Diffusion Graph Model for Time Series Anomaly Detection via Anomaly-aware Graph Sparsification and Augmentation

Disen Lan*†
South China University of Technology
Guangzhou, China
disenlan1002@gmail.com

Guibin Zhang*†
Tongji University
Shanghai, China
guibinz@outlook.com

Rongjin Guo
South China University of Technology
Guangzhou, China
ethereal3529@gmail.com

ABSTRACT

Unsupervised methods, particularly reconstruction-based methods have become the dominant approach for multivariate time series anomaly detection (TSAD), which distinguish between normal and abnormal series based on the magnitude of the reconstruction error. However, in this process, the heterophilic connections (normal ↔ abnormal datapoints) often lead the model to simultaneously capture the distributions of both normal and abnormal data, impeding effective anomaly detection based on reconstruction error. To address this challenge, we introduce a novel diffusion graph model framework, dubbed `DiG`, which jointly models spatial-temporal correlations and explicitly severs heterophilic connections for improved reconstruction. Specifically, `DiG` first transforms multivariate time series into a spatial-temporal joint graph and utilizes graph diffusion to progressively denoise and learn anomaly-free node representations. Through the tailored anomaly-aware graph sparsification and contrastive augmentation, `DiG` effectively captures anomaly patterns and eliminates anomaly-related heterophily correlations on the spatio-temporal joint graph. Extensive experiments on TSAD datasets demonstrate that `DiG` achieves state-of-the-art performance, showcasing the expressiveness of our framework.

CCS CONCEPTS

• **Information systems** → **Data mining**; • **Mathematics of computing** → **Time series analysis**.

KEYWORDS

Time series anomaly detection, Graph neural networks, Diffusion model

1 INTRODUCTION

Time series anomaly detection is an important research direction in the field of signal processing and analysis. Its goal is to identify those data points in time series or signal data that significantly deviate from normal pattern. These anomalies may represent some unusual events, faults, errors or potential opportunities, which are of great value to many practical applications. However, anomalies are typically rare and often obscured by a vast amount of normal data points. Therefore, time series anomaly detection is usually based on unsupervised settings because the process of time series data sampling and labeling is difficult and costly.

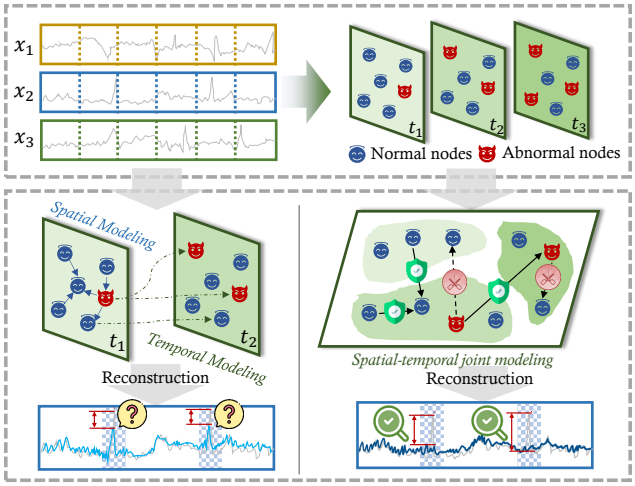

Figure 1: Time series anomaly detection based on spatio-temporal joint modeling and reconstruction method.

The development of deep learning has led to the widespread application of neural network methods in time series anomaly detection. Traditional TS-based methods like Long Shorm-Term Memory (LSTM) network [15] can model intra-series temporal correlations and perform anomaly detection based on prediction or reconstruction errors [2] in an explicit manner, while ignoring the inter-series correlations which limits the model's ability to detect complex anomaly patterns in multivariate time series data. Graph neural networks (GNNs) [10, 26] demonstrates the potential for modeling the relationship between time and variables. Existing graph-based time series anomaly detection methods adopt spatio-temporal graph neural networks (STGNNs) to model inter- and intra-series correlations in multivariate time series data [1, 8, 18]. However, these methods define full-connected graph structures for modeling spatio-temporal correlations without considering anomaly awareness, which may be affected by the harmful impact of unexpected and noisy abnormal data when modeling normal data distributions.

Current neural network-based unsupervised anomaly detection approaches can be divided into prediction-based and reconstruction-based. If a model is trained on a large number of normal time points, it is more likely to fail to predict or reconstruct the anomaly data [30]. Benefiting from the remarkable generation ability of diffusion models[7, 24], recent research works focus on improving time series analysis performance by leveraging the strong power of diffusion models [13, 17, 32], which inspires us to employ diffusion

*Equal Contribution.
†Corresponding Authors.

WWW'25 AI4TS Workshop, April 28-May 2, Sydney, Australia
2025. ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/XXXXX

models to learn and generate high quality graph structures for capturing spatio-temporal correlations in multivariate time series data. However, due to the difficulty in obtaining labels in unsupervised settings, guiding diffusion process for graph structures learning still remains a tough challenge.

Further, the dominance of normal time series patterns and the scarcity of anomaly have become a consensus among researchers. For the very small number of abnormal data, they have inconsistent similarities with the vast majority of other normal data. Therefore, the model will still tends to reduce their reconstruction errors reconstructs the abnormal data as effectively as possible. The aforementioned observations prompt question about anomaly detection: *Can we ideally reconstruct anomaly into normal data so as to maximize its reconstruction error for better anomaly detection*?

In this context, we introduce diffusion graph model, namely `DiG`, for time series anomaly detection. Specifically, (1) we propose a novel *Diffusion Graph Process* for high quality spatio-temporal graph generation, which can take advantage of the power of diffusion models in generation and the strong ability of GNNs in spatio-temporal dependency modeling by corrupting and recovering the correlations in graph through the diffusion graph forward and reverse processes; and (2) we design *Anomaly-aware Heterophily Graph Sparsification* and *Contrastive Augmentation*, which can aware and identify suspected anomaly in the graph structure by perceiving heterophily correlations, destroy their connections with other nodes and augment the normal time series representation learning, forcing to model the distribution of normal data, making the reconstruction of abnormal time points more difficult and detect them more precisely.

