MSL, SMAP and SMD. These datasets encompass monitoring data from various domains, including water utility data, server information, and machine sensor data. Furthermore, to better visualize the detection outcomes of the proposed model for various types of anomalies, we generate a synthetic dataset with periodic patterns and introduce different types of noises as anomalous timestamps to be detected. Detailed introduction to these datasets is given in **Appendix A.2**.

5.2 Compared Models

We conduct comprehensive experiments with multiple baselines, including DAGMM [43], LSTM-VAE [23], MSCRED [38], OmniAnomaly [28], MTAD-GAT [40], THOC [26], TranAD [29], IMDiffusion [7] and TimesNet [33]. These compared models include CNN-based, transformer-based, and graph-based models.

5.3 Experimental Settings

In our experiments, we compare the proposed model and other baselines with F1 scores. Meanwhile, we also provide the evaluation results with and without Point Adjustment (PA) [28], which refines the identification of anomalies to improve the accuracy of detection results. All experiments are run 5 times and we record the average F1 scores. Detailed model settings can be found in **Appendix A.3**.

5.4 Experimental Results

5.4.1 Performance on Benchmark Datasets. First, we evaluate the proposed MSHTrans model on several benchmark datasets. Table 1 summarizes the comparative performance between MSHTrans and state-of-the-art methods, with the final column indicating the average F1 scores across all evaluated datasets. The experimental results demonstrate that our framework outperforms baseline methods in most scenarios, achieving average performance improvements of 8.21% (without PA) and 3.52% (with PA) over the second-best model. Notably, MSHTrans exhibits significant advantages compared to the graph-based MTAD-GAT model. Furthermore, it also maintains superiority over both transformer-based and CNN-based competitors. These observations validate the effectiveness and superiority of the proposed MSHTrans.

5.4.2 Visualization of Anomaly Scores and Series Reconstruction. In Figure 3, we visualize the predicted results of MSHTrans on the multivariate synthetic data that include both point and sequence anomalies. First, MSHTrans can accurately detect distinct types of anomalies. When an anomaly occurs in a certain sensor, the model fails to reconstruct features similar to the original series based on the learned hypergraph relationships and other global signals. This is because the model can only perform coarse-grained sequence reconstruction based on long-term dependencies in hypergraphs,

