MODELING MULTIVARIATE BIOSIGNALS WITH GRAPH NEURAL NETWORKS AND STRUCTURED STATE SPACE

Siyi Tang[†], **Jared Dunnmon**[†], **Liangqiong Qu**[‡], **Khaled Saab**[†], **Tina Baykaner**[†], **Christopher Lee-Messer**[†], **Daniel Rubin**[†]
[†]Stanford University, [‡] University of Hong Kong
siyitang@stanford.edu

ABSTRACT

Multivariate biosignals are prevalent in many medical domains. Modeling multivariate biosignals is challenging due to (1) long-range temporal dependencies and (2) complex spatial correlations between electrodes. To address these challenges, we propose representing multivariate biosignals as time-dependent graphs and introduce GRAPHS4MER, a general graph neural network (GNN) architecture that models spatiotemporal dependencies in multivariate biosignals. Specifically, (1) we leverage the Structured State Spaces architecture, a state-of-the-art deep sequence model, to capture long-range temporal dependencies in biosignals and (2) we propose a graph structure learning layer to learn dynamically evolving graph structures in the data. We evaluate our proposed model on three distinct tasks and show that GRAPHS4MER consistently improves over existing models, including (1) seizure detection from electroencephalography signals, outperforming a previous GNN with self-supervised pre-training by 3.1 points in AUROC; (2) sleep staging from polysomnography signals, a 4.1 points improvement in macro-F1 score over existing sleep staging models; and (3) electrocardiogram classification, outperforming previous state-of-the-art models by 2.7 points in macro-F1 score.

1 INTRODUCTION

Multivariate biosignals are signals measured by multiple sensors and play critical roles in many medical domains. Several challenges exist in modeling spatiotemporal dependencies in multivariate biosignals. First, most biosignals are sampled at a high sampling rate, which results in long sequences that can be up to tens of thousands of time steps. Moreover, biosignals often involve long-range temporal correlations (Berthouze et al., 2010). Therefore, a model that is capable of modeling long-range temporal correlations is needed to better capture temporal dependencies in biosignals. Recently, the Structured State Space sequence model (S4) (Gu et al., 2022), a deep sequence model based on the classic state space approach, has outperformed previous state-of-the-art (SoTA) models on challenging long sequence modeling tasks, such as Long Range Arena benchmark (Tay et al., 2020), raw speech classification (Gu et al., 2022), and audio generation (Goel et al., 2022).

Second, sensors have complex, non-Euclidean spatial correlations. Graphs are data structures that can represent complex, non-Euclidean correlations in the data (Chami et al., 2022; Bronstein et al., 2017). Previous temporal graph neural networks (GNNs) for modeling multivariate time series (Covert et al., 2019; Tang et al., 2022b; Li et al., 2018; Wu et al., 2019; Zheng et al., 2020; Jiang & Luo, 2022; Tian & Chan, 2021) only use sequences up to hundreds of time steps and require a predefined, static graph structure. However, the graph structure of multivariate biosignals may not be easily defined due to variability in sensor locations. Moreover, the underlying graph connectivity can evolve over time due to changes in the underlying biology. Hence, the ability to dynamically learn the underlying graph structures is highly desirable. Graph structure learning (GSL) aims to jointly learn an optimized graph structure and its node and graph representations (Zhu et al., 2021). GSL has been employed for modeling traffic flows (Zhang et al., 2020; Tang et al., 2022a; Shang et al., 2021; Wu et al., 2019; Bai et al., 2020), fMRI (El-Gazzar et al., 2021; 2022), and sleep staging (Jia et al., 2020). However, these studies are limited to sequences of less than 1k time steps and do not capture dynamic graph structures evolving over time.