Overall, this paper presents following contributions:

- We propose a novel framework named `DiG` for multivariate time series anomaly detection, which can utilize *Diffusion Graph Process* to generate high quality spatio-temporal graph structure that is able to model spatio-temporal joint distributions and correlations in multivariate time series data naturally.
- We design *Anomaly-aware Heterophily Graph Sparsification*, which can destroy the anomaly-related connections by perceiving spatio-temporal heterophily between normal and abnormal data, providing guidance for diffusion graph generation and achieving anomaly awareness graph sparsification in unsupervised settings.
- We introduce *Anomaly-aware Contrastive Augmentation*, which utilizes the consistency of normal data to enhance representation learning, and promote anomaly modeling as normal data by using normal data as anchors through anomaly-aware contrastive learning and augmentation.
- We conduct extensive experiments regarding time series anomaly task on real-world TSAD datasets. `DiG` achieves the state-of-the-art anomaly detection results, demonstrating the effectiveness of our proposed framework.

2 RELATED WORK

Unsupervised Time Series Anomaly Detection. Since obtaining scarce abnormal labels is difficult and costly, time series anomaly detection is usually established in an unsupervised setting [29]. OmniAnomaly [30] utilize unsupervised reconstruction in LSTM-VAE with a normalizing flow to detect anomalies. Anomaly Transformer [29] utilizes association discrepancy in time series to detect anomalies with minimax strategy. However, modeling intra-series dependency while ignoring the inter-series correlations would limit the model's ability to detect complex anomaly patterns in multivariate time series data.

Graph Neural Networks. Graph Neural Networks (GNNs) demonstrate their potentials in many real-world scenarios by leveraging their powerful topological awareness [26, 33–35], and have been derived many variants such spatio-temporal graph neural networks (STGNNs) [8, 18, 22], which can capture the complex spatio-temporal correlations effectively. However, STGNNs often use different types of deep neural networks to model the multivariate or spatio dependencies and temporal correlations separately [18], making it hard to capture potential cross spatio-temporal correlations. In addiction, previous STGNNs emphasize the significance of spatio-temporal correlations modeling, while ignoring potential heterophilic correlations between normal and abnormal nodes when conducting anomaly detection task.

Diffusion Models for Time Series Analysis. Recently, diffusion models have seen widespread application in time series [4, 27, 31]. For instance, TimeDiff [17] uses the distribution of past observations to drive the diffusion model to generate moment values in the future for time series forecasting task. DiffShape [13] uses the distribution of subsequences as a condition and then employs diffusion models to generate shapelets for time series classification task. Concurrent research work mostly utilize u-nets or transformers [7, 24] as the backbone architecture in diffusion process. However, these diffusion models may not be suitable for tasks like time series anomaly detection that require establishing spatio-temporal joint correlations. We further explore combining GNN and diffusion model for modeling spatio-temporal correlations and joint distribution in denoising process simultaneously and naturally.

3 METHODOLOGY

3.1 Notations & Problem Definition

3.1.1 Notations. Considering a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ denotes the nodes and $\mathcal{E}$ denotes the edges. We use $\mathbf{Z} \in \mathbb{R}^{N \times D}$ to denote the feature matrix of the graph $\mathcal{G}$, where $N = |\mathcal{V}|$ representing the number of nodes and each node $v_i \in \mathcal{V}$ has an D-dimensional feature vector. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ represents the spatio dependency between nodes, where $\mathbf{A}[i, j] > 0$ if $(v_i, v_j) \in \mathcal{E}$ else $\mathbf{A}[i, j] = 0$ when $(v_i, v_j) \notin \mathcal{E}$. We can denote graph neural network (GNN) as $\mathcal{G} = \{\mathbf{Z}, \mathbf{A}\}$ alternatively.

3.1.2 Problem Definition. In time series anomaly detection task, the input multivariate time series data denotes $\mathbf{X} \in \mathbb{R}^{N \times L}$ which is connected from N sensors with the length of L time points. Our goal is to output a label vector $\mathbf{Y} \in \mathbb{R}^{L}$ for each sensor to determine whether the i-th time point y_i is an anomaly. It should be noted that the training dataset does not have anomaly labels, only the testing dataset has labels in the unsupervised settings. The forecast or reconstruction errors can be used as discrepancy measures between anticipated and real signals. Due to anomalies are usually