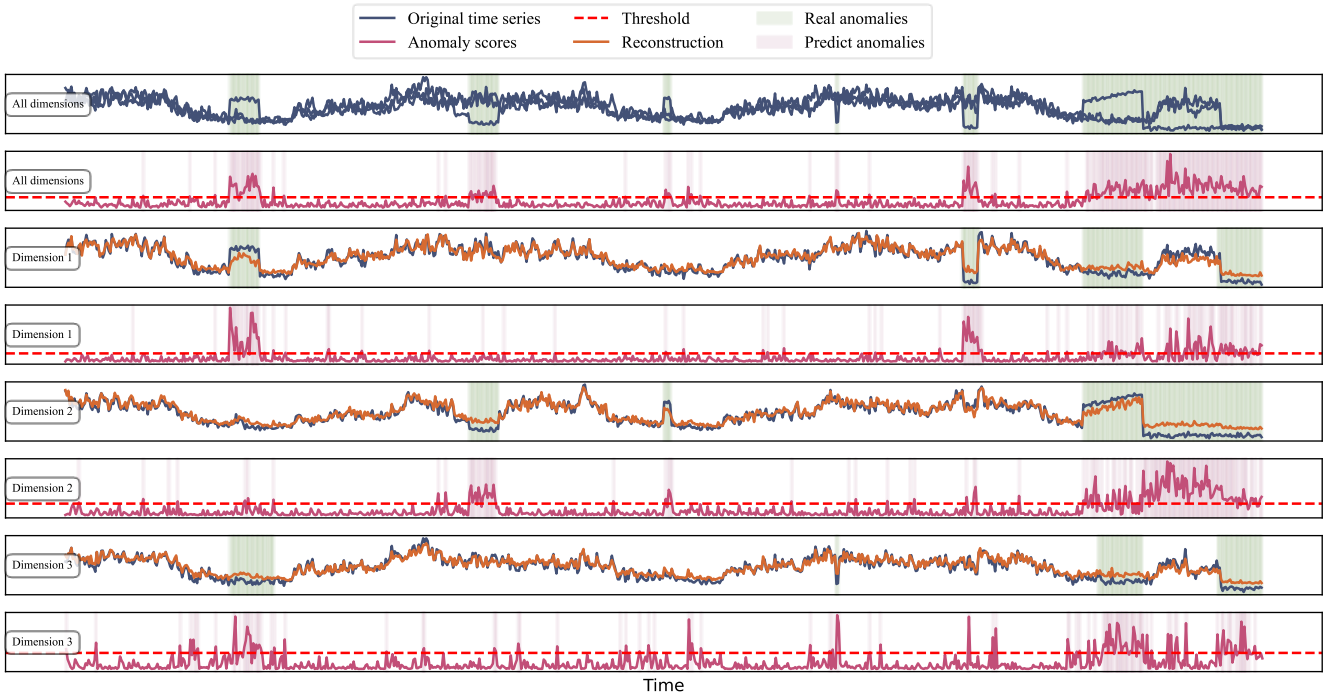

Figure 3: Visualization of outcomes (anomaly scores and reconstructed time series) of MSHTrans.

as well as global signals such as seasonality and trends. Short-term dependencies fail to work due to the occurrence of anomalies, leading to ineffective neighborhood feature propagation. When the time series experiences prolonged anomalies (e.g., the end of the second dimension in Figure 3), the long-term dependencies also fail, resulting in a further increase in reconstruction errors and anomaly scores. Second, when all dependencies approach failure, the model can only perform simple data reconstruction based on the time series analysis conducted by the time series decomposition modules, causing the anomaly scores to stabilize at a relatively high value, thereby prompting the model to detect long-term anomalies. Additionally, experimental results demonstrate that the model can also effectively detect subtle anomalies in other related dimensions caused by the anomaly of a certain sensor. Lastly, the overall anomaly score can jointly consider all variables, thereby providing comprehensive anomaly detection results.

5.4.3 Visualization of Learned Hypergraphs. Figure 4 visualizes the learned fused hypergraph incidence matrix and its corresponding adjacency matrix in MSHTrans. We have the following observations from Figure 4. First, as the visualized hypergraph indicates, some of the learned hyperedges tend to connect adjacent time nodes (i.e., short-term dependencies), while other hyperedges connect more distant timestamps, thereby constructing long-term dependencies. We observe from the learned adjacency matrix that the feature propagation between timestamps is highly dependent on the current timestamp itself, resulting in a significantly diagonal nature of the adjacency matrix. Additionally, a large number of short-term dependencies form the block structure along the diagonal of the

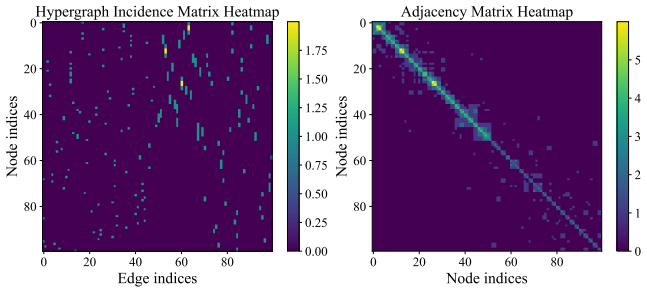

Figure 4: Visualization of learned hypergraph (left) and the corresponding adjacency matrix (right) on SWaT dataset.

adjacency matrix. These structural properties quantitatively validate the ability of the proposed hypergraph learning mechanism to jointly model multi-scale temporal dependencies.