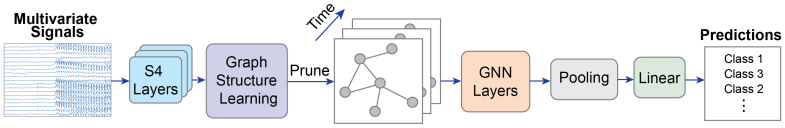

Figure 1: Architecture of GRAPHS4MER. The model has three main components: (1) stacked S4 layers; (2) a graph structure learning (GSL) layer; (3) GNN layers.

In this study, we address the foregoing challenges and make the following main contributions. **First**, we propose GRAPHS4MER (Figure 1), a general GNN architecture for modeling multivariate biosignals. Our modeling contributions are: (1) we leverage S4 to capture long-range temporal dependencies in biosignals, (2) our model dynamically learns the underlying graph structures in the data without predefined graphs, and (3) our approach is a novel, effective way of combining S4, GSL, and GNN. **Second**, we evaluate GRAPHS4MER on three datasets with distinct data modalities and tasks. Our model consistently outperforms existing methods on (1) seizure detection from EEGs, outperforming a previous GNN with self-supervised pre-training by 3.1 points in AUROC; (2) sleep staging from polysomnography signals, outperforming existing models by 4.1 points in macro-F1 score; (3) ECG classification, outperforming previous SoTA models by 2.7 points in macro-F1 score. **Lastly**, qualitative interpretability analysis suggests that our GSL method learns meaningful graph structures that reflect the underlying seizure classes, sleep stages, and ECG abnormalities.

2 METHODS

Problem setup. Let $\mathbf{X} \in \mathbb{R}^{N \times T \times M}$ be a multivariate biosignal, where N is the number of sensors, T is the sequence length, and M is the input dimension of the signal (typically $M = 1$). We represent the multivariate signal as a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$, where the set of nodes $\mathcal{V}$ corresponds to the sensors (channels/leads), $\mathcal{E}$ is the set of edges, and $\mathbf{W}$ is the adjacency matrix. Here, $\mathcal{E}$ and $\mathbf{W}$ are unknown and will be learned by our model. While our formulation is general to any type of node or graph classification and regression tasks, we focus on graph classification tasks in this work.

Temporal modeling with S4. We leverage S4 to capture long-range temporal dependencies in biosignals. Naively applying S4 to projects the N signal channels to the hidden dimension with a linear layer (Gu et al., 2022), which may be suboptimal because it neglects the underlying graph structure of biosignals. Instead, we use stacked S4 layers to embed signals in each sensor independently, resulting in an embedding $\mathbf{H} \in \mathbb{R}^{N \times T \times D}$ (referred to as "S4 embeddings" hereafter) for each input signal $\mathbf{X}$, where D is the hidden dimension.

Dynamic graph structure learning. To model spatial dependencies in biosignals (i.e., graph adjacency matrix $\mathbf{W}$), we develop a GSL layer to learn the similarities between nodes. To capture the dynamics of signals that evolve over time, our GSL layer learns a *unique* graph structure within a short time interval r, where r is a pre-specified resolution. Instead of learning a unique graph at each time step, we choose to learn a graph over a time interval of length r because (1) aggregating information across a longer time interval can result in less noisy graphs and (2) it is more computationally efficient. For convenience, we let the predefined resolution r be an integer and assume that the sequence length T is divisible by r without overlaps, and denote $n_d = \frac{T}{r}$ as the number of dynamic graphs. We adopt self-attention (Vaswani et al., 2017) and use the attention weights as edge weights. The adjacency matrix of the t-th dynamic graph, $\overline{\mathbf{W}}^{(t)} \in \mathbb{R}^{N \times N}$, is learned as follows:

$$\mathbf{Q} = \mathbf{h}^{(t)}\mathbf{M}^Q, \;\; \mathbf{K} = \mathbf{h}^{(t)}\mathbf{M}^K, \;\; \overline{\mathbf{W}}^{(t)} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D}}), \text{ for } t = 1, 2, ..., n_d \tag{1}$$

Here, $\mathbf{h}^{(t)} \in \mathbb{R}^{N \times D}$ is mean-pooled S4 embeddings within the t-th time interval of length r; $\mathbf{M}^Q \in \mathbb{R}^{D \times D}$ and $\mathbf{M}^K \in \mathbb{R}^{D \times D}$ are weights projecting $\mathbf{h}^{(t)}$ to query $\mathbf{Q}$ and key $\mathbf{K}$, respectively. The above equations can be easily extended to multihead self-attention (Vaswani et al., 2017).