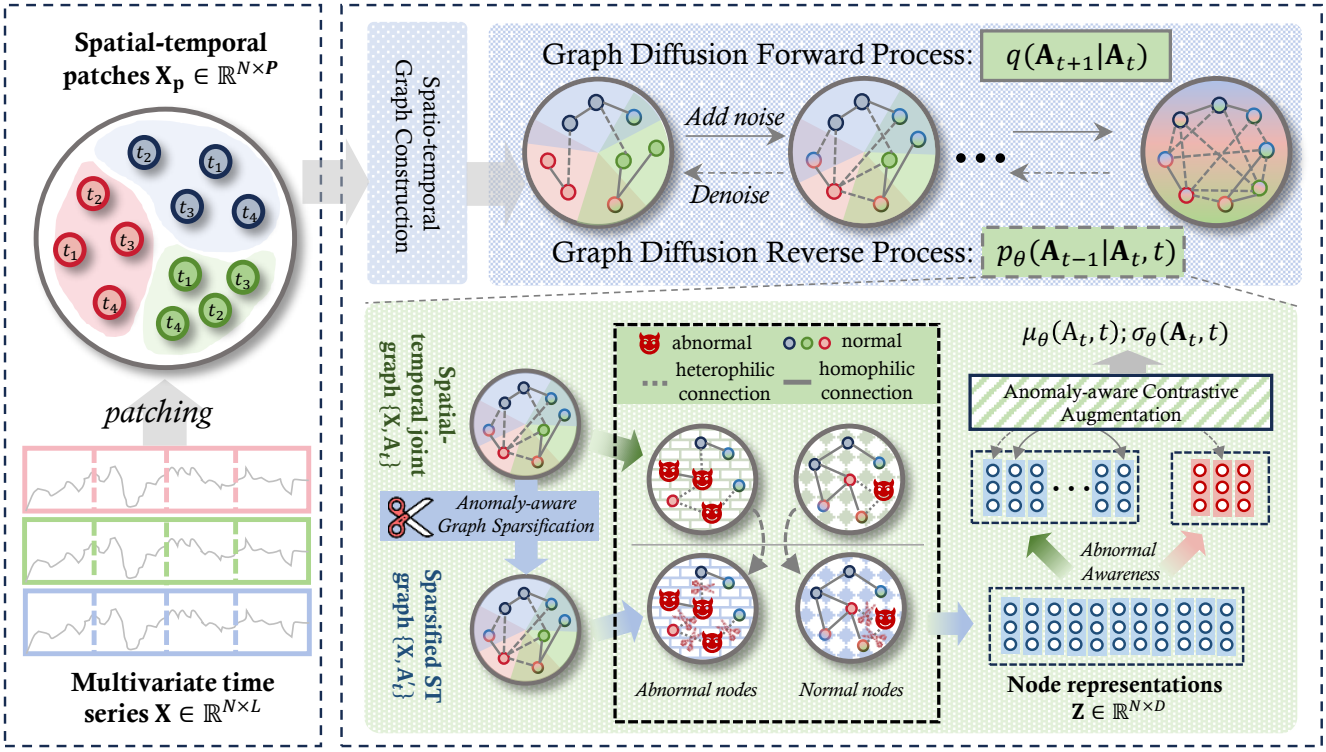

Figure 2: Overall framwork of DiG.

unexpected emergencies and extremely rare, if a model is trained on vast amount of normal data, it will fail to forecast or reconstruct some time points which are more likely associated with anomalies.

3.2 Overall Framework

We proposes a novel framework named DiG for multivariate time series anomaly detection. Figure 2 illustrates the overall framework of our proposed DiG. First we introduce Diffusion Graph DiG Block to generate the spatio-temporal joint graphs, which can take advantage of the power of diffusion models in generation [7, 24] and the strong ability of GNNs [1, 18] in spatio-temporal dependency modeling by addressing the negative correlations of noisy and unexpected anomalies in time series data. In specific, we corrupt the correlations in the original graph structures, then recover the original correlations through the denoising diffusion process, thereby mitigating the noisy anomaly-related correlations and generating spatio-temporal graph structures. Moreover, to achieve the anomaly-aware graph structure generation and to minimize the impact of abnormal data on the model, we design an *Anomaly-aware Heterophily Graph Sparsification* and *Contrastive Augmentation* method that guides DiG for high quality graph generation. This method allows the model to aware and the anomaly-related heterophily correlations, which are noisy and irrelevant for the normal time series data modeling and reconstruction, encourage the model to learn the normal data distribution, restore abnormal data to normal during the reconstruction stage, thereby detecting anomalies better and improving performance.

In the following parts, we will delve into the technical details of DiG, including *Diffusion Graph Process*, *Anomaly-aware Heterophily Graph Sparsification* and *Anomaly-aware Contrastive Augmentation*.

3.3 Diffusion Graph Process

We introduce Diffusion Graph (DiG), a novel diffusion model architecture for spatio-temporal joint modeling in multivariate time series. We first construct a spatio-temporal joint graph to capture temporal, spatio, including cross spatio-temporal correlations, thereby being able to discover potential anomaly-related correlations from a wide range of normal data distributions. Besides, there is a forward process with steps $t \in \{0, 1, \ldots, T\}$ introducing random noise to corrupt the original graph structures, which simulates the unpredictable abnormal noise and anomaly effects that are widely present in the real world; and a reverse process with steps $t \in \{T, T-1, \ldots, 0\}$ in our diffusion graph models focusing on denoising and recovering the noise-corrupted graph structures, mitigating the negative and task-irrelevant noise impacts, thereby boosting graph robustness and performance in modeling normal data distributions.

3.3.1 spatio-temporal Joint Graph Construction. Instead of using two different components for separate spatio-temporal modeling, we attempt to contract spatio-temporal joint graph. Given an input multivariate time series data $\mathbf{X} \in \mathbb{R}^{N \times L}$, where N denotes the number of sensers (spatio dimension) and L denotes the length of each time series (temporal dimension), in order to model spatio-temporal

correlations jointly, we first combine the spatio and temporal dimension in a uniform way and then utilize tokenization or patching [14] to obtain richer semantic information in time series $\mathbf{X}_P \in \mathbb{R}^{L_P \times P}$, where $L_P = \frac{N \times L}{P}$ and P is the patch length. For the convenience of description, we will write L_P as N representing the number of spatio-temporal joint nodes in the following contents (not to be confused with the spatio dimension N presented in the above). We use a simple linear layer $\mathrm{F} : P \rightarrow D$ to project $\mathbf{X}_P$ into the latent space and obtain the spatio-temporal joint node features $\mathbf{Z} = \mathrm{F}(\mathbf{X}_P) + \mathbf{PE} \in \mathbb{R}^{N \times D}$ with extra position embedding [20].

Because there is no predefined adjacency matrix $\mathbf{A}$, we can obtain it by: $\mathbf{A} := \sigma(\mathbf{Z} \cdot \mathbf{Z}^T)$ by the learnable node features, where σ is an activation function such as Softmax$(\cdot)$ to control the edge correlation values between 0 and 1, then we obtain the graph $\mathcal{G} = (\mathbf{Z}, \mathbf{A})$. By constructing the spatio-temporal joint graph, besides the spatio correlations and temporal correlations, we can also model the cross spatio-temporal correlations in a direct and universal way.

