

MedGNN: Towards Multi-resolution Spatiotemporal Graph Learning for Medical Time Series Classification

Wei Fan*
weifan.oxford@gmail.com
University of Oxford
Oxford, UK

Jingru Fei*
jingrufei@bit.edu.cn
Beijing Institute of Technology
Beijing, China

Dingyu Guo
kevguo@uchicago.edu
University of Chicago
Chicago, US

Kun Yi†
kunyi.cn@gmail.com
State Information Center of China
North China Institute of Computing
Technology
Beijing, China

Xiaozhuang Song
xiaozhuangsong1@link.cuhk.edu.cn
The Chinese University of Hong
Kong, Shenzhen
Shenzhen, China

Haolong Xiang
hlxiang@nuist.edu.cn
Nanjing University of Information
Science and Technology
Nanjing, China

Hangting Ye
yeht2118@mails.jlu.edu.cn
Jilin University
Changchun, China

Min Li†
limin@mail.csu.edu.cn
Central South University
Changsha, China

Abstract

Medical time series has been playing a vital role in real-world healthcare systems as valuable information in monitoring health conditions of patients. Accurate classification for medical time series, e.g., Electrocardiography (ECG) signals, can help for early detection and diagnosis, thus improving patient outcomes and the quality of life. Traditional methods towards medical time series classification rely on handcrafted feature extraction and statistical methods; with the recent advancement of artificial intelligence, the machine learning and deep learning methods have become more popular. However, existing methods often fail to fully model the complex spatial dynamics under different scales, which ignore the dynamic multi-resolution spatial and temporal joint inter-dependencies. Moreover, they are less likely to consider the special baseline wander problem as well as the multi-view characteristics of medical time series, which largely hinders their prediction performance. To address these limitations, we propose a Multi-resolution Spatiotemporal Graph Learning framework, *MedGNN*, for medical time series classification. Specifically, we first propose to construct multi-resolution adaptive graph structures to learn dynamic multi-scale embeddings. Then, to address the baseline wander problem, we propose Difference Attention Networks to operate self-attention mechanisms on the finite difference for temporal modeling. Moreover, to learn the multi-view characteristics, we utilize the Frequency Convolution Networks to capture complementary information of medical time series from the frequency domain. In addition, we introduce

the Multi-resolution Graph Transformer architecture to model the dynamic dependencies and fuse the information from different resolutions. Finally, we have conducted extensive experiments on multiple medical real-world datasets that demonstrate the superior performance of our method. Our Code is available at this repository: <https://github.com/aikunyi/MedGNN>.

CCS Concepts

• **Applied computing** → **Health care information systems**.

Keywords

Medical Time Series; Spatiotemporal Learning; Graph Neural Networks; Time Series Classification

ACM Reference Format:

Wei Fan, Jingru Fei, Dingyu Guo, Kun Yi, Xiaozhuang Song, Haolong Xiang, Hangting Ye, and Min Li. 2025. MedGNN: Towards Multi-resolution Spatiotemporal Graph Learning for Medical Time Series Classification. In *Proceedings of the ACM Web Conference 2025 (WWW '25)*, April 28-May 2, 2025, Sydney, NSW, Australia. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3696410.3714514>

1 Introduction

Medical time series data, such as Electroencephalography (EEG) and Electrocardiography (ECG) signals, has been playing a vital role in real-world healthcare systems by providing valuable information in monitoring health conditions of patients. EEG signals, which measure the electrical activity of the brain, are widely used to diagnose and monitor various neurological disorders, including epilepsy, Alzheimer's disease, and sleep disorders [7]. Similarly, ECG signals, which record the electrical activity of the heart, are essential for diagnosing and monitoring cardiovascular diseases, such as arrhythmias, myocardial infarction, and congestive heart failure [4]. Classifying these medical time series is of paramount importance as it could enable the early detection of abnormalities,

*Both authors contributed equally to this research.

†Corresponding authors.



accurate diagnosis, and personalized treatment. By identifying patterns and features indicative of specific conditions, medical time series classification can assist clinicians in making timely diagnoses [14] and facilitate adapting treatment plans accordingly, potentially leading to improved patient outcomes and quality of life.

Traditionally, medical time series classification has been primarily relied on handcrafted feature extraction, which often involve domain expertise to identify relevant features from the raw data. For instance, for ECG analysis, many features such as R-peak amplitude and heart rate variability would be manually extracted [4]. Later, statistical methods have also been applied to medical time series classification: the autoregressive models, hidden Markov models, and Gaussian mixture models have been used to capture the temporal dependencies and dynamics in ECG and EEG signals [30, 33, 35]. Though statistical methods can provide robust results and handle uncertainty, they often make strong assumptions about the data distribution and may struggle with complex, non-linear patterns. With the advent of artificial intelligence, various deep learning methods have been applied to medical time series classification [25]: convolution neural networks have been particularly successful in learning representations directly from raw time series, such as EEGNet [19]; Transformer-based methods have been applied into the medical time series classification [38]. In addition, graph neural network has also been adopted for multivariate time series classification [48, 50].

However, these methods often fail to fully model the complex spatial (channel) dynamics under different scales, which ignore the dynamic multi-resolution spatial and temporal joint interdependencies. Moreover, most of them are usually for general classification, without considering the special problem, such as baseline wander, as well as the multi-view characteristics in medical time series, which largely hinders their prediction performance. To address these limitations, we aim to propose a novel framework to learn the multi-scale and multi-view representations for medical time series classification. However, several challenges arise in achieving this goal: (i) *how to model the dynamic spatial structures between different time series channels with multiple resolutions?* Since the medical time series usually includes multiple channels, the dynamic spatial dependencies keep changing with the resolutions or scales of time series, which need to be properly modeled for accurate classification; (ii) *how to learn the multi-view characteristics of medical time series while addressing the baseline wander problem?* The baseline wander problem [5], i.e., the constant offsets or slow drifts towards baseline measurements of medical series would always hinders the models in learning the key patterns and fluctuations; meanwhile the multi-view characteristics of medical time series based on the features from both the time domain and the frequency domain are usually ignored, largely hindering the classification.

To tackle these challenges, we introduce a Multi-resolution Spatiotemporal Graph Learning framework, *MedGNN*. Specifically, in our MedGNN framework, we first propose to construct multi-resolution adaptive graph structures to learn dynamic spatial temporal representations, where we utilize different kernels of convolutions to extract multi-scale medical time series embeddings to cover the local and global patterns. We construct multi-resolution graphs based on the learned embeddings to model the dynamic spatial dependencies among different channels; the graph structures are adaptively learned to reflect the changing correlations at

different resolutions. Then, to address the baseline wander problem and learn the multi-view characteristics, we propose two novel networks for temporal modeling: (i) the Difference Attention Networks focus on the temporal changes in the medical time series, which operates self-attention mechanisms on the finite difference (e.g., first-order difference) along the temporal dimension, targeting capturing key temporal patterns while mitigating the impact of baseline wander; (ii) the Frequency Convolution Networks captures complementary information in the frequency domain by applying Fourier transform and frequency-domain convolutions [44, 47], providing a multi-view perspective of the temporal dynamics for medical time series. In addition, to learn the complicated multi-resolution spatiotemporal graph representations, we utilize the Multi-resolution Graph Transformer architecture to model the dynamic spatial dependencies and fuse the information from different resolutions. Our main contributions are mainly as follows:

- We propose a novel approach for medical time series classification to capture the complex multi-view spatiotemporal dependencies and multi-scale dynamics of medical time series through multi-resolution learning.
- We construct adaptive graph structures at different resolutions to model spatial correlations among time series channels and utilize a Multi-resolution Graph Transformer architecture for the resolution learning and information fusion.
- We propose Difference Attention Network and Frequency Convolution Network, for temporal modeling to overcome baseline wander problem of medical time series and meanwhile capture multi-view characteristics from the time and frequency domain.
- We have conducted extensive experiments on multiple medical time series datasets, including both ECG and EEG signals, to demonstrate the superior performance of our proposed framework compared to state-of-the-art methods, highlighting its great potential for the real-world clinical applications.

2 Related Work

2.1 Medical Time Series Classification

Time series classification is a crucial yet challenging problem in the field of data mining, as it involves identifying patterns in sequential data over time [12]. Medical time series, as a specialized form of time series data collected from human physiological signals, such as EEG [43] and ECG [18], present unique challenges and opportunities [15]. Continuously analyzing medical time series, especially as new conditions or classes of data emerge, is essential for health monitoring [40] and making informed medical decisions [1], highlighting the importance of medical time series classification.