To guide the GSL process, for the t-th dynamic graph, we add a k-nearest neighbor (KNN) graph $\mathbf{W}_{\text{KNN}}^{(t)}$ to the learned adjacency matrix $\overline{\mathbf{W}}^{(t)}$, where each node's k-nearest neighbors are defined by cosine similarity between their respective values in $\mathbf{h}^{(t)}$. i.e., $\mathbf{W}^{(t)} = \epsilon \mathbf{W}_{\text{KNN}}^{(t)} + (1-\epsilon)\overline{\mathbf{W}}^{(t)}$, where

$\epsilon \in [0, 1)$ is a hyperparameter for the weight of the KNN graph. To introduce graph sparsity, we prune $\mathbf{W}^{(t)}$ by removing edges whose weights are smaller than a threshold κ (a hyperparameter).

To encourage the learned graphs to have desirable properties, including smoothness, sparsity, and connectivity (Chen et al., 2020; Kalofolias, 2016; Zhu et al., 2021), we include three regularization terms as detailed in Equations 2–3 in Appendix A. The regularization loss is the weighted sum of the three terms and averaged across all dynamic graphs, where the weights are hyperparameters.

Model architecture. The overall architecture of our model (Figure 1), GRAPHS4MER, consists of three main components: (1) stacked S4 layers with residual connection to model temporal dependencies in signals within each sensor independently, which maps raw signals $\mathbf{X} \in \mathbb{R}^{N \times T \times M}$ to S4 embedding $\mathbf{H} \in \mathbb{R}^{N \times T \times D}$, (2) a GSL layer to learn dynamically evolving adjacency matrices $\mathbf{W}^{(1)}, ..., \mathbf{W}^{(n_d)}$, and (3) GNN layers to learn spatial dependencies between sensors given the learned graph structures and node features $\mathbf{H}$. We use an expressive GNN architecture, Graph Isomorphism Network (Xu et al., 2019; Hu et al., 2020), in our experiments. Finally, a temporal pooling and a graph pooling layer are added to aggregate temporal and spatial representations, respectively, followed by a fully connected layer to produce a prediction for each multivariate biosignal. Source code is publicly available at `https://github.com/tsy935/graphs4mer`.

3 EXPERIMENTS

We evaluate GRAPHS4MER on three tasks. **(1) Seizure detection from EEGs.** We use the public TUSZ v1.5.2 dataset (Shah et al., 2018), and follow the same experimental setup of seizure detection on 60-s EEGs as in Tang et al. (2022b). The number of EEG sensors is 19 and the sequence length is 12k time steps. The task is binary classification of detecting whether or not a 60-s EEG contains seizure. Baselines are prior SoTA models from Tang et al. (2022b). **(2) Sleep staging from polysomnography signals.** We use the public DOD-H (Guillot et al., 2020) dataset. The number of polysomnography (PSG) sensors is 16 and the sequence length is 7.5k time steps. The task is to classify each 30-s PSG signal as one of the five sleep stages: wake, rapid eye movement (REM), N1, N2, and N3. Baseline models include existing sleep staging models (Guillot et al., 2020; Guillot & Thorey, 2021; Hochreiter & Schmidhuber, 1997). **(3) ECG classification.** We use the public ICBEB ECG dataset (Liu et al., 2018), and follow the same data split as described in Strodthoff et al. (2021). The number of ECG channels is 12 and the sequence lengths vary from 600 to 6k time steps. The task is to classify each ECG into at least one of nine classes (see Table 7 in Appendix B). Baseline models include seven prior CNNs/RNNs for ECG classification (Strodthoff et al., 2021; Ismail Fawaz et al., 2020; He et al., 2019; Wang et al., 2017; Hochreiter & Schmidhuber, 1997). For each model, we ran three runs with different random seeds and report mean and standard deviation of results. See Appendix B–C for detailed experimental setup, model training procedures, and hyperparameters.