3.3.2 Diffusion Graph Forward Process. Given the input spatio-temporal joint graph $\mathcal{G}$ with the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N,N}$ representing the spatio-temporal joint correlations between every two nodes, we denote the original input as $\mathbf{A}_0$. The diffusion graph forward process $q(\mathbf{A}_t|\mathbf{A}_{t-1})$ conforms to the Markov hypothesis, random noise sampled from the Gaussian distribution is gradually added to the original graph structure $\mathbf{A}_0$ until it becomes a completely random Gaussian distribution $\mathbf{A}_T$. Specifically, $\mathbf{A}_0$ as well as condition information (such as diffusion timestep t) would be transformed in the forward process as follows:

$$q(\mathbf{A}_{t+1}|\mathbf{A}_t) = \mathcal{N}(\mathbf{A}_t; (1 - \sqrt{\beta_t})\mathbf{A}_{t-1}, \beta_t \mathbf{I}) \tag{1}$$

where β_t is a hyperparameter representing the noise schedule. We employ a reparameterization method [7] such that $\mathbf{A}_t = \sqrt{\bar{\alpha}_t}\mathbf{A}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$ where $\bar{\alpha}_t = \sum_{i=1}^{t}(1 - \beta_i)$ and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

3.3.3 Diffusion Graph Reverse Process. During diffusion graph reverse process, we aims to eliminate the noise introduced in forward process, and recover the original adjacency matrix $\mathbf{A}$ representing node correlations. The reverse process $p(\mathbf{A}_{t-1}|\mathbf{A}_t)$ converts the Gaussian noise-corrected $\mathbf{A}_T \sim \mathcal{N}(0, \mathbf{I})$ to the generated $\mathbf{A}_0$ using a learnable neural network, which is also based on the Markov chain:

$$p_\theta(\mathbf{A}_{t-1}|\mathbf{A}_t, t) = \mathcal{N}(\mathbf{A}_{t-1}; \mu_\theta(\mathbf{A}_t, t), \sigma_\theta(\mathbf{A}_t, t)^2 \mathbf{I}) \tag{2}$$

where $\mu_\theta(\mathbf{A}_t, t)$ and $\sigma_\theta(\mathbf{A}_t, t)$ represent the mean and variance of the reverse process at diffusion step t, respectively, and both are parameterized by θ. Based on [7], we set $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ and $\tilde{\beta}_1 = \beta_1$, μ_θ and σ_θ can be parameterized as:

$$\begin{cases} \mu_\theta(\mathbf{A}_t, t) = \dfrac{1}{\sqrt{\alpha_t}}(\mathbf{A}_t - \dfrac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}}\epsilon_\theta(\mathbf{A}_t, t)) \\ \sigma_\theta(\mathbf{A}_t, t) = \sqrt{\tilde{\beta}_t} \end{cases} \tag{3}$$

where $\epsilon_\theta(\mathbf{A}_t, t)$ denotes the predicted noise level under diffusion step t, which is implemented by neural networks with trainable parameters θ. Pratically, we empoly a Multi-Layer Perceptron (MLP) to implement the network parameterized by θ that takes current input $\mathbf{A}_t$ and the timestep embeddding of t as inputs to predict ϵ_θ and generate $\mathbf{A}_{t-1}$.

3.3.4 Diffusion Graph Optimization. In practive, the diffusion graph reverse process can be summerized as learning to predict the Gaussian noise $\epsilon_\theta(\mathbf{A}_t, t)$ then solving $\mu_\theta(\mathbf{A}_t, t)$ according to Eq. 3. In this way, the objective of diffusion graph optimization is to minimize the mean square error between model predicted noise and the Gaussian noise:

$$\mathcal{L}_{diff} = \mathbb{E}_{t, \mathbf{A}_0, \epsilon}||\epsilon - \epsilon_\theta(\mathbf{A}_t, t)||_2^2 \tag{4}$$

In addition, we incorporate the denoising diffusion implicit models [9] to speed up the diffusion graph generation.

3.4 Anomaly-aware Heterophily Graph Sparsificaion

In order to better distinguish normal and abnormal data and detect anomalies in multivariate time series, we design *Anomaly-aware Heterophily Graph Sparsificaion* As shown in Figure 2, we first construct graph for the input time series and learn the node features and connections or adjacency matrix between nodes.

3.4.1 Anomaly-Related Heterophily Graph Connections & Sparsification. Generally, the normal nodes tend to be homogeneous with high features similarity, while anomaly node features are often significantly different from those of other normal nodes. However, a very small number of anomaly nodes show heterogeneity and are clearly distinguished from normal nodes, so the correlations between normal and abnormal is very weak [23]. The constructed spatio-temporal joint graph often contains noisy or heterogeneous edges which may lead to suboptimal performance in anomaly detection task, because the connections between normal and abnormal nodes are noisy and irrelevant, making it difficult to model the normal data distributions and should be removed, namely ***Anomaly-Related Heterophily Connections***.

Our work focuses on remove the *anomaly-related heterophily connections* and construct a sparse subgraph $\mathcal{G}^{\text{sub}}$ which can filter out abnormal noise edges in $\mathcal{G}$, and reconstruct the distribution of normal data effectively. Note that most of the data are normal, the anomaly nodes are highly rare and sparse, we simply utilize k-neighbor subgraph [38] for *anomaly heterophily connections* sparsification, each node in the subgraph connects no more than k edges from its neighbor nodes ($k < N$). Note that the input graph is constructed by the spatio-temporal joint method, the graph is full connected and we can cut off the spatio, temporal and spatio-temporal joint *anomaly-related heterophily correlations or edges*.

Different form the heterogeneous connections defined in previous heterogeneous GNNs [16] based on the given labeled dataset in supervised settings, we must distinguish the normal and abnormal time points in unsupervised settings for the reason that it is difficult to obtain the rare anomaly labels. Given an input spatio-temporal joint grpah $\mathcal{G} = (\mathbf{Z}, \mathbf{A})$ and integer k, we sparsicify k-neighbor subgraph by calculating nodes' similarity and sample the top-k large as *homogeneous edges* and top-$(N - k)$ small as ***anomaly-related heterophily edges*** correspondingly. We design a sparse graph structure learner $f^{\mathcal{G}}(\cdot)$ for anomaly-related heterogeneous

graph sparcification. Specifically, for $\forall u, v \in \mathcal{V}(u \neq v)$, we utilize $f_\phi(\cdot)$ to calculate the edge features or similarity in $\mathcal{G}$:

$$\pi_{u,v} = f^{\mathcal{G}}(z_u, z_v) \tag{5}$$

Then we employ Softmax$(\cdot)$ function to scale the value of similarity to $[0, 1)$:

$$a_{u,v} = \frac{\exp(\pi_{u,v})}{\sum_{w \in \{\mathcal{V}/u\}} \exp(\pi_{u,w})} \tag{6}$$

For all node $u \in \mathcal{V}$ and its neighbors $v \in \{\mathcal{V}/u\}$, we select top-k large values in $\mathbf{A}$ as *homogeneous edges* and top-$(N-k)$ small as ***anomaly-related heterophily edges*** correspondingly. We define sparse graph mask m_g where *homogeneous edges* are 1 and ***anomaly-related heterophily edges*** are 0. We can utilize m_g to remove the *anomaly-related heterophily edges*, guiding our `DiG` to generate sparse graph structure $\mathbf{A}' = \mathbf{A} \odot m_g$, whose *anomaly-related heterophily edges* have been masked and removed.