Traditionally, one of the most widely used approaches for medical time series classification has been the nearest neighbor (NN) [21] classifier, often combined with distance measures such as dynamic time warping (DTW) [2] or shapelet-based methods [6]. These techniques have demonstrated effectiveness in various applications due to their simplicity and interpretability. Later statistical models such as the autoregressive models [30] and Gaussian mixture models [35] have been used to capture the medical time series. In recent years, deep learning methods have significantly advanced the field of medical time series classification. For example, EEGNet [19] introduced the use of depthwise and separable convolutions to develop

a model specifically designed for EEG data, capturing essential EEG feature extraction techniques for brain-computer interfaces (BCI). COMET [37] proposed a hierarchical contrastive representation learning framework that is tailored to the unique characteristics of medical time series data. Medformer [38] introduced a multi-granularity patching transformer architecture that addresses the specific challenges of medical time series classification, providing a specialized solution for capturing complex temporal patterns.

2.2 Graph Neural Networks for Time Series

Graph neural networks (GNNs) have shown promising performance in time series analysis due to their ability to capture complex dependencies between time-series variables [16]. By representing data as a graph, GNNs can effectively model relationships across different variables and time steps. Some GNN-based models, such as STGCN [49] and DCRNN [20], rely on a pre-defined graph structure, which is often unavailable or difficult to determine in many real-world scenarios. To address this limitation, recent research has focused on learning graph structures directly from the data, enabling automatic modeling of the topological relationships among variables. AGCRN [3] enhances graph convolutional networks through a data-adaptive graph generation module and a node-adaptive parameter learning module. MTGNN [42] introduces an effective approach to learn and exploit the inherent dependencies among variables. FourierGNN [46] captures frequency domain spatial-temporal correlations. More recent models have continued to push the boundaries of GNN-based time series analysis. RainDrop [52] introduces a GNN framework designed to handle irregularly sampled and multivariate time series, learning the dynamics of sensors directly from observational data without requiring any prior knowledge of the relationships. SimTSC [50] presents a simple yet general framework that uses GNNs to model similarity information, which helps improve time series classification by leveraging similarity patterns across different time points and variables. MTS2Graph [48] offers a strategy by constructing a graph that captures the temporal relationships between extracted patterns at each layer of the network. These methods underscore the development in GNN-based methods, highlighting their potential to revolutionize time series classification by effectively capturing complex dependencies and adapting to diverse scenarios.

3 Problem Formulation

Given a collection of medical time series from N participants denoted by $\{P_1, \dots, P_N\}$, each participant has multiple samples of collected series $\{X_{P_n}^1, \dots, X_{P_n}^{k_n}\}$, where k_n is the number of samples for the n -th participant. Each data sample $X_{P_n}^i \in \mathbb{R}^{T \times C}$ represents the collected multivariate medical time series (e.g., multi-lead ECG) in one participation, where T denotes the number of timestamps and C is the number of channels; the corresponding label for sample $X_{P_n}^i$ is represented as $y_{P_n}^i \in \{0, 1\}$ for binary medical classification problems where 0 indicates a healthy participant and 1 indicates a participant diagnosed with a specific disease, or represented as $y_{P_n}^i \in \{1, 2, \dots, c\}$ for multi-class medical classification problems where each class corresponds to one kind of diseases or conditions. The objective of medical time series classification problem is to learn a mapping function $f : \mathbb{R}^{T \times C} \rightarrow \mathbb{R}^1$ that can accurately

predict the label based on medical time series samples in each participation. Formally, given a participant's time series sample $X_{P_n}^i$, the goal is to predict the corresponding label $y_{P_n}^i$ to indicate the disease or condition, which can be written as:

$$\hat{y}_{P_n}^i = f(X_{P_n}^i; \theta), \quad (1)$$

where $\hat{y}_{P_n}^i$ is label for i -th sample for n -th participant; the mapping function f is parameterized by the learnable parameters θ .

To ensure the real-world clinical utility of the model, it is important to evaluate the generalization ability on unseen participants. Thus we split the dataset based on participants: specifically, we let \mathcal{P}_{train} , \mathcal{P}_{val} , and \mathcal{P}_{test} denote the disjoint sets of participants used for training, validation, and testing, respectively. With training and testing are on different participants, the generalization of the developed model can be evaluated on unseen participants or patients, which can simulate a more realistic estimate of its potential performance in real-world clinical settings. In addition, since one patient may visit hospitals for tests for many times, we also consider a complementary settings that split the dataset into training, validation and test data only relying on individual samples that can be represented by \mathcal{X}_{train} , \mathcal{X}_{val} and \mathcal{X}_{test} as disjoint sets of samples.

4 Methodology

To address the aforementioned challenges of medical time series classification, in this section, we elaborate on our proposed *MedGNN*, a Multi-resolution Spatiotemporal Graph Learning framework. The overall architecture of MedGNN is illustrated in Figure 1. It mainly consists of multi-resolution graph construction, difference attention networks, frequency convolution networks and multi-resolution graph transformer. For the medical time series, multi-resolution graph construction is utilized to learn the dynamic spatiotemporal representations, and then difference attention networks and frequency convolution networks are designed to capture the comprehensive temporal dynamics. Finally, multi-resolution graph transformer is employed to model the dynamic spatial dependencies from different resolutions. In the following, we elaborate on the core components of MedGNN from Section 4.1 to Section 4.4.

4.1 Multi-resolution Graph Construction with Multi-scale Embedding Learning

Medical time series typically consist of multiple channels that are often closely correlated. For example, in EEG signals, different brain regions may exhibit synchronization patterns, indicating real functional connectivity in the brain [32]; in ECG signals, different leads can provide complementary information about the cardiac electrical activity [4]. This motivates us to construct explicit structures to model channel-level dynamics and temporal dynamics for medical time series. To this end, we propose a novel Multi-resolution Adaptive Graph Structure Learning approach to model different levels of dynamics. This approach mainly consists of two main steps: 1) learning multi-resolution embeddings, and 2) constructing multi-resolution graphs based on the learned embeddings.

4.1.1 Multi-scale Embedding Learning. Given a medical time series sample $X_{P_n}^i \in \mathbb{R}^{T \times C}$, we aim to learn multi-scale temporal embeddings that capture the local patterns and dynamics at different time scales. To achieve this, we employ a set of 1-d convolution

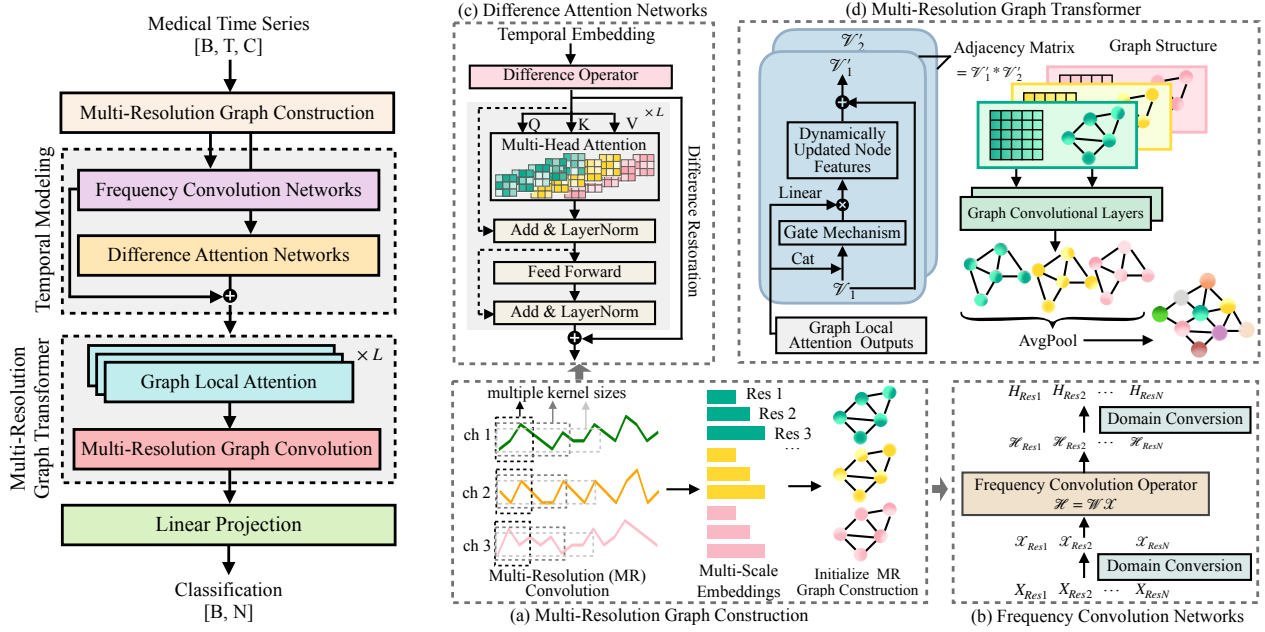


Figure 1: The overall architecture of MedGNN. (a) Multi-resolution graph construction is utilized to learn the dynamic spatiotemporal representations. (b) Frequency convolution networks are applied to provide a multi-view perspective of the temporal dynamics by applying convolutions in the frequency domain. (c) Difference attention networks are employed to capture key temporal patterns while mitigating the impact of baseline wander. (d) Multi-resolution graph transformer is leveraged to model the dynamic spatial dependencies and fuse the information from different resolutions.

networks for different resolutions. Let K_1, \dots, K_M denote a series of M kernel sizes, where each kernel $K_m \in \mathbb{R}^{k_m \times 1}$ has a size of k_m along the time dimension for different channels. The multi-scale embeddings are obtained by applying the 1-d convolution operations with different kernel sizes, formally by:

$$Z^{i(m)} = \text{Conv1D}(X_{P_n}^i, K_m), \quad m = 1, \dots, M, \quad (2)$$

where $Z^{i(m)} \in \mathbb{R}^{T_m \times C}$ represents the learned embeddings at the m -th resolution for i -th sample where we drop P_n for brevity, $T_m = \lceil T/k_m \rceil$ is the size of transformed temporal embeddings, and C is the number of channels.