Table 1: Seizure detection results. Best and second best results are **bolded** and underlined.

Model	AUROC
LSTM (Hochreiter & Schmidhuber, 1997)	0.715 ± 0.016
Dense-CNN (Saab et al., 2020)	0.796 ± 0.014
CNN-LSTM (Ahmedt-Aristizabal et al., 2020)	0.682 ± 0.003
Dist-DCRNN w/o pre-training (Tang et al., 2022b)	0.793 ± 0.022
Corr-DCRNN w/o pre-training (Tang et al., 2022b)	0.804 ± 0.015
Dist-DCRNN w/ pre-training (Tang et al., 2022b)	0.875 ± 0.016
Corr-DCRNN w/ pre-training (Tang et al., 2022b)	0.850 ± 0.014
GRAPHS4MER (ours)	**0.906 ± 0.012**

Seizure detection results. As shown in Table 1, GRAPHS4MER outperforms the previous SoTA, Dist-DCRNN with pre-training, by 3.1 points in AUROC. Notably, Dist-DCRNN was pre-trained using a self-supervised task (Tang et al., 2022b), whereas GRAPHS4MER was trained from scratch without the need of pre-training.

Table 2: Sleep staging results.

Model	Macro-F1	Kappa
LSTM (Hochreiter & Schmidhuber, 1997)	0.609 ± 0.034	0.539 ± 0.046
SimpleSleepNet (Guillot et al., 2020)	0.720 ± 0.001	0.703 ± 0.013
RobustSleepNet (Guillot & Thorey, 2021)	0.777 ± 0.007	0.758 ± 0.008
DeepSleepNet (Supratak et al., 2017)	0.716 ± 0.025	0.711 ± 0.032
GRAPHS4MER (ours)	**0.818 ± 0.008**	**0.802 ± 0.014**

Sleep staging results. As shown in Table 2, GRAPHS4MER outperforms RobustSleepNet, a specialized sleep staging model, by 4.1 points in macro-F1. Note that the baselines preprocess the PSG signals using short-time Fourier transform, whereas our model directly operates on raw PSG signals without preprocessing.

ECG classification results. Table 3 compares GRAPHS4MER to existing models. GRAPHS4MER provides 2.7 points improvement in macro-F1 score over XResNet1D (He et al., 2019).

Table 3: ECG classification results.

Model	Macro F1-Score	Macro F2-Score	Macro G2-Score	Macro AUROC
InceptionTime (Ismail Fawaz et al., 2020)	0.778 ± 0.020	0.792 ± 0.023	0.576 ± 0.027	0.964 ± 0.008
XResNet1D (He et al., 2019)	0.782 ± 0.016	0.803 ± 0.016	0.587 ± 0.019	0.971 ± 0.001
ResNet1D (Wang et al., 2017)	0.772 ± 0.015	0.788 ± 0.008	0.570 ± 0.014	0.963 ± 0.004
FCN (Wang et al., 2017)	0.736 ± 0.015	0.764 ± 0.014	0.538 ± 0.014	0.950 ± 0.003
Bidir-LSTM (Hochreiter & Schmidhuber, 1997)	0.748 ± 0.009	0.769 ± 0.004	0.548 ± 0.005	0.945 ± 0.002
WaveletNN (Strodthoff et al., 2021)	0.621 ± 0.013	0.643 ± 0.019	0.396 ± 0.014	0.911 ± 0.002
GRAPHS4MER (ours)	**0.809 ± 0.004**	**0.804 ± 0.004**	**0.609 ± 0.005**	**0.977 ± 0.001**