3.5 Anomaly-aware Contrastive Augmentation

When obtaining the heterophily sparse graph, the normal-abnormal wise heterophily correlations have been removed, retaining the normal-normal wise and abnormal-abnormal wise homogeneous edges. Since the impact of anomalies on modeling normal data distribution is reduced, the model can better reconstruct normal data. Similarly, abnormal data can also reduce the reconstruction error and lead to better reconstruction, but this is not what we expect. We need to reconstruct normal data well and amplify the reconstruction error of abnormal data to perform anomaly detection. Therefore, we propose *Anomaly-aware Contrastive Augmentation* to solve this problem.

To achieve anomaly awareness, based on the dominance of normal data and the scarcity of abnormal data, if a node is not similar to most other data, it can be considered an anomaly, so we can utilize node embedding similarities to distinguish normal data and potential anomalies. Specifically, given the node features $\mathbf{Z} \in \mathbb{R}^{N \times D}$, we calculate the similarity matrix $\mathbf{M}_{sim} = \frac{\mathbf{Z} \cdot \mathbf{Z}^T}{|\mathbf{Z}||\mathbf{Z}|} \in \mathbb{R}^{N \times N}$, and select the top-$S$ nodes most non-similar as abnormal nodes, while the left top-$(N-S)$ as normal nodes:

$$\begin{cases} \mathcal{V}^- = \mathcal{V}_{abnormal} = \text{argmin}(\sum_{j=1}^{N} \mathbf{M}_{sim}[i, j], S), \\ \mathcal{V}^+ = \mathcal{V}_{normal} = \mathcal{V} - \mathcal{V}^- = \mathcal{V} - \mathcal{V}_{abnormal} \end{cases} \tag{7}$$

where $\mathcal{V}^+$ is the normal node set of indices corresponding to the top-S elements, and $\mathcal{V}^-$ is the potential abnormal node set of indices. Note that $S \ll N - S$ for the scarcity of abnormal nodes practically.

In time series anomaly detection scenarios, there is consistency in normal data, and exists a clear distance between normal and abnormal data. From the normal data perspective, we utilize contrastive learning [3], which can align normal distribution and increase the representation difference between normal and abnormal data naturally. Formally, the normal contrastive loss is defined as based on InfoNCE loss [19]:

$$\mathcal{L}_{con}^{nor} = -\frac{1}{|\mathcal{V}^+|} \sum_{i \in \mathcal{V}^+} \log \frac{\mathbb{1}_{j \in \mathcal{V}^+} \exp(\text{sim}(z_i, z_j)\tau)}{\sum_{k=1}^{N} \exp(\text{sim}(z_i, z_k)/\tau)} \tag{8}$$

where sim$(\cdot)$ denotes inner product and τ is the temperature coefficient. From the abnormal data perspective, our objective is not modeling anomaly and reconstruct perfectly but focusing on amplify its reconstruction error for detection. Therefore, we can also adopt contrastive learning. The difference is that for abnormal data, we use normal data as positive examples to align its representation to the normal distribution, while use the same type abnormal data as negative samples to separate the abnormal data, which making it reconsturct abnormal data into normal data, and expand the reconstruction error. Formally, similar with normal contrastive loss, the abnormal contrastive loss is defined as:

$$\mathcal{L}_{con}^{abn} = -\frac{1}{|\mathcal{V}^-|} \sum_{i \in \mathcal{V}^-} \log \frac{\mathbb{1}_{j \in \mathcal{V}_+} \exp(\text{sim}(z_i, z_j)\tau)}{\sum_{k \in \mathcal{V}^-} \exp(\text{sim}(z_i, z_k)/\tau)} \tag{9}$$

The total contrastive loss for anomaly-aware contrastive augmentation is as follows:

$$\mathcal{L}_{con} = \alpha \mathcal{L}_{con}^{nor} + \beta \mathcal{L}_{con}^{abn} \tag{10}$$

where α and β are loss coefficients for controlling their weights, respectively.

3.6 Objective Function

The graph structures $\mathbf{A}$ generated by *Diffusion Graph Process* and *Anomaly-aware Heterophily Graph Sparsification* as well as the node features $\mathbf{Z}$ can be fed into graph neural network to reconstruct original time series data $\widehat{\mathbf{X}} = \mathcal{G}(\mathbf{Z}, \mathbf{A})$.

Considering that the training purpose of the proposed methodology contains minimizing the time series reconstruction, optimizing the diffusion graph model by diffsion loss (Eq. 4) and anomaly-aware contrastive augmentation by contrastive loss (Eq. 10), the total objective function of `DiG` is defined as follows:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_1 \mathcal{L}_{diff} + \lambda_2 \mathcal{L}_{con} \tag{11}$$

where the reconstruction loss $\mathcal{L}_{rec} = ||\mathbf{X} - \hat{\mathbf{X}}||_2^2$ is the mean square error (MSE). To trade off different losses in Eq. 11, we adopt the tuning strategy presented by [5] for adjusting the hyperparameter λ_1 and λ_2 automatically.

4 EXPERIMENTS

4.1 Experiment Setup

Datasets. We conducted experiments on five real-world datasets for time series anomaly detection task. We summerized the statistics of datasets in Table 2: (1) SMD (Server Machine Dataset). (2) MSL (Mars Science Laboratory rover). (3) SMAP (Soil Moisture Active Passive satellite). (4) SWaT (Secure Water Treatment). (5) PSM (Pooled Server Metrics).