4.1.2 Multi-resolution Graph Construction. After obtaining the multi-resolution embeddings, we then aim to construct multi-resolution graphs for structure learning. Specifically, we create a series graphs $\{\mathcal{G}^{i(1)}, \dots, \mathcal{G}^{i(M)}\}$, where each graph $\mathcal{G}^{i(m)} = (A^{i(m)}, X^{i(m)})$ initialized as a fully-connected graph corresponds to a specific resolution. $A^{i(m)} \in \mathbb{R}^{C \times C}$ represents the adjacency matrix, and for each graph $\mathcal{G}^{i(m)}$, the node set $\mathcal{V}^{i(m)}$ consists of C nodes, where each node $v_c^{i(m)} \in \mathcal{V}^{i(m)}$ corresponds to a channel in the original series. The node feature matrix $X^{i(m)} \in \mathbb{R}^{C \times T_m}$ is formed by the learned embeddings at the m -th resolution:

$$X^{i(m)} = (Z^{i(m)})^\top. \quad (3)$$

The edge set $\mathcal{E}^{i(m)}$ represents the dependencies and correlations among the channels. We initialize the adjacency matrix $A^{i(m)} \in \mathbb{R}^{C \times C}$ as a learnable matrix to capture the edge weights between pairs of nodes. The edge weights can be learned adaptively to reflect the dynamic channel correlations at different resolutions.

By constructing multi-resolution graphs, we could obtain a hierarchical representation that captures both channel-level and temporal dynamics at different time scales for the medical time series. This rich representation allows for modeling the complex local and longer contextual spatiotemporal patterns in medical time series.

4.2 Difference Attention Networks for the Baseline Wander in Temporal Dynamics

For the physiological time series in the medical domain, the baseline wander [26] - the constant offsets or slow drifts towards baseline measurements - is a common artifact in ECG and EEG recordings. This could make the model capture the less meaningful patterns, i.e., the slow fluctuations around the baseline that may be caused by accidental factors such as patient movement or respiration. To address this problem, we further propose a new architecture, Difference Attention Networks, to learn the temporal dynamics of medical time series in a more focused fashion. The basic idea is to incorporate the concept of *finite difference* into the attention mechanisms. Note that the difference operates on the temporal dimension of the input data.

Specifically, instead of directly applying self-attention to the node features, we compute the differences along the temporal dimension and apply self-attention to these differences. We first add paddings to the temporal embedding dimensions to get $X'^{i(m)} \in \mathbb{R}^{C \times T_m+1}$ obtained from the adaptive graph learning module at the m -th resolution, we first compute the first-order finite difference along the temporal dimension:

$$D^{i(m)}(t) = X'^{i(m)}(t+1) - X'^{i(m)}(t), \quad t = 1, \dots, T_m, \quad (4)$$

where $D^{i(m)} \in \mathbb{R}^C$ represents the first-order difference of series representation at time step t , and $X^{i(m)}(t)$ represents the value of $X^{i(m)}$ at t . We then apply the multi-head self-attention mechanism to the difference representations $D^{i(m)}$ to learn the temporal dependencies. The self-attention is calculated as:

$$\text{Attn}^{i(m)} = \text{Softmax} \left(\frac{D^{i(m)} W_Q^{i(m)} (D^{i(m)} W_K^{i(m)})^\top}{\sqrt{d}} \right) D^{i(m)} W_V^{i(m)}, \quad (5)$$

where $W_Q^{i(m)}, W_K^{i(m)}$ are learnable weight matrices for queries and keys. The outputs of difference self-attention are as:

$$X_{\text{DSA}}^{i(m)} = \text{DifferenceAttention}(X^{i(m)}) = \text{Linear}_{\text{DA}}(\text{Attn}^{i(m)}), \quad (6)$$

where $\text{Linear}_{\text{DA}}(\cdot)$ represents the final linear layers for output, and $W_V^{i(m)}$ are learnable weight matrices for values in self-attention. The final outputs of Difference Attention Networks are formed as:

$$X_{\text{DA}}^{i(m)} = X_{\text{DSA}}^{i(m)} + X^{i(m)}. \quad (7)$$

The Difference Attention Networks provide a novel way to focus on the temporal changes in medical time series, reducing the influence of slow baseline drifts and highlighting the meaningful patterns.

4.3 Frequency Convolution Networks for the Multi-view Temporal Representations

Though the difference attention network can help address the baseline wander problem, the processed representations are in the “difference space” and it might lose some information in the original data space. To better capture the multi-view information of medical time series signals, we further introduce the frequency convolution networks to enhance the temporal representations [10, 11, 45, 47]. Note that the temporal convolution networks are operated in parallel with Difference Attention Networks for multi-view information.

Specifically, given the node feature matrix $X^{i(m)} \in \mathbb{R}^{C \times T_m}$ obtained from the multi-resolution graph construction at the m -th resolution, we first apply the Fourier transform to convert the temporal signals from the time domain to the frequency domain:

$$\mathcal{X}^{i(m)} = \mathcal{F}(X^{i(m)}) = \int_{-\infty}^{\infty} X^{i(m)}(t) e^{-j2\pi ft} dt, \quad (8)$$

where \mathcal{F} denotes the Fourier transform, f is the frequency variable, t is the integral variable, and j is the imaginary unit. The resulting $\mathcal{X}^{i(m)} \in \mathbb{C}^{C \times S}$ represents the frequency domain representation, where S is the number of frequency components. Next, we apply Fourier convolution layers to the frequency domain representations to capture the dependencies and patterns in the frequency space:

$$\mathcal{H}^{i(m)} = \text{FourierConvolution}(\mathcal{X}^{i(m)}) = \mathcal{X}^{i(m)} \odot \mathcal{W}^{i(m)}, \quad (9)$$

where $\mathcal{W}^{i(m)} \in \mathbb{C}^{C \times S}$ represents the learnable convolution kernels in the frequency domain, and \odot denotes the element-wise multiplication. The resulting $\mathcal{H}^{i(m)} \in \mathbb{C}^{C \times S}$ represents the convolved frequency domain representations. Finally, we apply the inverse Fourier transform to recover the temporal representations from the frequency domain back to the time domain:

$$X_{\text{FC}}^{i(m)} = \mathcal{F}^{-1}(\mathcal{H}^{i(m)}) = \int_{-\infty}^{\infty} \mathcal{H}^{i(m)}(f) e^{j2\pi ft} df, \quad (10)$$

where \mathcal{F}^{-1} denotes the inverse Fourier transform, and $X_{\text{FC}}^{i(m)} \in \mathbb{R}^{C \times T_m}$ represents the recovered temporal representations. The frequency convolution networks provide a complementary view of the temporal dynamics by operating in the frequency domain.