5.4.4 Ablation Study and Impact of Window Sizes. Finally, we conduct an ablation study to investigate the effectiveness of different loss functions adopted in the proposed framework with varying window sizes, as demonstrated in Figure 5. As shown in Figure 5, the model performance declines when a particular loss function is removed. Among these loss functions, the hyperedge loss has the most significant impact. Additionally, as the window size increases, the performance of all models improves. However, the performance of the complete MSHTrans changes minimally and tends to stabilize when the window size exceeds 40. This observation indicates that the model can adaptively learn effective long-term and short-term

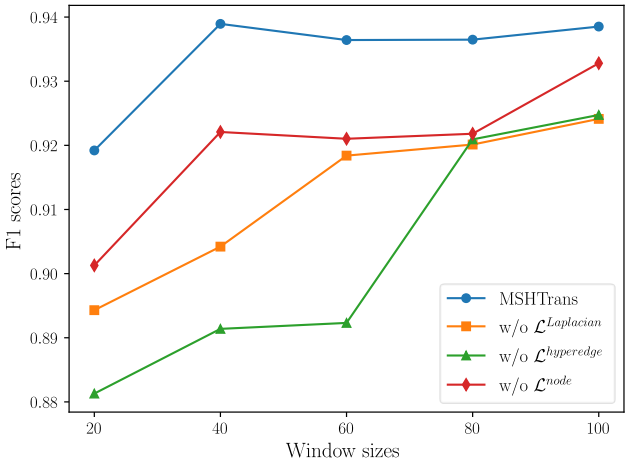

Figure 5: Ablation study (adjusted F1 scores) of MSHTrans with different window sizes on SWaT dataset.

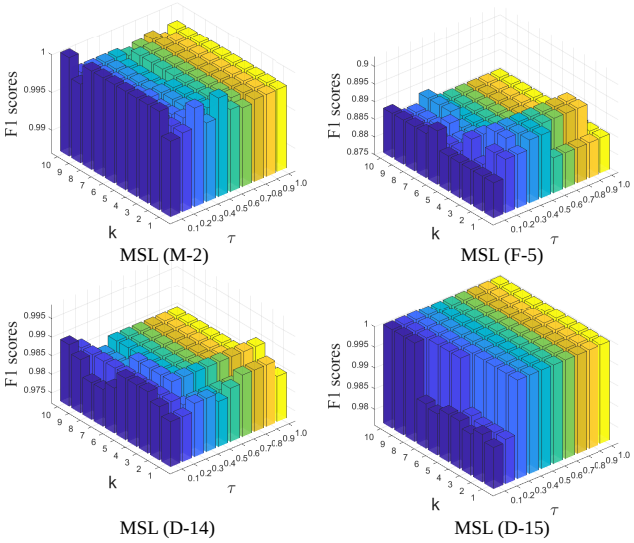

Figure 6: Parameter sensitivity w.r.t. k and τ of MSHTrans.

dependencies within windows of varying sizes, thereby achieving optimal performance for the current window and minimizing the negative impacts of insufficient window sizes.

5.5 Parameter Sensitivities

In this subsection, we investigate the parameter sensitivities (adjusted F1 scores) w.r.t. k and τ, that is, the hyperparameters in top-k selection and updating rate for hypergraph construction, as shown in Figure 6. Firstly, we can observe that a small value of k (e.g., $k = 1$) often leads to poor performance due to the insufficient number of nodes connected by hyperedges in the hypergraph. This issue may be alleviated with an increase in the updating rate τ, as the model can more quickly adjust the hypergraph based on the loss functions. Furthermore, as k increases, the performance generally improves or stabilizes. However, the optimal updating rate τ may vary across different datasets. A higher τ can sometimes result in performance degradation, possibly due to the instability in the graph structure caused by overly rapid updates to the hypergraph, which negatively impacts performance. Therefore, a moderate k value and updating rate τ are conducive to achieving the optimal model performance. Based on experimental experience, we uniformly set $k = 5$ and $\tau = 0.3$ to achieve optimal performance across most datasets.

Table 2: Performance (F1 scores) of MSHTrans with different scale numbers on SWaT dataset (window size: 100).