Baselines. We extensively compare `DiG` with other 9 popular time series anomaly detection baselines, including: OmniAnomaly [30], Anomaly Transformer [29], TimesNet [25], GDN [6], MTAD-GAT [37], GReLeN [36], DiffAD [27], D^{3}R [21] and ImDiffusion [4], covering TS-based, graph-based and diffusion-based methods.

Table 1: Experiment results of time series anomaly detection. The P, R and F1 represent precision, recall and F1-score (as %), respectively. A higher value of P, R and F1 indicates a better anomaly detection performance.

Dataset	SMD			MSL			SMAP			SWaT			PSM			AVG
Metric	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
OmniAnomaly [30]	83.34	94.49	88.57	88.67	91.17	89.89	74.16	97.76	84.34	81.42	84.30	82.83	88.39	74.46	80.83	85.51
ATransformer [29]	88.55	92.67	90.56	91.45	91.48	91.46	93.52	98.55	**95.97**	88.43	92.06	90.21	90.22	98.79	94.31	92.50
TimesNet [25]	76.81	85.09	80.74	81.19	78.05	79.59	83.78	61.87	71.18	88.87	93.06	90.92	98.22	92.21	95.21	83.53
GDN [6]	71.70	99.74	83.42	99.13	82.41	90.00	74.80	98.91	85.18	99.35	68.12	80.82	99.11	85.92	92.05	86.29
MTAD-GAT [37]	82.10	92.15	86.83	79.17	98.24	87.68	89.06	91.23	90.13	97.18	69.57	81.09	95.89	89.87	92.79	87.70
GReLeN [36]	88.00	94.73	91.24	88.71	90.63	89.66	81.92	92.54	86.90	95.60	83.50	89.10	94.20	92.10	93.10	90.00
DiffAD [27]	85.23	73.26	78.79	86.92	59.15	70.40	93.64	88.95	91.23	96.52	86.44	91.20	97.46	88.36	92.68	84.86
D^3R [21]	86.36	79.63	82.86	92.95	57.10	70.74	96.55	90.95	93.67	92.67	94.90	**93.77**	99.27	92.89	95.97	87.40
ImDiffusion [4]	77.88	64.49	70.55	93.97	53.71	68.35	94.35	52.60	67.54	99.85	65.51	79.12	99.82	81.95	90.01	75.11
DiG (Ours)	90.10	93.74	**91.88**	92.09	95.15	**93.59**	94.26	93.81	94.03	91.20	91.65	91.42	95.16	97.58	**96.36**	**93.46**

Table 2: Statistic of datasets

Datasets	Length	Channels	Anomaly Ratio (%)
SMD	1,416,825	38	4.2%
MSL	132,046	55	10.5%
SMAP	562,800	25	12.8%
SWaT	944,919	51	12.1%
PSM	220,322	25	27.8%

Implementation. We set the input time series length of 128 universally. The patch length is set to 8 in a non-overlapping manner and latent dimension is set to 32. We define heterophily graph sparsity ratio r, which is set to 20%. By defining r we can calculate $K = r \times N \times N$ for anomaly-related heterophily edges awareness and $S = r \times N$ for anomaly nodes awareness. For the diffusion model, the diffusion step is 100, and we chose a linear schedule β from 10^{-4} to 0.02. The timestep embedding dimension is 128. We simply adopt GCN [12] as the graph model backbone. The running epochs is 10, following the previous research work. We opt for Adam [11] optimizer. The batch size is 128 and the learning rate is 1e-4. All the baselines follow their original experiment settings. All the experiments are implemented in Pytorch with a single NVIDIA RTX 3090 24GB GPU. We utilize reconstruction error as anomaly score based on train and valid dataset, and adopt the widely-used point adjustment strategy, following previous works [28–30].

4.2 Main Results

We summarize the experiment performance of DiG regarding time series anomaly detection task. As shown in Table 1, DiG achieves the best average-level F1 score, demonstrating superior performance compared to the baselines. Compare to TS-based anomaly detection methods, we can achieve 0.96% ~ 9.93% F1-score improvement. It is worth mentioning that compared with other graph-based and diffusion-based methods, our diffusion-based graph generation method can achieve 3.46% ~ 18.35 % improvement.

4.3 Model Analysis

Ablation study. We conduct ablation experiments on DiG and its following five variants: (1) **DiG w/o *Spar.*** denotes removing *anomaly-aware heterophily graph sparsification.* (2) **DiG w/o *CA.*** denotes that we does not employ *anomaly-aware contrastive augmentation* to optimize our model. (3) **DiG w/o *Diff.* & Spar.**. denotes that both *diffusion graph process* and *anomaly-aware graph sparsification* are removed in the study. (4) **DiG w/o *Spar. & CA.*** denotes we neither adopt graph sparsification nor contrastive augmentation for anomaly detection. (5) **DiG w/o *Diff, & Spar. & CA.*** denotes removing all our designed three components, remaining the vanilla graph model for time series reconstruction and anomaly detection.

As shown in Table 3, our ablation study indicates that all our proposed components in DiG framework are effectiveness and helpful for anomaly detection in multivariate time series, demonstrating that: (1) the graph structure generated by *diffusion graph process* is more efficient in modeling of spatio-temporal correlations in multivariate time series compared to vanilla graph model; (2) The ability of anomaly detection can be effectively improved by heterogeneous edges sparsification and anomaly-aware contrastive augmentation. In summary, removing any one component deteriorates the anomaly detection performance of DiG.