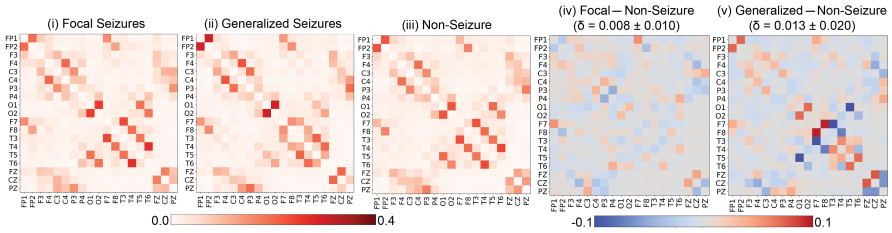

Figure 2: Mean adjacency matrix for EEG for **(i)** focal seizures, **(ii)** generalized seizures, and **(iii)** non-seizure. **(iv)** Difference between focal seizure and non-seizure. **(v)** Difference between generalized seizure and non-seizure. δ indicates the mean and standard deviation of the absolute values of the differences between two mean adjacency matrices.

Ablations. We investigate the importance of (1) graph-based representation, where we remove GSL and GNN layers; (2) S4 encoder, where we replace S4 layers with GRUs (Cho et al., 2014); (3) long-range temporal modeling for GSL, where we apply GSL and GNN layers to raw signals, followed by S4 layers; and (4) GSL, where we remove GSL layer and use predefined graphs. We use the distance-based EEG graph from Tang et al. (2022b) for seizure detection; DOD-H and ICBEB datasets have no predefined graph, and thus we use a KNN graph based on cosine similarity of raw signals between nodes. We find that removing any of these components decreases model performance (Table 4).

Table 4: Ablation results.

Model	TUSZ (AUROC)	DOD-H (Macro-F1)	ICBEB (Macro-F1)
S4	0.824 ± 0.011	0.778 ± 0.009	0.781 ± 0.003
GRAPHS4MER w/o S4	0.705 ± 0.095	0.634 ± 0.061	0.197 ± 0.005
GSL-GNN-S4	0.882 ± 0.014	0.797 ± 0.011	0.772 ± 0.012
GRAPHS4MER w/o GSL	0.899 ± 0.010	0.765 ± 0.016	0.797 ± 0.012
GRAPHS4MER	**0.906 ± 0.012**	**0.818 ± 0.008**	**0.809 ± 0.004**

Interpretation of graphs. To investigate if the learned graphs are meaningful, we visualize the mean adjacency matrices in the correctly predicted test samples, grouped by seizure classes, sleep stages, or ECG classes. The adjacency matrices were reviewed by clinical experts in terms of whether the differences in adjacency matrices between diseased (or sleep) and normal (or wake) classes reflect characteristics of the diseased (or sleep) states. For the EEG use case (Figure 2), the magnitude of differences between generalized seizure and non-seizure adjacency matrices (Figure 2v) is larger than the magnitude of differences between focal seizure and non-seizure (Figure 2iv). This suggests that the abnormalities in generalized seizures are more synchronized across channels, which is consistent with the literature that generalized seizures are characterized with abnormally synchronized brain activity (Gloor et al., 1990; Amor et al., 2009). Similarly, for PSG and ECG, we observe that the learned graphs reflect sleep stages and ECG abnormalities (see Appendix D).

4 CONCLUSION AND FUTURE WORK

In conclusion, we presented GRAPHS4MER, a general GNN integrating S4 and GSL for modeling multivariate biosignals. Our method set new SoTA performance on seizure detection, sleep staging, and ECG classification, and learned meaningful graph structures that reflect seizure classes, sleep stages, and ECG abnormalities. Exciting future directions include (1) leveraging domain knowledge for improved GSL, (2) investigating other ways of combining S4 and GSL to further improve long-range GSL, and (3) applying our methods to other use cases, including regression tasks.