# Scales	# Nodes	# Hyperedges	F1 scores w/o PA	F1 scores with PA
1	[100]	[50]	0.701	0.881
2	[100, 50]	[50, 30]	0.751	0.921
3	[100, 50, 25]	[50, 30, 20]	**0.776**	**0.936**
4	[100, 50, 25, 12]	[50, 30, 20, 10]	**0.775**	**0.938**

5.6 Impact of Scale Numbers

Table 2 illustrates the impact of scale numbers of the proposed MSHTrans, which also shows the node numbers and hyperedge numbers at each scale. Experimental results show that performance improves progressively as the number of scales increases, significantly outperforming the single-scale baseline (i.e., modeling only fine-grained timestamp correlations within the initial window). Generally, when the number of scales is larger than 3, the additional performance gains from increasing the scale number become marginal. Therefore, this paper sets the number of scales to 3, which is sufficient to adequately model the timestamp dependencies within the window and achieve excellent performance.

6 Conclusion

To leverage the short-term and long-term correlations in time series anomaly detection tasks, we propose a novel multi-scale hypergraph transformer model named MSHTrans that integrates adaptive hypergraph learning and time series decomposition analysis. The proposed model succeeds in utilizing trainable hypergraphs to model both short-term and long-term dependencies while capturing global signals through time series decomposition and integration. Comprehensive experimental results demonstrate that the proposed MSHTrans can fully utilize fine-grained and coarse-grained representations, achieving superior performance compared to other state-of-the-art methods. This study primarily focuses on dependencies within individual scales and consistency between scales. As for future work, we will explore cross-scale dependencies and non-linear relationships using graph-based approaches.

7 Acknowledgment

This work is supported by the Hong Kong Research Grant Council (with Grant No. RIF R2002-20F), and the Seed Funding for Collaborative Research Grants of HKBU (with Grant No. RC-SFCRG/23-24/R2/SCI/06). We also appreciate Dr. Ke Zhu from the Hong Kong Electric Company (HEC), who significantly facilitated our research on real-world time series anomaly detection.

References

[1] Siddharth Bhatia, Arjit Jain, Pan Li, Ritesh Kumar, and Bryan Hooi. 2021. Mstream: Fast anomaly detection in multi-aspect streams. In *Proceedings of the Web Conference*. 3371–3382.

[2] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. 2021. A review on outlier/anomaly detection in time series data. *Comput. Surveys* 54, 3 (2021), 1–33.

[3] Paul Boniol, Michele Linardi, Federico Roncallo, Themis Palpanas, Mohammed Meftah, and Emmanuel Remy. 2021. Unsupervised and scalable subsequence anomaly detection in large data series. *The VLDB Journal* 30, 6 (2021), 909–931.

[4] Junfu Chen, Dechang Pi, and Xixuan Wang. 2024. A two-stage adversarial transformer based approach for multivariate industrial time series anomaly detection. *Applied Intelligence* 54, 5 (2024), 4210–4229.

[5] Peng Chen, Yingying ZHANG, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. 2024. Pathformer: Multi-scale Transformers with Adaptive Pathways for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.

[6] Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. 2020. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems* 35, 4 (2020), 83–93.

[7] Yuhang Chen, Chaoyun Zhang, Minghua Ma, Yudong Liu, Ruomeng Ding, Bowen Li, Shilin He, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. 2023. ImDiffusion: Imputed Diffusion Models for Multivariate Time Series Anomaly Detection. *Proceedings of the VLDB Endowment* 17, 3 (2023), 359–372.

[8] Zekai Chen, Dingshuo Chen, Xiao Zhang, Zixuan Yuan, and Xiuzhen Cheng. 2021. Learning graph structures with transformer for multivariate time-series anomaly detection in IoT. *IEEE Internet of Things Journal* 9, 12 (2021), 9179–9189.

[9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[10] Bowen Deng, Tong Wang, Lele Fu, Sheng Huang, Chuan Chen, and Tao Zhang. 2025. THESAURUS: Contrastive Graph Clustering by Swapping Fused Gromov-Wasserstein Couplings. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 16199–16207.

[11] Maher Dissem, Manar Amayri, and Nizar Bouguila. 2024. Neural Architecture Search for Anomaly Detection in Time-Series Data of Smart Buildings: A Reinforcement Learning Approach for Optimal Autoencoder Design. *IEEE Internet of Things Journal* 11, 10 (2024), 18059–18073.

[12] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Vol. 33. 3558–3565.