4.4 Learning Spatial Dynamics with Multi-resolution Graph Transformer

After obtaining the temporal representations from the Difference Attention Networks $X_{\text{DA}}^{i(m)}$ and the Frequency Convolution Networks $X_{\text{FC}}^{i(m)}$ at each resolution m , we aim to further capture the inter-series (spatial) dependencies based on the explicit structures of multi-resolution graphs. For this aim, we propose the multi-resolution graph transformer model for spatial dependency learning. Specifically, first, for each resolution, we sum the two learned temporal representations $X_{\text{DA}}^{i(m)}$ and $X_{\text{FC}}^{i(m)}$ to obtain the fused multi-view representations at the m -th resolution:

$$X_{\text{fused}}^{i(m)} = X_{\text{DA}}^{i(m)} + X_{\text{FC}}^{i(m)}. \quad (11)$$

Next, inspired by Graph Transformer [31], we apply the local attention mechanisms first and then pass the representations to the graph neural networks for processing. Specifically, the local attention for graph processing is first computed as:

$$\alpha_{pq} = \frac{\exp(g(\mathbf{x}_p, \mathbf{x}_q) \cdot b_{pq})}{\sum_{k \in \mathcal{N}(v_p)} \exp(g(\mathbf{x}_p, \mathbf{x}_k) \cdot b_{pk})}, \quad (12)$$

where α_{pq} is the attention score, \mathbf{x}_p and \mathbf{x}_q are their node features, g is a function that computes the similarity between two nodes, i.e., the dot-product. b_{pq} is a local attention bias term. Note that the neighborhood is dynamically learned in Section 4.1. Then we can calculate the graph representations by:

$$X_{\text{GA}}^{i(m)} = \text{LocalAttention}(X_{\text{fused}}^{i(m)}) = \alpha X_{\text{fused}}^{i(m)}, \quad (13)$$

where $X_{\text{GA}}^{i(m)} \in \mathbb{R}^{C \times T_m}$ represents the spatially attended representations. The multi-head self-attention allows the model to attend to different spatial locations and capture the dependencies among different time series channels. After obtaining the spatially attended representations, we apply graph convolution networks to further incorporate the graph structure information at each resolution. Given the constructed graph $\mathcal{G}^{i(m)} = (A^{i(m)}, X_{\text{GA}}^{i(m)})$ at the m -th resolution, we perform graph convolution as follows:

$$X_{\text{GT}}^{i(m)} = \text{GraphConv}(\hat{A}^{i(m)}, X_{\text{GA}}^{i(m)}, W_{\text{GT}}^{i(m)}), \quad (14)$$

where $\hat{A}^{i(m)} = \tilde{D}^{-\frac{1}{2}} \tilde{A}^{i(m)} \tilde{D}^{-\frac{1}{2}}$ is the normalized adjacency matrix, $\tilde{A}^{i(m)} = A^{i(m)} + I$ is the adjacency matrix with self-loops, I is the identity matrix, \tilde{D} is the diagonal degree matrix of $\tilde{A}^{i(m)}$, $W_{\text{GT}}^{i(m)} \in \mathbb{R}^{T_m \times T_m}$ is the learnable weight matrix for graph convolution. The resulting $X_{\text{GT}}^{i(m)} \in \mathbb{R}^{C \times T_m}$ represents the graph convoluted representations at the m -th resolution. Since we have multiple resolutions, we need to fuse the representations obtained from different resolutions to capture the multi-resolution dynamics. We achieve this by applying average pooling across the resolution dimension:

$$X_{\text{MRFused}}^i = \text{AvgPool}(X_{\text{GT}}^{i(1)}, \dots, X_{\text{GT}}^{i(M)}), \quad (15)$$

Table 1: Results of sample-based evaluation on ADFD dataset (3-classes). The best results are in red and the second are in blue.

Models	MedGNN	Medformer	iTransformer	PatchTST	FEDformer	Crossformer	FourierGNN	CrossGNN	TodyNet	SimTSC
Accuracy	98.42 ± 0.04	97.62 ± 0.17	65.10 ± 0.25	66.39 ± 0.55	78.77 ± 0.91	88.92 ± 0.94	67.27 ± 0.68	55.38 ± 2.44	90.54 ± 0.68	90.96 ± 0.15
Precision	98.31 ± 0.02	97.67 ± 0.13	62.75 ± 0.24	65.28 ± 0.54	77.75 ± 0.85	88.59 ± 0.95	65.26 ± 0.74	53.02 ± 2.74	90.10 ± 0.64	90.72 ± 0.29
Recall	98.29 ± 0.05	97.37 ± 0.24	62.49 ± 0.34	65.29 ± 0.72	77.97 ± 0.89	88.24 ± 1.05	65.08 ± 0.79	50.41 ± 2.91	90.35 ± 0.83	90.54 ± 0.13
F1 score	98.30 ± 0.12	97.51 ± 0.18	62.51 ± 0.36	65.15 ± 0.54	77.82 ± 0.85	88.39 ± 1.00	65.08 ± 0.77	48.64 ± 4.67	90.21 ± 0.72	90.61 ± 0.16
AUROC	99.93 ± 0.11	99.84 ± 0.02	81.63 ± 0.20	83.20 ± 0.30	92.36 ± 0.49	97.40 ± 0.35	83.26 ± 0.46	70.19 ± 2.70	98.04 ± 0.23	98.22 ± 0.11

Table 2: Overall results of subject-based evaluation. The best results are in red and the second best are in blue.

Models		MedGNN	Medformer	iTransformer	PatchTST	FEDformer	Crossformer	FourierGNN	CrossGNN	TodyNet	SimTSC
ADFD	Accuracy	56.12 ± 0.11	53.25 ± 1.10	52.13 ± 1.37	43.68 ± 0.38	46.25 ± 0.62	50.42 ± 1.26	52.00 ± 1.33	47.16 ± 1.69	42.34 ± 2.48	49.26 ± 2.58
	Precision	55.07 ± 0.09	51.30 ± 1.13	46.58 ± 1.08	41.60 ± 0.91	46.33 ± 1.65	45.48 ± 2.13	47.20 ± 0.87	44.74 ± 2.41	42.99 ± 2.51	47.04 ± 0.05
	Recall	55.47 ± 0.34	50.76 ± 0.83	46.96 ± 1.02	40.46 ± 1.25	44.22 ± 1.02	45.80 ± 1.82	48.21 ± 0.95	41.39 ± 1.38	42.48 ± 2.57	42.75 ± 0.30
	F1 score	55.00 ± 0.24	50.75 ± 0.91	46.60 ± 0.90	39.97 ± 1.92	43.75 ± 0.88	45.46 ± 2.03	47.30 ± 0.75	39.76 ± 1.87	41.05 ± 2.03	41.79 ± 1.25
	AUROC	74.68 ± 0.33	70.59 ± 1.31	67.14 ± 0.97	59.34 ± 0.75	62.48 ± 1.45	66.24 ± 1.86	66.18 ± 0.94	58.61 ± 1.56	60.69 ± 2.63	65.10 ± 0.58
APAVA	Accuracy	82.60 ± 0.35	72.66 ± 6.57	75.33 ± 1.42	68.50 ± 1.95	74.87 ± 2.07	75.09 ± 0.71	68.47 ± 7.86	54.40 ± 3.15	72.66 ± 5.70	82.35 ± 0.77
	Precision	87.70 ± 0.22	73.53 ± 8.23	75.32 ± 1.80	77.98 ± 1.96	74.45 ± 1.45	79.64 ± 3.04	67.60 ± 8.76	46.37 ± 9.06	74.74 ± 7.33	85.49 ± 1.68
	Recall	78.93 ± 0.09	70.20 ± 5.46	73.10 ± 1.90	61.94 ± 2.58	73.24 ± 3.32	70.67 ± 0.75	66.10 ± 7.94	50.60 ± 3.19	71.25 ± 4.61	78.69 ± 0.70
	F1 score	80.25 ± 0.16	70.69 ± 5.90	73.52 ± 1.83	59.25 ± 3.79	73.33 ± 3.21	71.06 ± 0.89	66.36 ± 8.13	47.54 ± 6.17	70.91 ± 5.23	79.90 ± 0.77
	AUROC	85.93 ± 0.26	75.52 ± 7.36	85.79 ± 1.79	66.32 ± 3.20	83.59 ± 1.92	79.14 ± 4.86	75.03 ± 11.62	49.59 ± 5.19	78.85 ± 4.70	82.37 ± 2.79
PTB	Accuracy	84.53 ± 0.28	79.76 ± 1.24	83.41 ± 2.04	76.33 ± 1.06	76.90 ± 2.65	81.46 ± 3.62	79.70 ± 3.30	75.92 ± 2.06	77.41 ± 3.65	76.99 ± 0.98
	Precision	87.35 ± 0.45	81.95 ± 0.97	87.88 ± 1.59	79.36 ± 1.63	78.94 ± 3.57	84.99 ± 3.56	81.46 ± 4.58	78.35 ± 1.79	82.50 ± 2.10	80.61 ± 1.70
	Recall	77.90 ± 0.66	71.50 ± 2.08	75.67 ± 3.06	66.05 ± 1.52	67.30 ± 3.86	73.32 ± 5.41	71.53 ± 4.55	65.60 ± 3.22	67.12 ± 6.09	66.92 ± 1.82
	F1 score	80.40 ± 0.62	73.44 ± 2.26	78.22 ± 3.29	67.16 ± 1.89	68.52 ± 4.56	75.34 ± 5.84	73.33 ± 5.19	66.49 ± 3.81	67.81 ± 7.62	68.19 ± 2.21
	AUROC	93.31 ± 0.46	93.13 ± 0.50	90.61 ± 2.46	88.48 ± 0.85	86.39 ± 2.86	89.71 ± 3.71	84.93 ± 3.44	89.93 ± 0.47	91.72 ± 1.55	88.60 ± 0.66
PTB-XL	Accuracy	73.87 ± 0.18	72.87 ± 0.23	69.13 ± 0.21	73.15 ± 0.24	57.23 ± 9.48	73.23 ± 0.18	63.49 ± 0.88	63.09 ± 0.63	72.21 ± 0.65	71.87 ± 0.91
	Precision	66.26 ± 0.29	64.24 ± 0.42	59.46 ± 0.44	65.47 ± 0.50	52.32 ± 6.38	64.92 ± 0.58	52.05 ± 1.21	49.60 ± 0.93	64.30 ± 0.68	63.61 ± 1.73
	Recall	61.13 ± 0.23	60.09 ± 0.30	54.56 ± 0.23	60.64 ± 0.44	48.76 ± 7.21	61.13 ± 0.45	47.85 ± 1.13	44.94 ± 0.65	57.12 ± 1.66	58.86 ± 0.99
	F1 score	62.54 ± 0.20	61.70 ± 0.23	56.24 ± 0.24	62.46 ± 0.28	47.75 ± 8.36	62.51 ± 0.31	48.92 ± 1.19	45.08 ± 0.44	58.94 ± 1.75	60.19 ± 0.60
	AUROC	90.21 ± 0.15	89.64 ± 0.19	86.67 ± 0.18	89.61 ± 0.15	82.15 ± 4.25	89.96 ± 0.16	82.45 ± 0.86	81.43 ± 0.33	89.14 ± 0.44	88.19 ± 0.78
TDBRAIN	Accuracy	91.04 ± 0.09	88.71 ± 1.26	74.87 ± 1.30	73.90 ± 4.40	77.56 ± 1.67	82.48 ± 1.41	76.04 ± 3.14	70.60 ± 3.31	89.58 ± 2.15	89.06 ± 1.67
	Precision	91.15 ± 0.12	88.84 ± 1.13	74.99 ± 1.33	74.21 ± 4.26	78.00 ± 1.84	82.71 ± 1.21	76.22 ± 3.22	70.71 ± 3.21	90.21 ± 1.39	89.41 ± 0.86
	Recall	91.04 ± 0.20	88.71 ± 1.26	74.87 ± 1.30	73.90 ± 4.40	77.56 ± 1.67	82.48 ± 1.41	76.04 ± 3.14	70.60 ± 3.31	89.58 ± 3.87	89.06 ± 0.50
	F1 score	91.04 ± 0.08	88.70 ± 1.27	74.85 ± 1.29	73.79 ± 4.48	77.48 ± 1.65	82.44 ± 1.44	76.00 ± 3.14	70.55 ± 3.38	89.54 ± 4.69	89.04 ± 1.01
	AUROC	96.74 ± 0.04	96.24 ± 0.59	83.66 ± 1.14	80.93 ± 6.12	86.41 ± 1.45	91.58 ± 0.88	84.11 ± 2.83	78.42 ± 3.43	97.41 ± 1.00	97.28 ± 1.69