Visualization analysis. To explain how our model works, we provide visualization by ablating *graph sparsification* and *contrastive augmentation.* As shown in Figure 3, compared with reconstruction-based model (left column), since normal-abnormal wise heterogeneous correlations are removed by *anomaly-aware heterophily graph sparsification*, both normal and anomaly data can be reconstructed more closed to the original input time series data. However, since the influence of normal data on anomalies is reduced and the homogeneous correlations between anomalies is still preserved, they can be reconstructed better, thus reducing the reconstruction error that leads to detection errors (middle column). When DiG combines

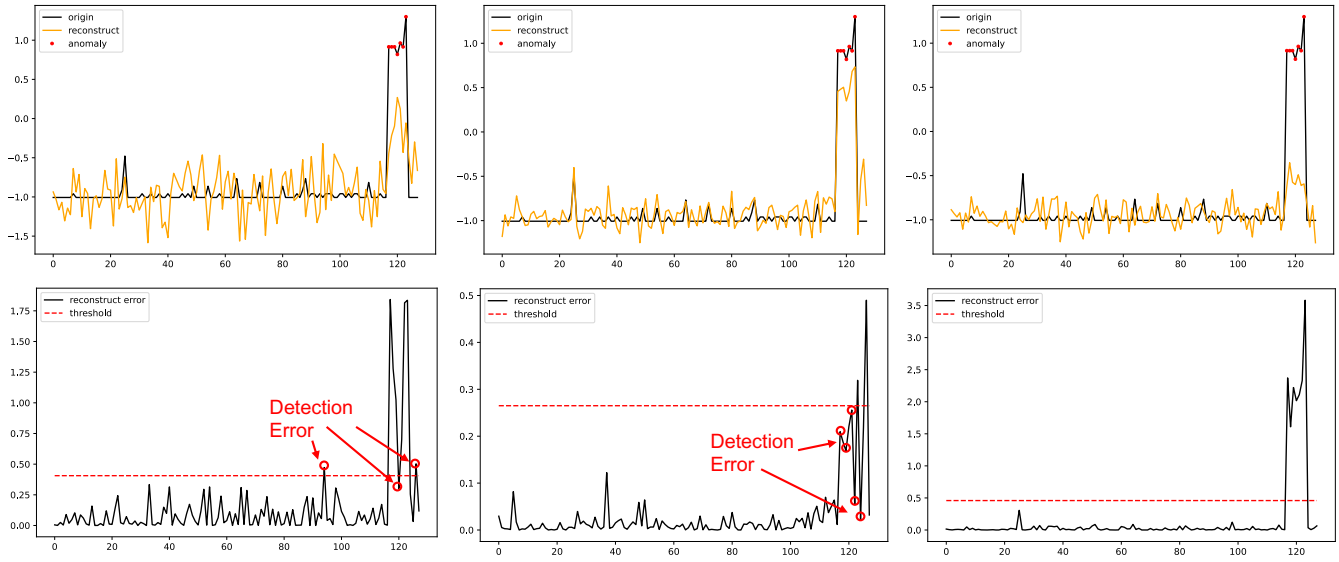

Figure 3: Visualization of different variations of DiG. The left column figures are belong to DiG w/o sparsification and contrastive augmentation. The middle column figures are belong to DiG w/o contrastive augmentation and the right column figures are belong to DiG. The top row figures are the input time seires data and reconstruction output from MSL dataset, and the bottom row figures are the reconstruction error (anomaly score) and the threshold which is used for anomaly detection.

Table 3: Ablation study on DiG. F1-scores (%) for DiG and its ablated versions are reported.

Dataset	SMD	MSL	SMAP	SWaT	PSM	AVG
w/o *Spar.*	89.24	90.23	89.65	90.97	95.38	91.09
w/o *CA.*	88.47	90.66	91.89	89.16	95.47	91.13
w/o *Diff. & Spar.*	85.73	86.74	87.51	87.50	93.99	88.29
w/o *Spar. & CA.*	82.92	79.59	85.97	83.62	92.27	84.87
w/o *Diff. & Spar. & CA.*	79.66	74.46	78.26	79.67	90.84	80.58
DiG	**91.88**	**93.59**	**94.03**	**91.42**	**96.36**	**93.46**

graph sparsification with *contrastive augmentation*, normal data representation can be modeled, and abnormal data can be aligned with normal data features through anomaly-aware contrastive learning, so as to reconstruct anomaly as normal data as possible and amplify the reconstruction error of abnormal data for more precise anomaly detection (right column).

Parameter sensitivity. We analyzed how the performance of DiG is influenced by changing the model hyperparameters, including patch length, latent dimension, the number of GNN layers and graph sparsity rate. As depicted in Figure 4, our findings include: (1) DiG is insensitive to the patch length choices from 2 to 16. To trade off the performance and efficiency, we opt for the patch length of 8. (2) The latent dimension of 32 with 3 GNN layers can achieve the optimal F1-score, indicating the best anomaly detection performance. (3) DiG is sensitive to the graph sparsity ratio greater than 30%. Due to both heterophily graph sparsification and contrastive augmentation based on abnormal awareness can be influenced by the graph sparse ratio r, too large r will cause the model to aware more number of abnormal data, thereby increasing the impact of abnormal edges and node features, bringing more irrelevant noise and leading to a decrease in anomaly detection performance. Therefore, we select the optimal graph sparse ratio of 20%.

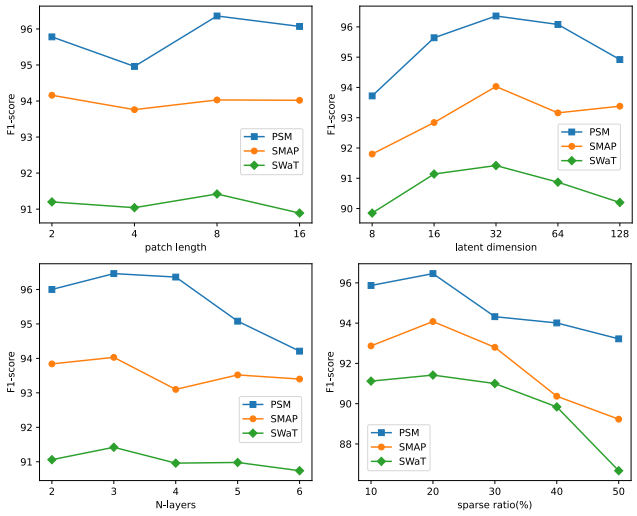

Figure 4: Parameter sensitivity analysis on patch length, latent dimension, the number of GNN layers and graph sparsity rate.