[13] Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. 2022. HGNN+: General hypergraph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 3 (2022), 3181–3199.

[14] Zilong He, Pengfei Chen, Xiaoyun Li, Yongfeng Wang, Guangba Yu, Cailin Chen, Xinrui Li, and Zibin Zheng. 2020. A spatiotemporal deep learning approach for unsupervised anomaly detection in cloud systems. *IEEE Transactions on Neural Networks and Learning Systems* 34, 4 (2020), 1705–1719.

[15] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 387–395.

[16] Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I. Webb, Irwin King, and Shirui Pan. 2024. A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 12 (2024), 10466–10485.

[17] Ming Jin, Yu Zheng, Yuan-Fang Li, Siheng Chen, Bin Yang, and Shirui Pan. 2023. Multivariate Time Series Forecasting With Dynamic Graph Neural ODEs. *IEEE Transactions on Knowledge and Data Engineering* 35, 9 (2023), 9168–9180.

[18] Fatemeh Khanmohammadi and Reza Azmi. 2024. Time-Series Anomaly Detection in Automated Vehicles Using D-CNN-LSTM Autoencoder. *IEEE Transactions on Intelligent Transportation Systems* 25, 8 (2024), 9296–9307.

[19] Gen Li and Jason J Jung. 2021. Dynamic relationship identification for abnormality detection on financial time series. *Pattern Recognition Letters* 145 (2021), 194–199.

[20] Yifan Li, Xiaoyan Peng, Jia Zhang, Zhiyong Li, and Ming Wen. 2021. DCT-GAN: dilated convolutional transformer-based GAN for time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering* 35, 4 (2021), 3632–3644.

[21] Yixin Liu, Kaize Ding, Qinghua Lu, Fuyi Li, Leo Yu Zhang, and Shirui Pan. 2023. Towards Self-Interpretable Graph-Level Anomaly Detection. In *Advances in Neural Information Processing Systems*, Vol. 36. 8975–8987.

[22] Fengcheng Lu and Michael Kwok-Po Ng. 2024. FastHGNN: A New Sampling Technique for Learning with Hypergraph Neural Networks. *ACM Transactions on Knowledge Discovery from Data* 18, 8 (2024), 1–26.

[23] Daehyung Park, Yuuna Hoshi, and Charles C Kemp. 2018. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters* 3, 3 (2018), 1544–1551.

[24] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3009–3017.

[25] Zongjiang Shang, Ling Chen, Binqing Wu, and Dongliang Cui. 2024. Ada-MSHyper: Adaptive Multi-Scale Hypergraph Transformer for Time Series Forecasting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[26] Lifeng Shen, Zhuocong Li, and James Kwok. 2020. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in Neural Information Processing Systems* 33 (2020), 13016–13026.

[27] Huan Song, Deepta Rajan, Jayaraman Thiagarajan, and Andreas Spanias. 2018. Attend and diagnose: Clinical time series analysis using attention models. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, Vol. 32.

[28] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2828–2837.

[29] Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. 2022. TranAD: deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment* 15, 6 (2022), 1201–1214.

[30] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proceedings of the 6th International Conference on Learning Representations*.

[32] Zhen Wang, Ting Jiang, Zenghui Xu, Ji Zhang, and Jianliang Gao. 2023. Irregularly Sampled Multivariate Time Series Classification: A Graph Learning Approach. *IEEE Intelligent Systems* 38, 3 (2023), 3–11.

[33] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. [n. d.]. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *The 11th International Conference on Learning Representations*.

[34] Zhihao Wu, Zhao Zhang, and Jicong Fan. 2023. Graph convolutional kernel machine versus graph convolutional networks. *Advances in Neural Information Processing Systems* 36 (2023), 19650–19672.

[35] Jaehyuk Yi and Jinkyoo Park. 2020. Hypergraph convolutional recurrent neural network. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3366–3376.

[36] Zahra Zamanzadeh Darban, Geoffrey I Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. 2024. Deep learning for time series anomaly detection: A survey. *Comput. Surveys* 57, 1 (2024), 1–42.

[37] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. 2021. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2114–2124.

[38] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. 2019. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1409–1416.