where AvgPool(\cdot) denotes the average pooling operation, and the output $X_{MRFused}^i \in \mathbb{R}^{C \times T}$ represents the fused multi-resolution representations. Finally, we feed the fused multi-resolution representations $X_{MRFused}^i$ into a linear or fully-connected layer followed by a softmax activation function for classification:

$$\hat{y}^i = \text{Softmax}(\text{Linear}_{GT}(X_{MRFused}^i)), \quad (16)$$

where $\text{Linear}_{GT}(\cdot)$ is a learnable linear layer that maps the fused representations to the output space, and $\hat{y}^i \in \mathbb{R}^K$ represents the predicted probabilities for the K classes.

5 Experiments

In this section, we perform extensively experiments with five real-world medical time series benchmarks to assess the performance of our proposed MedGNN. Furthermore, we conduct thorough analytical experiments and visualization studies concerning the different components of the MedGNN framework.

5.1 Experimental Settings

5.1.1 Datasets. We conduct empirical analyses on five representative medical datasets, i.e., ADFD [24], APAVA [8], TDBRAIN [34], PTB [29], and PTB-XL [36]. These datasets include three EEG

datasets and two ECG datasets. The data preprocessing and train-validation-test split are following the previous work [39]. For further details on the datasets, please refer to the Appendix A.1.

5.1.2 Baselines. We compare our proposed MedGNN with the representative and state-of-the-art models for time series classification. We choose the baseline methods from two categories: (1) GNN-based models, which include TodyNet [22], SimTSC [51], FourierGNN [46], and CrossGNN [13]; (2) Transformer-based models, including iTransformer [23], PatchTST [27], FEDformer [54], Crossformer [53], Autoformer [41] and more recent Medformer [39]. Further details about the baselines can be found in Appendix A.2.

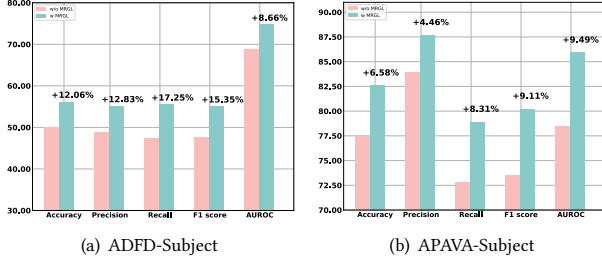
5.1.3 Implementation Details. All experiments are implemented using Pytorch 2.2 [28] and conducted on 4 GeForce RTX 4090 GPUs. We employ cross-entropy loss as the loss function and present five metrics: accuracy, precision, recall, F1 score, and AUROC. Further implementation details are presented in Appendix A.3 and A.4.

5.2 Main Results

We present results of our proposed MedGNN compared to several representative baselines with two different experimental setup (e.g., sample-based and subject-based evaluation) in Tables 1 and 2. Note that the percentage symbol (%) is omitted in the experimental results, and this will not be repeated below.

Table 3: Ablation studies on the effects of Difference Attention (DA) under the subject-based setup.

Datasets	APAVA				TDBRAIN				PTB-XL			
Metrics	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
w DA	82.60	87.70	78.93	80.25	91.04	91.15	91.04	91.04	73.87	66.26	61.13	62.54
w/o DA		69.88	69.94	66.20	66.38	70.52	70.86	70.52	70.40	69.33	60.05	56.07
Improvement	18.20%	25.40%	19.24%	20.89%	29.10%	28.63%	29.10%	29.31%	6.54%	10.35%	9.02%	8.52%

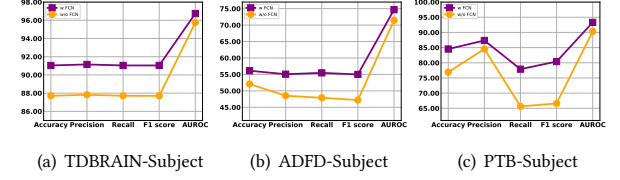
**Figure 2: Ablation study of Multi-Resolution Graph Learning (MRGL) under the Subject-based setup.**

5.2.1 Sample-based Evaluation. In this setup, the training, validation, and test sets are divided based on test samples. Samples from different subjects are randomly shuffled and then assigned to the corresponding sets. It is a reasonable setting in real-world medical scenarios because many patients return for multiple visits even though it might have the information leakage problem from the perspective deep learning. Since this setting is easier than the subject-based evaluation, we only assess the ADFD dataset to enable a simple comparison for reference. Table 1 has shown the overall results sample-based evaluation of MedGNN compared with several baseline algorithms. We can easily observe that our proposed MedGNN outperforms all the baselines as illustrated in and achieves the best performance in five different metrics.

5.2.2 Subject-based Evaluation. The training, validation, and test sets are split based on subjects (patients) under this setup. Each subject, along with their corresponding samples, is assigned to one of the three sets (training, validation, or test) according to a predefined ratio or subject IDs. This ensures that samples from the same subject only appear in one set, preventing overlap between the sets. Table 2 presents the results on five datasets under the subject-based setup. Overall, our model achieves leading performance on most datasets, achieving 22 top-1 and 2 top-2 out of 25 in total across five datasets. Notably, MedGNN ranks first in terms of F1 score across all five datasets, demonstrating that MedGNN is not only accurate but also achieves a good balance between precision and recall, indicating its exceptional robustness in handling classification tasks. This balance is particularly important in critical applications such as medical diagnosis, where both false positives (misclassifying healthy individuals as sick) and false negatives (misclassifying sick individuals as healthy) can have serious consequences.