ACKNOWLEDGMENTS

We thank HAI and Google Cloud for the support of Google Cloud credits for this work.

REFERENCES

David Ahmedt-Aristizabal, Tharindu Fernando, Simon Denman, Lars Petersson, Matthew J Aburn, and Clinton Fookes. Neural memory networks for seizure type classification. *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, 2020:569–575, July 2020.

Frédérique Amor, Sylvain Baillet, Vincent Navarro, Claude Adam, Jacques Martinerie, and Michel Le Van Quyen. Cortical local and long-range synchronization interplay in human absence seizure initiation. *Neuroimage*, 45(3):950–962, April 2009.

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17804–17815, 2020.

Belkin and Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Adv. Neural Inf. Process. Syst.*, 2001.

Luc Berthouze, Leon M James, and Simon F Farmer. Human EEG shows long-range temporal correlations of oscillation amplitude in theta, alpha and beta bands across a wide age range. *Clin. Neurophysiol.*, 121(8):1187–1197, August 2010.

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Process. Mag.*, 34(4):18–42, July 2017.

Chami, Abu-El-Haija, Perozzi, Ré, and others. Machine learning on graphs: A model and comprehensive taxonomy. *Journal of Machine Learning Research*, 2022.

Yu Chen, Lingfei Wu, and Mohammed Zaki. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. *Adv. Neural Inf. Process. Syst.*, 33:19314–19326, 2020.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, Doha, Qatar, October 2014.

Ian C Covert, Balu Krishnan, Imad Najm, Jiening Zhan, Matthew Shore, John Hixson, and Ming Jack Po. Temporal graph convolutional networks for automatic seizure detection. In *Proceedings of the 4th Machine Learning for Healthcare Conference*, volume 106 of *Proceedings of Machine Learning Research*, pp. 160–180. PMLR, 2019.

Ahmed El-Gazzar, Rajat Mani Thomas, and Guido van Wingen. Dynamic adaptive Spatio-Temporal graph convolution for fMRI modelling. In *Machine Learning in Clinical Neuroimaging*, pp. 125–134, Cham, 2021. Springer International Publishing.

Ahmed El-Gazzar, Rajat Mani Thomas, and Guido van Wingen. Improving the diagnosis of psychiatric disorders with self-supervised graph state space models. *arXiv preprint arXiv:2206.03331*, 2022.

P Gloor, M Avoli, and G Kostopoulos. Thalamocortical relationships in generalized epilepsy with bilaterally synchronous Spike-and-Wave discharge. In Massimo Avoli, Pierre Gloor, George Kostopoulos, and Robert Naquet (eds.), *Generalized Epilepsy: Neurobiological Approaches*, pp. 190–212. Birkhäuser Boston, Boston, MA, 1990.

Karan Goel, Albert Gu, Chris Donahue, and Christopher Re. It's raw! Audio generation with state-space models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 7616–7633. PMLR, 17–23 Jul 2022.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Antoine Guillot and Valentin Thorey. RobustSleepNet: Transfer learning for automated sleep staging at scale. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 29:1441–1451, July 2021.

Antoine Guillot, Fabien Sauvet, Emmanuel H During, and Valentin Thorey. Dreem open datasets: Multi-Scored sleep datasets to compare human and automated sleep staging. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 28(9):1955–1965, September 2020.

Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 558–567. IEEE, June 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJlWWJSFDH.

Conrad Iber, Sonia Ancoli-Israel, Andrew L Chesson, and Stuart F Quan. *The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications*, volume 1. American academy of sleep medicine, 2007.

Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. InceptionTime: Finding AlexNet for time series classification. *Data Min. Knowl. Discov.*, 34(6):1936–1962, November 2020.

Ziyu Jia, Youfang Lin, Jing Wang, Ronghao Zhou, Xiaojun Ning, Yuanlai He, and Yaoshuai Zhao. Graphsleepnet: Adaptive spatial-temporal graph convolutional networks for sleep stage classification. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 1324–1330. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/184. URL https://doi.org/10.24963/ijcai.2020/184. Main track.