[39] Yuxin Zhang, Yiqiang Chen, Jindong Wang, and Zhiwen Pan. 2023. Unsupervised Deep Anomaly Detection for Multi-Sensor Time-Series Signals. *IEEE Transactions on Knowledge and Data Engineering* 35, 2 (2023), 2118–2132.

[40] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. 2020. Multivariate time-series anomaly detection via graph attention network. In *2020 IEEE International Conference on Data Mining*. 841–850.

[41] Wendong Zheng and Jun Hu. 2023. Multivariate Time Series Prediction Based on Temporal Change Information Learning Method. *IEEE Transactions on Neural Networks and Learning Systems* 34, 10 (2023), 7034–7048.

[42] Shuman Zhuang, Zhihao Wu, Zhaoliang Chen, Hong-Ning Dai, and Ximeng Liu. 2025. Refine then Classify: Robust Graph Neural Networks with Reliable Neighborhood Contrastive Refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 13473–13482.

[43] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*.

Appendix: Supplemental Material

A.1 Algorithm Description of MSHTrans

Algorithm 5 outlines the process of the MSHTrans framework. The detailed model structure is described as follows:

(1) **Encoder:** In each channel, the encoder sequentially performs intra-scale hypergraph attention, multi-head hypergraph convolution, time series decomposition, and fusion, with residual connections applied. Finally, multi-scale outputs and hypergraphs are aggregated through the fusion process defined in Eqs. (21) and (23).

(2) **Decoder:** The decoder adopts a single-channel structure that first performs time series decomposition on the original window to obtain initial periodic and trend components. Subsequently, the multi-head hypergraph convolution is conducted with the initial seasonality, fused time series features from encoders and the integrated hypergraph. Then another time series decomposition module is adopted to analyze the learned series features after the hypergraph message passing. Residual connections are also implemented in the decoder. Ultimately, a series fusion module is applied to generate a unified multi-scale representation, which is used to compute reconstruction loss and anomaly scores.

A.2 Dataset Descriptions

Table 3 provides the details of the datasets used in this paper. Additionally, To better visualize the effectiveness of the MSHTrans model in detecting different types of anomalies within a focused time period, we generate a synthetic dataset combining trigonometric functions and Gaussian noise, containing 3 dimensions. The test set contains both point-based and sequence-based anomalies that are either randomly generated or derived from existing datasets like MSL.

Table 3: Details of tested benchmark datasets in this paper.

Datasets	# Dimensions	Anomaly Rates	Domains
SWaT	51	0.1214	Water Monitoring
WADI	123	0.0571	Water Monitoring
MSL	55	0.1048	Spacecraft Monitoring
SMAP	25	0.1283	Spacecraft Monitoring
SMD	38	0.0416	Server Monitoring

A.3 Detailed Experimental Settings

In this subsection, we provide the common settings for all models:

- Optimizer: Adam;
- Learning rate: 0.001;
- Window size S: 100;
- Stride: 1 (MSL), 10 (other datasets).

For some specific hyperparameters of the proposed MSHTrans, the settings are listed below:

- Scale number L: 3;
- Number of top significant connections selected k: 5;
- Kernel size for convolution-based downsampling: 2
- Hyperedge numbers $\{M_s\}_{s=0}^{L-1} = \{50, 30, 20\}$;
- Hidden units in encoders and decoder: 64;
- Dropout rate: 0.1;
- Hypergraph updating rate τ: 0.3;
- Number of heads: 3.

Algorithm 5: Training Process of MSHTrans

Input : Multivariate time-series data $\mathbf{X} \in \mathbb{R}^{T \times D}$.

`/* Model Initialization; */`

Initialize all trainable weights and biases.