5 CONCLUSION

In this paper, we proposed `DiG`, a novel graph-based diffusion model for time series anomaly detection. `DiG` adopts diffusion graph process to generate the spatio-temporal joint graph structure, and removes the heterophily correlations between normal and abnormal data through anomaly-aware graph sparsificaton to . We also introduce anomaly-aware contrastive augmentation to amplify anomaly reconstruct error and perform more precise detection. `DiG` achieves the state-of-the-art performance on real-world time series anomaly detection datasets, demonstrate the effectiveness of our proposed framework. In the future, we plan to extend our proposed framework to more real-work scenarios and contribute the anomaly detection community.

REFERENCES

[1] Zhanxing Zhu Bing Yu, Haoteng Yin. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *IJCAI*.

[2] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *Comput. Surveys* (2009).

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*.

[4] Yuhang Chen, Chaoyun Zhang, Minghua Ma, Yudong Liu, Ruomeng Ding, Bowen Li, Shilin He, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. 2024. ImDiffusion: Imputed Diffusion Models for Multivariate Time Series Anomaly Detection. In *VLDB*.

[5] Roberto Cipolla, Yarin Gal, and Alex Kendall. 2018. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *CVPR*.

[6] Ailin Deng and Bryan Hooi. 2021. Graph Neural Network-Based Anomaly Detection in Multivariate Time Series. In *AAAI*.

[7] Jonathan Ho, Ajay Jain, and Pieter Abbee. 2020. Denoising Diffusion Probabilistic Models. In *NeurIPS*.

[8] Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. 2025. TimeFilter: Patch-Specific Spatial-Temporal Graph Filtration for Time Series Forecasting. arXiv:2501.13041 [cs.LG] https://arxiv.org/abs/2501.13041

[9] Stefano Ermon Jiaming Song, Chenlin Meng. 2021. Denoising Diffusion Implicit Models. In *ICLR*.

[10] Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I. Webb, Irwin King, and Shirui Pan. 2023. A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection. In *arXiv preprint arXiv:2307.03759*.

[11] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. In *ICML*.

[12] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[13] Zhen Liu, Wenbin Pei, Disen Lan, and Qianli Ma. 2024. Diffusion Language-Shapelets for Semi-supervised Time-Series Classification. In *AAAI*.

[14] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *ICLR*.

[15] 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* (2018).

[16] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *arXiv preprint arXiv:2002.05287*.

[17] Lifeng Shen and James T. Kwok. 2023. Non-autoregressive Conditional Diffusion Models for Time Series Prediction. In *ICML*.

[18] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *AAAI*.

[19] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. In *NeurIPS*.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *NeurIPS*.

[21] Chengsen Wang, Zirui Zhuang, Qi Qi, Jingyu Wang, Xingyu Wang, Haifeng Sun, and Jianxin Liao. 2023. Drift doesn't Matter: Dynamic Decomposition with Diffusion Reconstruction for Unstable Multivariate Time Series Anomaly Detection. In *NeurIPS*.

[22] Kun Wang, Hao Wu, Yifan Duan, Guibin Zhang, Kai Wang, Xiaojiang Peng, Yu Zheng, Yuxuan Liang, and Yang Wang. 2024. NuwaDynamics: Discovering and Updating in Causal Spatio-Temporal Modeling. In *ICLR*.

[23] Kun Wang, Guibin Zhang, Xinnan Zhang, Junfeng Fang, Xun Wu, Guohao Li, Shirui Pan, Wei Huang, and Yuxuan Liang. 2024. The heterophilic snowflake hypothesis: Training and empowering gnns for heterophilic graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3164–3175.

[24] Peebles William and Xie Saining. 2023. Scalable Diffusion Models with Transformers. In *ICCV*.

[25] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *ICLR*.

[26] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *TNNLS* (2020).

[27] Chunjing Xiao, Zehua Gou, Wenxin Tai, Kunpeng Zhang, and Fan Zhou. 2023. Imputation-based Time-Series Anomaly Detection with Conditional Weight-Incremental Diffusion Models. In *SIGKDD*.

[28] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Zhihan Li Jiahao Bu, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, Jian Chen, Zhaogang Wang, and Honglin Qiao. 2020. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *WWW*.

[29] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Anomaly transformer: Time series anomaly detection with association discrepancy. In *ICLR*.

[30] Su Ya, 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 *SIGKDD*.

[31] Yiyuan Yang, Ming Jin, Haomin Wen, Chaoli Zhang, Yuxuan Liang, Lintao Ma, Yi Wang, Chenghao Liu, Zenglin Xu Bin Yang, Jiang Bian, Shirui Pan, and Qingsong Wen. 2024. A survey on diffusion models for time series and spatio-temporal data. In *arXiv preprint arXiv:2404.18886*.

[32] Xinyu Yuan and Yan Qiao. 2024. Diffusion-TS: Interpretable Diffusion for General Time Series Generation. In *ICLR*.

[33] Guibin Zhang, Xiangguo Sun, Yanwei Yue, Chonghe Jiang, Kun Wang, Tianlong Chen, and Shirui Pan. 2024. Graph sparsification via mixture of graphs. *arXiv preprint arXiv:2405.14260* (2024).

[34] Guibin Zhang, Kun Wang, Wei Huang, Yanwei Yue, Yang Wang, Roger Zimmermann, Aojun Zhou, Dawei Cheng, Jin Zeng, and Yuxuan Liang. 2024. Graph Lottery Ticket Automated. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=nmBjBZoySX

[35] Guibin Zhang, Yanwei Yue, Kun Wang, Junfeng Fang, Yongduo Sui, Kai Wang, Yuxuan Liang, Dawei Cheng, Shirui Pan, and Tianlong Chen. 2024. Two heads are better than one: Boosting graph sparse training via semantic and topological awareness. *arXiv preprint arXiv:2402.01242* (2024).

[36] Weiqi Zhang, Chen Zhang, and Fugee Tsung. 2022. GRELEN: Multivariate Time Series Anomaly Detection from the Perspective of Graph Relational Learning. In *IJCAI*.

[37] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. 2021. Multivariate Time-Series Anomaly Detection via Graph Attention Network. In *ICDM*.

[38] Cheng Zheng, Bo Zong, Wei Cheng, Dongjin Song, Jingchao Ni, Wenchao Yu, Haifeng Chen, and Wei Wang. 2018. Robust graph representation learning via neural sparsification. In *ICML*.