Weiwei Jiang and Jiayun Luo. Graph neural network for traffic forecasting: A survey. *Expert Syst. Appl.*, 207:117921, November 2022.

Vassilis Kalofolias. How to learn a graph from smooth signals. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 920–929, Cadiz, Spain, 2016. PMLR.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJiHXGWAZ.

Feifei Liu, Chengyu Liu, Lina Zhao, Xiangyu Zhang, Xiaoling Wu, Xiaoyan Xu, Yulin Liu, Caiyun Ma, Shoushui Wei, Zhiqiang He, Jianqing Li, and Eddie Ng Yin Kwee. An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection. *Journal of Medical Imaging and Health Informatics*, 8(7):1368–1373, 2018.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Antonio Ortega, Pascal Frossard, Jelena Kovačević, José M F Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proc. IEEE*, 106(5):808–828, May 2018.

Ilene M Rosen, Douglas B Kirsch, Kelly A Carden, Raman K Malhotra, Kannan Ramar, R Nisha Aurora, David A Kristo, Jennifer L Martin, Eric J Olson, Carol L Rosen, James A Rowley, Anita V Shelgikar, and American Academy of Sleep Medicine Board of Directors. Clinical use of a home sleep apnea test: An updated american academy of sleep medicine position statement. *J. Clin. Sleep Med.*, 14(12):2075–2077, December 2018.

Khaled Saab, Jared Dunnmon, Christopher Ré, Daniel Rubin, and Christopher Lee-Messer. Weak supervision as an efficient approach for automated seizure detection in electroencephalography. *NPJ Digit Med*, 3:59, April 2020.

Vinit Shah, Eva von Weltin, Silvia Lopez, James Riley McHugh, Lillian Veloso, Meysam Golmohammadi, Iyad Obeid, and Joseph Picone. The Temple University Hospital Seizure Detection Corpus. *Front. Neuroinform.*, 12:83, November 2018.

Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=WEHSlH5mOk.

Nils Strodthoff, Patrick Wagner, Tobias Schaeffter, and Wojciech Samek. Deep learning for ECG analysis: Benchmarks and insights from PTB-XL. *IEEE J Biomed Health Inform*, 25(5):1519–1528, May 2021.

Akara Supratak, Hao Dong, Chao Wu, and Yike Guo. DeepSleepNet: A model for automatic sleep stage scoring based on raw Single-Channel EEG. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 25(11): 1998–2008, November 2017.

Jiabin Tang, Tang Qian, Shijing Liu, Shengdong Du, Jie Hu, and Tianrui Li. Spatio-temporal latent graph structure learning for traffic forecasting, 2022a. URL https://arxiv.org/abs/2202.12586.

Siyi Tang, Jared Dunnmon, Khaled Kamal Saab, Xuan Zhang, Qianying Huang, Florian Dubost, Daniel Rubin, and Christopher Lee-Messer. Self-Supervised graph neural networks for improved electroencephalographic seizure analysis. In *International Conference on Learning Representations*, April 2022b.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers, 2020. URL https://arxiv.org/abs/2011.04006.

Chenyu Tian and Wai Kin (victor) Chan. Spatial-temporal attention wavenet: A deep learning framework for traffic prediction considering spatial-temporal dependencies. *IET Intel. Transport Syst.*, 15(4):549–561, April 2021.

Vaswani, Shazeer, Parmar, and others. Attention is all you need. *Adv. Neural Inf. Process. Syst.*, 2017.

Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1578–1585. ieeexplore.ieee.org, May 2017.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 1907–1913. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/264. URL https://doi.org/10.24963/ijcai.2019/264.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Qi Zhang, Jianlong Chang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Spatio-Temporal graph structure learning for traffic forecasting. *AAAI*, 34(01):1177–1185, April 2020.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. GMAN: A graph Multi-Attention network for traffic prediction. *AAAI*, 34(01):1234–1241, April 2020.