`/* Multi-Scale Window Sampling; */`

Generate input multi-scale window data $\{\mathbf{W}_{(s)}\}_{s=0}^{L-1}$ with Eqs. (1), (2) and (3);

while *not converge* **do**

`/* Encoder; */`

for $s = 0 \rightarrow (L-1)$ **do**

Conduct intra-scale hypergraph attention with $\text{IntraHAtt}\left(\mathbf{W}_{(s)}, \mathcal{G}_{(s)}\right)$;

Obtain $\mathbf{Z}_{(s)}$ with multi-head hypergraph convolution $\text{MHConv}\left(\hat{\mathbf{H}}_{(s)}, \mathbf{W}_{(s)}\right)$;

Obtain seasonality embeddings $\mathcal{Z}_{(s)}^{\text{sea}}$ and trend embeddings $\mathcal{Z}_{(s)}^{\text{trend}}$ with $\text{TSDecom}\left(\mathbf{W}_{(s)} + \mathbf{Z}_{(s)}\right)$;

Obtain fused $\mathcal{Z}_{(s)}^{\text{evo}} = \text{STFusion}\left(\mathbf{W}_{(s)}, \mathcal{Z}_{(s)}^{\text{sea}}, \mathcal{Z}_{(s)}^{\text{trend}}\right)$ with Eq. (20);

Conduct multi-scale fusion with Eq. (21) and get $\mathcal{Z}^{\text{fusion}} = \text{FeedForward}(\mathcal{Z}_{(0)}^{\text{evo}})$ with updated $\mathcal{Z}_{(0)}^{\text{evo}}$;

Conduct multi-scale fusion with Eq. (23) to get fused $\mathbf{H}^{\text{fusion}}$;

`/* Decoder; */`

Conduct time series decomposition for the original window with $\mathcal{Z}_{\text{de1}}^{\text{sea}}, \mathcal{Z}_{\text{de1}}^{\text{trend}} = \text{TSDecom}(\mathbf{W})$;

Conduct multi-head hypergraph convolution with $\mathbf{Z}_{\text{de}}^{\text{conv}} = \text{MHConv}\left(\mathbf{H}^{\text{fusion}}, [\mathcal{Z}_{\text{de1}}^{\text{sea}}, \mathcal{Z}^{\text{fusion}}]\right)$;

Conduct time series decomposition with $\mathcal{Z}_{\text{de2}}^{\text{sea}}, \mathcal{Z}_{\text{de2}}^{\text{trend}} = \text{TSDecom}\left(\mathbf{Z}_{\text{de}}^{\text{conv}}\right)$;

Obtain fused embeddings with $\mathcal{Z}_{\text{de}}^{\text{evo}} = \sigma\left(\text{STFusion}\left(\mathbf{W}, \mathcal{Z}_{\text{de2}}^{\text{sea}}, \mathcal{Z}_{\text{de1}}^{\text{trend}} + \mathcal{Z}_{\text{de2}}^{\text{trend}}\right)\right)$;

`/* Model Optimization; */`

Compute training loss $\mathcal{L}$ with Eq. (24);

Update all trainable parameters with gradient descent and back propagation;

`/* Obtain anomaly scores */`

Compute the anomaly scores $\mathcal{S}$ with Eq. (26);

Return Anomaly scores $\mathcal{S} \in \mathbb{R}^{T}$.