5.3 Ablation Studies

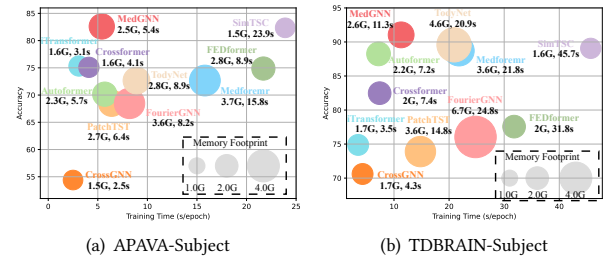
5.3.1 Study of the Multi-Resolution Graph Learning. We explore the impact of multi-resolution graph learning (MRGL) on MedGNN's performance across various metrics. To assess its effectiveness, we compare two versions of the model: one with MRGL enabled and one without it. The MRGL-enhanced version demonstrates substantial performance gains as shown in Figure 2, particularly

**Figure 3: Ablation study of Frequency Convolution Networks (FCN) under the Subject-based setup.**

in recall and F1 score, indicating the model's improved ability to detect and balance positive predictions with accurate classifications. This improvement highlights MRGL's capacity to capture multi-resolution dependencies effectively, improving the model's robustness in handling complex medical time series.

5.3.2 Study of the Frequency Convolution Networks. To evaluate the impact of the Frequency Convolution Networks (FCN) within MedGNN, we perform ablation experiments, as shown in Figure 3, where "w/o FCN" denotes the version without FCN. The results clearly demonstrate that incorporating FCN improves MedGNN's performance across all metrics on the experimental datasets. Moreover, frequency convolution helps prevent potential information loss from relying solely on difference attention, further enhancing the temporal representation.

5.3.3 Study of the Difference Attention Networks. In this section, we aim to investigate the effectiveness of the difference attention in the MedGNN framework. Table 3 has shown the performance comparison of the variant on two EEG datasets (APAVA, TDBRAIN) and one ECG dataset. The version equipped with difference attention achieves improvements of 23.65%, 27.02%, 24.17%, and 25.10% in the metrics of Accuracy, Precision, Recall, and F1 score on EEG datasets in average, while 6.54%, 10.35%, 9.02%, and 8.52% respectively on ECG. This indicates that difference attention can enhance the model's performance in medical time series classification tasks by minimizing the impact of slow baseline drifts and bringing important patterns to the forefront.

**Figure 4: Model effectiveness and efficiency comparison on two datasets under the Subject-based setup.**

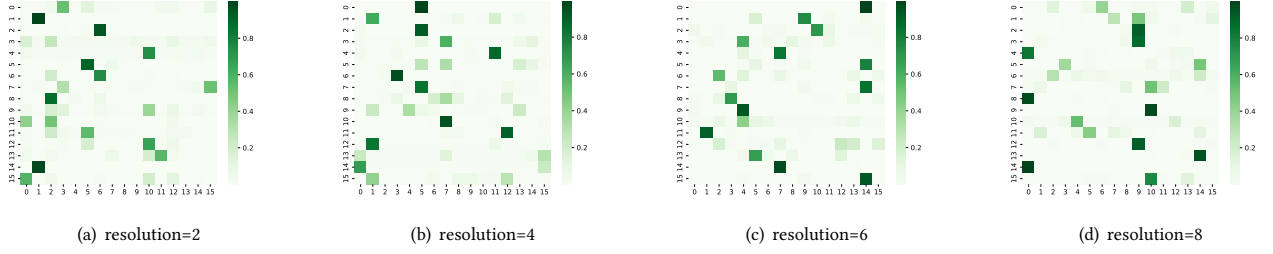


Figure 5: Adjacent matrices of multi-resolution graphs learned from APAVA dataset.

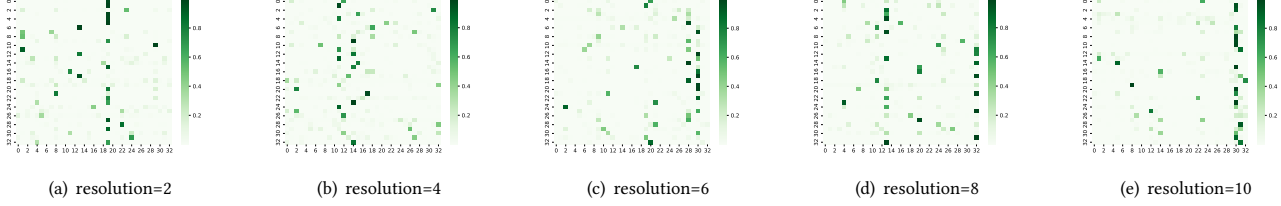


Figure 6: Adjacent matrices of multi-resolution graphs learned from TDBRAIN dataset.

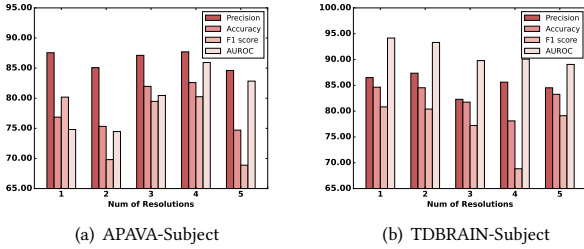


Figure 7: Effect of different numbers of resolutions.

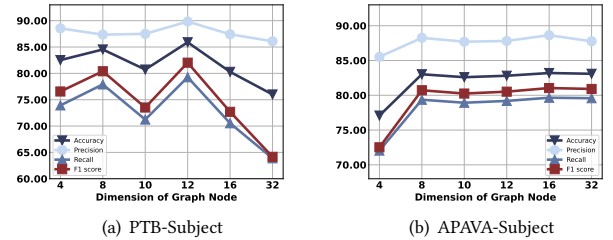


Figure 8: Effect of different graph node dimension.

5.4 Additional Experiments

5.4.1 Efficiency Analysis. We comprehensively evaluate the model efficiency from three aspects: classification performance (Accuracy), training speed, and memory footprint. Specifically, we choose two different sizes of datasets: the APAVA (23 subjects, 5,967 16-channel samples) and TDBRAIN (72 subjects, 6,240 33-channel samples) datasets. The closer a marker is to the upper-left corner of Figure 4, the higher the model’s accuracy and the faster its training speed. Additionally, the smaller the marker’s area, the lower the memory usage during training. Therefore, we can conclude that although MedGNN’s training time and memory footprint are at a moderate level among all baselines, its classification performance is the best.

5.4.2 Visualizations. Figures 5 and Figure 6 present visualizations of the learned adjacency matrices across different resolutions, providing insights into what the model captures at each resolution. Overall, the matrices tend to be sparse, indicating that MedGNN focuses on learning meaningful correlations from the data rather than relying on superficial variable aggregation. We also find that the weight distributions of adjacency matrices vary across different resolutions within the same dataset, indicating that the relationships between variables learned through graph learning differ at each resolution. This is significant for practical applications, as the relationships between variables are often not fixed but instead vary depending on the scale or context.

5.4.3 Study of Numbers of Resolutions. Figure 7 shows the effect of the number of resolutions on classification performance. It suggests that appropriately increasing the number of resolutions has the potential to improve performance across various metrics. However, introducing overly coarse resolutions, where the receptive field becomes too large, may harm performance because excessive aggregation can lead to the loss of important fine-grained information and reduce the model’s ability to capture subtle but critical patterns.

5.4.4 Study of the Size of Graph Node Dimensions. Figure 8 displays the effect of different graph node dimensions in multi-resolution graph learning. It indicates that larger graph node dimensions do not necessarily lead to better performance. This is understandable given that we employ a multi-resolution graph learning strategy, meaning the learning burden at each resolution is relatively light, making smaller node dimensions sufficient. In contrast, larger graph node dimensions may introduce data sparsity issues and increase computational overhead, hindering the model’s learning ability.

6 Conclusion

In this paper, we have proposed a Multi-resolution Spatiotemporal Graph Learning framework, *MedGNN*, for medical time series classification. We have constructed multi-resolution adaptive graph structures to learn dynamic multi-scale embeddings. Based on the graph structure, we have proposed two types of networks, i.e., Difference Attention Networks and Frequency Convolution Networks, to address the baseline wander problem in medical time series

and learn the multi-perspective information for temporal modeling. We have also adapted Multi-resolution Graph Transformer for the dynamic spatial learning and information fusion. Extensive experiments have shown the superiority of our methods in different settings. We hope this could facilitate more future works on medical time series and healthcare applications.