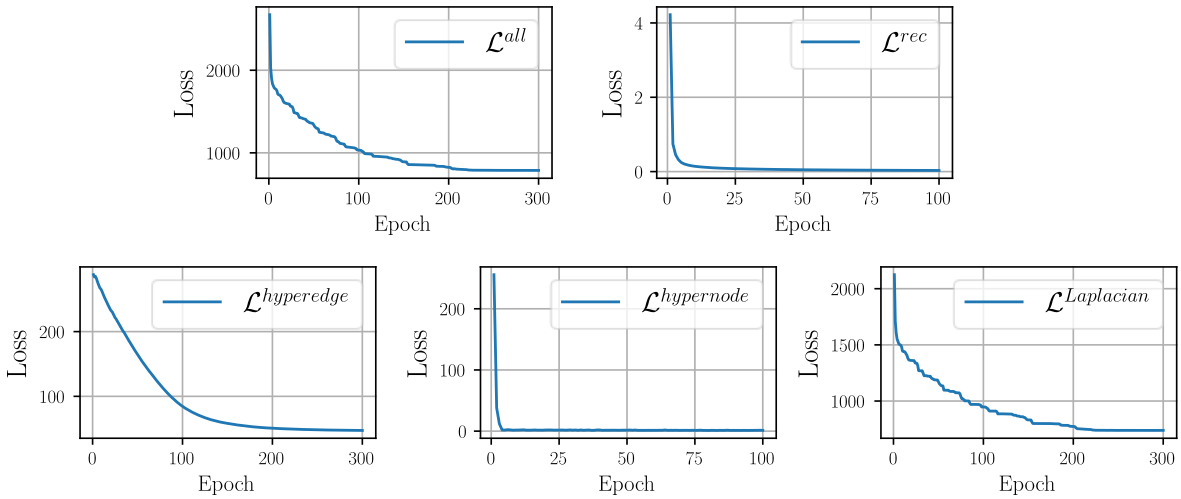

Figure 7: Curves of loss functions (total loss, reconstruction loss, hyperedge loss, node loss and Laplacian loss) of MSHTrans.

Table 4: Ablation studies (adjusted F1 scores) of MSHTrans.

Methods	SWaT	WADI	MSL	SMAP
w/o domsampling	0.920	0.499	0.889	0.846
w/o conv1D	0.902	0.544	0.901	0.844
w/o time series decomposition	0.888	0.519	0.892	0.857
w/o hypergraph transformer	0.704	0.396	0.718	0.688
w/o multi-scale fusion	0.897	0.550	0.902	0.855
MSHTrans	**0.936**	**0.568**	**0.904**	**0.861**

Table 5: Delay of MSHTrans and compared models.

Datasets	MTAD-GAT	THOC	TranAD	IMDiffusion	TimesNet	MSHTrans
SWaT	1943.0	535.0	297.1	686.3	208.2	**159.1**
WADI	5.43	1.52	0.43	8.45	**0.42**	0.53
MSL	53.9	66.3	19.4	35.5	24.8	**14.1**
SMAP	77.5	19.6	21.3	55.2	18.1	**13.5**

Table 6: Training time comparison (minutes) of MSHTrans and compared methods.

Datasets	SWaT	WADI	MSL	SMAP
DAGMM	281.04	63.48	6.85	8.48
LSTM-VAE	378.10	107.18	13.18	10.80
MSCRED	2247.71	529.45	13.07	6.85
OmniAnomaly	429.85	82.51	9.52	10.48
MTAD-GAT	41.25	3484.48	466.48	321.81
THOC	21.15	52.71	2.48	2.11
TranAD	18.46	45.41	1.74	1.66
IMDiffusion	53.11	100.48	5.48	5.33
TimesNet	54.08	133.09	7.37	53.07
MSHTrans	24.37	56.52	1.15	0.34

A.4 Convergency Analysis

Figure 7 shows the curves of different losses of the proposed MSHTrans on SWaT dataset. From the figure, we can see that the rapid and stable convergence of all loss functions demonstrates the excellent convergence properties of the model. Throughout the training process, the loss values consistently decreased at a steady rate, indicating that the optimization algorithm effectively minimized the optimization targets. Particularly, the reconstruction loss, which is closely related to the computation of anomaly scores, rapidly decreases at the early stages of training and eventually converges to a value close to zero.

A.5 Ablation study

Ablation study is conducted to examine the significance of each components in MSHTrans, as shown in Table 4. Experimental results indicate that each module is helpful to MSHTrans, especially hypergraph transformer with an average performance improvement of 31.86%. This observation reveals that hypergraph transformer is the key module in the proposed MSHTrans.

A.6 Delay of Anomaly Detection

We further look into the delay of MSHTrans (w/o point adjustment) and compared models in time series anomaly detection tasks, as shown in Table 5. Experimental results reveal that our method generally has a lower delay (or latency) of anomaly detection. Because the learned hypergraph-derived relationships (especially short-term dependencies) can promptly discover anomalous patterns, MSHTrans has a significant advantage of detecting anomalies timely.

A.7 Training Time Comparison

Table 6 shows the average training time (mins) required for MSHTrans and compared models. It can be observed that although MSHTrans is not always the most time-efficient algorithm, its overall training time is comparable to other SOTA methods. This further validates its effectiveness on real-world anomaly detection tasks.