7 Acknowledgments

This work was partially supported by National Natural Science Foundation of China (No. 62402531) and National Natural Science Foundation of China (No. 623B2043).

References

- [1] Wasyihun Sema Admass and Girmaw Andualem Bogale. 2024. Arrhythmia classification using ECG signal: A meta-heuristic improvement of optimal weighted feature integration and attention-based hybrid deep learning model. *Biomed. Signal Process. Control* 87, Part B (2024), 105565.
- [2] Anthony J. Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn J. Keogh. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Discov.* 31, 3 (2017), 606–660.
- [3] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. In *NeurIPS*.
- [4] Selcan Kaplan Berkaya, Alper Kursat Uysal, Efnan Sora Gunal, Semih Ergin, Serkan Gunal, and M Bilginer Gulmezoglu. 2018. A survey on ECG analysis. *Biomedical Signal Processing and Control* 43 (2018), 216–235.
- [5] Manuel Blanco-Velasco, Binwei Weng, and Kenneth E Barner. 2008. ECG signal denoising and baseline wander correction based on the empirical mode decomposition. *Computers in biology and medicine* 38, 1 (2008), 1–13.
- [6] Aaron Bostrom and Anthony J. Bagnall. 2017. Binary Shapelet Transform for Multiclass Time Series Classification. *Trans. Large Scale Data Knowl. Centered Syst.* 32 (2017), 24–46.
- [7] Michael X Cohen. 2017. Where does EEG come from and what does it mean? *Trends in neurosciences* 40, 4 (2017), 208–218.
- [8] J Escudero, Daniel Abásolo, Roberto Hornero, Pedro Espino, and Miguel López. 2006. Analysis of electroencephalograms in Alzheimer's disease patients with multiscale entropy. *Physiological measurement* 27, 11 (2006), 1091.
- [9] Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. 2023. Dish-TS: A General Paradigm for Alleviating Distribution Shift in Time Series Forecasting. In *AAAI*. AAAI Press, 7522–7529.
- [10] Wei Fan, Kun Yi, Hangting Ye, Zhiyuan Ning, Qi Zhang, and Ning An. 2024. Deep frequency derivative learning for non-stationary time series forecasting. *arXiv preprint arXiv:2407.00502* (2024).
- [11] Wei Fan, Shun Zheng, Xiaohan Yi, Wei Cao, Yanjie Fu, Jiang Bian, and Tie-Yan Liu. 2022. DEPTS: Deep Expansion Learning for Periodic Time Series Forecasting. In *ICLR*. OpenReview.net.
- [12] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. *Data Min. Knowl. Discov.* 33, 4 (2019), 917–963.
- [13] Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. 2023. Crossgnn: Confronting noisy multivariate time series via cross interaction refinement. *Advances in Neural Information Processing Systems* 36 (2023), 46885–46902.
- [14] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. *Data mining and knowledge discovery* 33, 4 (2019), 917–963.
- [15] Mahboobeh Jafari, Afshin Shoeibi, Marjane Khodatars, Sara Bagherzadeh, Ahmad Shalhaf, David López-García, Juan Manuel Górriz, and U. Rajendra Acharya. 2023. Emotion recognition in EEG signals using deep learning methods: A review. *Comput. Biol. Medicine* 165 (2023), 107450.
- [16] 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. *CoRR abs/2307.03759* (2023).
- [17] Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [18] Dani Kiyasseh, Tingting Zhu, and David A. Clifton. 2021. CLOCS: Contrastive Learning of Cardiac Signals Across Space, Time, and Patients. In *ICML (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 5606–5615.
- [19] Vernon J. Lawhern, Amelia J. Solon, Nicholas R. Waytowich, Stephen M. Gordon, Chou P. Hung, and Brent J. Lance. 2016. EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces. *CoRR abs/1611.08024* (2016).
- [20] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR (Poster)*.
- [21] Jason Lines and Anthony J. Bagnall. 2015. Time series classification with ensembles of elastic distance measures. *Data Min. Knowl. Discov.* 29, 3 (2015), 565–592.
- [22] Huaiyuan Liu, Donghua Yang, Xianzhang Liu, Xinglei Chen, Zhiyu Liang, Hongzhi Wang, Yong Cui, and Jun Gu. 2024. Todynnet: temporal dynamic graph neural network for multivariate time series classification. *Information Sciences* (2024), 120914.
- [23] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024. itransformer: Inverted transformers are effective for time series forecasting. *International Conference on Learning Representations* (2024).
- [24] Andreas Miliadous, Katerina D Tzimourta, Theodora Afrantou, Panagiotis Ioannidis, Nikolaos Grigoriadis, Dimitrios G Tsilikakis, Pantelis Angelidis, Markos G Tsipouras, Euripidis Glavas, Nikolaos Giannakeas, et al. 2023. A dataset of scalp EEG recordings of Alzheimer's disease, frontotemporal dementia and healthy subjects from routine EEG. *Data* 8, 6 (2023), 95.
- [25] Mohammad Amin Morid, Olivia R Liu Sheng, and Joseph Dunbar. 2023. Time series prediction using deep learning methods in healthcare. *ACM Transactions on Management Information Systems* 14, 1 (2023), 1–29.
- [26] Valentin Moy, Mohammed Sidi Yakoub, Sid-Ahmed Selouani, Yasser F. Alharbi, and Yousef Ajami Alotaibi. 2021. Classification of the Neural Correlates of Mind Wandering States in EEG Signals. In *TSP*. IEEE, 126–129.
- [27] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A time series is worth 64 words: Long-term forecasting with transformers. *ICLR* (2023).
- [28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).
- [29] PhysioToolkit PhysioBank. 2000. Physionet: components of a new research resource for complex physiologic signals. *Circulation* 101, 23 (2000), e215–e220.
- [30] Andrea L. Schaffer, Timothy A. Dobbins, and Sallie-Anne Pearson. 2021. Interrupted time series analysis using autoregressive integrated moving average (ARIMA) models: a guide for evaluating large-scale health interventions. *BMC medical research methodology* 21 (2021), 1–12.
- [31] Ahsan Shehzad, Feng Xia, Shagufta Abid, Ciyuan Peng, Shuo Yu, Dongyu Zhang, and Karin Verspoor. 2024. Graph Transformers: A Survey. *CoRR abs/2407.09777* (2024).
- [32] Michal Teplan et al. 2002. Fundamentals of EEG measurement. *Measurement science review* 2, 2 (2002), 1–11.
- [33] Simon L. Turner, Amalia Karahalios, Andrew B. Forbes, Monica Taljaard, Jeremy M. Grimshaw, Allen C. Cheng, Lisa Bero, and Joanne E. McKenzie. 2020. Design characteristics and statistical methods used in interrupted time series studies evaluating public health interventions: a review. *Journal of Clinical Epidemiology* 122 (2020), 1–11.
- [34] Hanneke van Dijk, Guido van Wingen, Damiaan Denys, Sebastian Olbrich, Rosalinde van Ruth, and Martijn Arns. 2022. The two decades brainclinics research archive for insights in neurophysiology (TDBRAIN) database. *Scientific data* 9, 1 (2022), 333.
- [35] Thomas Vincent, Laurent Risser, and Philippe Ciuciu. 2010. Spatially adaptive mixture modeling for analysis of fMRI time series. *IEEE transactions on medical imaging* 29, 4 (2010), 1059–1074.
- [36] Patrick Wagner, Nils Strodthoff, Ralf-Dieter Boussejot, Dieter Kreiseler, Fatima I. Lunze, Wojciech Samek, and Tobias Schaeffer. 2020. PTB-XL, a large publicly available electrocardiography dataset. *Scientific data* 7, 1 (2020), 1–15.
- [37] Yihe Wang, Yu Han, Haishuai Wang, and Xiang Zhang. 2023. Contrast Everything: A Hierarchical Contrastive Framework for Medical Time-Series. In *NeurIPS*.
- [38] Yihe Wang, Nan Huang, Taida Li, Yujun Yan, and Xiang Zhang. 2024. Medformer: A Multi-Granularity Patching Transformer for Medical Time-Series Classification. *CoRR abs/2405.19363* (2024).
- [39] Yihe Wang, Nan Huang, Taida Li, Yujun Yan, and Xiang Zhang. 2024. Medformer: A Multi-Granularity Patching Transformer for Medical Time-Series Classification. *arXiv preprint arXiv:2405.19363* (2024).
- [40] Zekai Wang, Stavros Stavrou, and Bing Yao. 2023. Hierarchical deep learning with Generative Adversarial Network for automatic cardiac diagnosis from ECG signals. *Comput. Biol. Medicine* 155 (2023), 106641.
- [41] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems* 34 (2021), 22419–22430.
- [42] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *KDD*. 753–763.
- [43] Chaoyi Yang, M. Brandon Westover, and Jimeng Sun. 2023. ManyDG: Many-domain Generalization for Healthcare Applications. In *ICLR*. OpenReview.net.
- [44] Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. 2024. FilterNet: Harnessing Frequency Filters for Time Series Forecasting. *arXiv preprint arXiv:2411.01623* (2024).

- [45] Kun Yi, Qi Zhang, Longbing Cao, Shoujin Wang, Guodong Long, Liang Hu, Hui He, Zhendong Niu, Wei Fan, and Hui Xiong. 2023. A Survey on Deep Learning based Time Series Analysis with Frequency Transformation. *CoRR* abs/2302.02173 (2023).
- [46] Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. 2024. FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective. *Advances in Neural Information Processing Systems* 36 (2024).
- [47] Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. 2024. Frequency-domain MLPs are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems* 36 (2024).
- [48] Raneen Younis, Abdul Hakmeh, and Zahra Ahmadi. 2024. MTS2Graph: Interpretable multivariate time series classification with temporal evolving graphs. *Pattern Recognit.* 152 (2024), 110486.
- [49] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *IJCAI*. 3634–3640.
- [50] Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou, and Xia Hu. 2022. Towards Similarity-Aware Time-Series Classification. In *SDM*. SIAM, 199–207.
- [51] Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou, and Xia Hu. 2022. Towards similarity-aware time-series classification. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. SIAM, 199–207.
- [52] Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. 2022. Graph-Guided Network for Irregularly Sampled Multivariate Time Series. In *ICLR*. OpenReview.net.
- [53] Yunhao Zhang and Junchi Yan. 2022. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*.
- [54] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*. PMLR, 27268–27286.

A Experimental Details

A.1 Datasets

The ADFD [24] dataset includes 88 subjects and contains 69,762 multivariate EEG samples, each recorded from 19 channels. Every sample represents a one-second time sequence with 256 time points, captured at a sampling rate of 256Hz. Each sample is labeled with one of three classes, indicating whether the subject is Healthy, has Dementia, or has Alzheimer’s disease. The APAVA [8] dataset includes 23 subjects and contains 5,967 multivariate EEG samples, each recorded across 16 channels. Each sample represents a one-second time sequence with 256 time points, recorded at a sampling rate of 256Hz. Every sample is accompanied by a binary label indicating whether the subject has been diagnosed with Alzheimer’s disease. The TDBRAIN [34] dataset consists of 72 subjects and provides 6,240 EEG samples, each recorded across 33 channels. These samples represent one-second time sequences, containing 256 data points, recorded at a frequency of 256Hz. Each sample is tagged with a binary label that indicates whether the subject has Parkinson’s disease. The PTB [29] dataset consists of 198 subjects and includes 64,356 multivariate ECG samples, each recorded across 15 channels. Each sample corresponds to a heartbeat represented by 300 time points, recorded at a sampling rate of 250Hz. A binary label accompanies each sample, indicating whether the subject has experienced a Myocardial Infarction. The PTB-XL [36] dataset comprises 17,596 subjects and includes 191,400 multivariate ECG samples, each recorded across 12 channels. Each sample represents a one-second time sequence with 250 time points, captured at a sampling rate of 250Hz. Each sample is labeled with one of five classes, representing different heart conditions.

A.2 Baselines

We choose ten well-acknowledged and state-of-the-art models for comparison to evaluate the effectiveness of our proposed MedGNN for medical time series classification, including GNN-based models and Transformer-based models. We introduce these models as follows:

Medformer [39] introduces three innovative mechanisms to utilize the distinctive properties of medical time series. These include cross-channel patching for learning multi-timestamp and multi-channel features, multi-granularity embedding for capturing features at various scales, and a two-stage multi-granularity self-attention mechanism to capture features both within and across granularities. The official implementation is available at <https://github.com/DL4mHealth/Medformer>.

iTransformer [23] reverses the structure of the Transformer by encoding individual series into variate tokens, allowing the attention mechanism to capture multivariate correlations, while applying feed-forward networks to each token for nonlinear representation learning. The official implementation is available at this repository: <https://github.com/thuml/iTransformer>.

PatchTST [27] segments time series into subseries-level patches, which serve as input tokens to the Transformer. These patch tokens replace traditional attention tokens, and a channel-independent structure further boosts efficiency. The official implementation is available at this repository: <https://github.com/yuqinie98/PatchTST>.

FEDformer [54] utilizes sparse attention with low-rank approximation in the frequency domain, achieving linear computational complexity and memory efficiency. It also introduces a mixture of expert decomposition to manage distribution shifts in time series [9]. The official implementation is available at this repository: <https://github.com/MAZiqing/FEDformer>.

Crossformer [53] embeds input data into a 2D vector array using Dimension-Segment-Wise embedding to retain time and dimension information, and employs a Two-Stage Attention layer to capture cross-time and cross-dimension dependencies efficiently. The official implementation is available at this repository: <https://github.com/Thinklab-SJTU/Crossformer>.

Autoformer [41] uses an auto-correlation mechanism instead of self-attention and incorporates a decomposition block to separate trend and seasonal components, enhancing learning. The official implementation is available at this repository: <https://github.com/thuml/Autoformer>.

FourierGNN [46] introduces a hypervariate graph, treating each series value as a node and representing sliding windows as space-time fully-connected graphs. It stacks Fourier Graph Operators (FGO) for matrix multiplications in Fourier space, providing high expressiveness with reduced complexity for efficient modeling. The official implementation is available at this repository: <https://github.com/aikunyi/FourierGNN>.

CrossGNN [13] refines both cross-scale and cross-variable interaction for multivariate time series with a linear complexity. Cross-scale GNN captures the scales with clearer trend and weaker noise, while cross-variable GNN maximally exploits the homogeneity and heterogeneity between different variables. The official implementation is available at this repository: <https://github.com/hqh0728/CrossGNN>.

TodyNet [22] captures hidden spatiotemporal dependencies without relying on a predefined graph structure. It introduces a temporal graph pooling layer to generate a global graph-level representation, leveraging learnable temporal parameters for graph learning. The official implementation is available at this repository: <https://github.com/liuxz1011/TodyNet>.

SimTSC [51] frames time series classification (TSC) as a node classification problem on graphs. It introduces a graph construction strategy and a batch training algorithm with negative sampling to enhance training efficiency. The official implementation is available at this repository: <https://github.com/daochenzha/SimTSC>.

A.3 Evaluation metrics

To comprehensively and fairly evaluate the performance of each model in the classification task, we select five evaluation metrics: Accuracy, Precision, Recall, F1 score, and AUROC. The definitions and specific calculation formulas for each metric are presented below:

Accuracy measures the proportion of correct predictions out of the total number of predictions. It's calculated as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}. \quad (17)$$

This metric is useful when the classes are balanced but may be misleading in cases of class imbalance.

Precision focuses on the quality of positive predictions and measures the proportion of correctly predicted positive instances out of all instances predicted as positive. It's especially useful when false positives need to be minimized. The formula is:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}. \quad (18)$$

Recall measures the proportion of actual positive instances that were correctly identified. It's important when false negatives are costly. The formula is:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}. \quad (19)$$

It shows how well the model captures all relevant instances.

The F1 score is the harmonic mean of precision and recall, balancing the two when one is more important than the other. It's particularly useful when dealing with imbalanced datasets, as it accounts for both false positives and false negatives. The formula is:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (20)$$

It gives a single metric that reflects both precision and recall performance.

AUROC measures the model's ability to distinguish between classes, regardless of the decision threshold. The ROC curve plots the true positive rate (recall) against the false positive rate (FPR), and AUROC is the area under this curve. A value of 1 indicates perfect classification, while 0.5 represents random guessing. It is a useful metric when the dataset is imbalanced and provides insight into how well the model separates the classes.

A.4 Implementation Details

We follow the same data processing and train-validation-test set split protocol employed in Medformer [38]. All the experiments are implemented in PyTorch 2.2.2 [28] and conducted on a server equipped with four GeForce RTX 4090 GPUs, each with 24GB of memory. We utilize ADAM [17] optimizer with an uniform initial learning rate $lr = 10^{-4}$ and cross-entropy loss for the model optimization. The batch size is selected from $\{32, 64, 128, 256\}$ and the number of training epochs is fixed to 10. We choose a subset from $\{2, 4, 6, 8, 10, 12, 14, 16\}$ as the multiple resolutions. The number of EncoderLayer L is selected from $\{4, 6\}$ and the dimension of embedding D is set from $\{256, 512\}$. And the graph node dimension is picked from $\{6, 8, 10\}$. We also report the standard deviation of MedGNN's performance over five runs with different seeds, as shown in Table 1 and 2, demonstrating the stability of its performance.