Yanqiao Zhu, Weizhi Xu, Jinghao Zhang, Yuanqi Du, Jieyu Zhang, Qiang Liu, Carl Yang, and Shu Wu. A survey on graph structure learning: Progress and opportunities. *arXiv e-prints*, pp. arXiv–2103, 2021.

APPENDIX

A GRAPH REGULARIZATION

Graph regularization encourages a learned graph to have several desirable properties, such as smoothness, sparsity, and connectivity (Chen et al., 2020; Kalofolias, 2016; Zhu et al., 2021). Let $\mathbf{X} \in \mathbb{R}^{N \times D}$ be a graph data with N nodes and D features. First, a common assumption in graph signal processing is that features change smoothly between adjacent nodes (Ortega et al., 2018). Given an undirected graph with adjacency matrix $\mathbf{W}$, the smoothness of the graph can be measured by the Dirichlet energy (Belkin & Niyogi, 2001):

$$\mathcal{L}_{\text{smooth}}(\mathbf{X}, \mathbf{W}) = \frac{1}{2N^2} \sum_{i,j} \mathbf{W}_{ij} ||\mathbf{X}_{i,:} - \mathbf{X}_{j,:}||^2 \ = \frac{1}{N^2} \text{tr}\big(\mathbf{X}^T \mathbf{L} \mathbf{X}\big) \tag{2}$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian, $\mathbf{D}$ is the degree matrix of $\mathbf{W}$. Minimizing Equation 2 therefore encourages the learned graph to be smooth. In practice, the normalized graph Laplacian $\hat{\mathbf{L}} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}$ is used so that it is independent of node degrees.

However, simply minimizing the smoothness may result in a trivial solution $\mathbf{W} = \mathbf{0}$ (Chen et al., 2020). To avoid this trivial solution and encourage sparsity of the learned graph, additional constraints can be added (Chen et al., 2020; Kalofolias, 2016):

$$\mathcal{L}_{\text{degree}}(\mathbf{W}) = -\frac{1}{N} \mathbf{1}^T \log(\mathbf{W}\mathbf{1}) \tag{3}$$

$$\mathcal{L}_{\text{sparse}}(\mathbf{W}) = \frac{1}{N^2} ||\mathbf{W}||_F^2 \tag{4}$$

where $||.||_F$ is the Frobenius norm of a matrix. Intuitively, $\mathcal{L}_{\text{degree}}$ penalizes disconnected graphs and $\mathcal{L}_{\text{sparse}}$ discourages nodes with high degrees (i.e., encourages the learned graph to be sparse).

B DETAILS OF DATASETS, EXPERIMENTAL SETUP, AND BASELINES

Temple University Hospital Seizure Detection Corpus (TUSZ). We use the publicly available Temporal University Hospital Seizure Detection Corpus (TUSZ) v1.5.2. for seizure detection (Shah et al., 2018). We follow the same experimental setup as in Tang et al. (2022b). The TUSZ train set is divided into train and validation splits with distinct patients, and the TUSZ test set is held-out for model evaluation. The following 19 EEG channels are included: FP1, FP2, F3, F4, C3, C4, P3, P4, O1, O2, F7, F8, T3, T4, T5, T6, FZ, CZ, and PZ. Because the EEG signals are sampled at different sampling rate, we resample all the EEG signals to 200 Hz. Following a prior study (Tang et al., 2022b), we also exclude 5 patients in the TUSZ test set who appear in both the TUSZ train and test sets. The EEG signals are divided into 60-s EEG clips without overlaps, and the task is to predict whether or not an EEG clip contains seizure. Each 60-s EEG clip has a sequence length of 12k time steps. The resolution r in GSL layer is chosen as 10-s (i.e., 2,000 time steps), which is inspired by how trained EEG readers analyze EEGs. We treat the EEG graphs as undirected. Table 5 shows the number of EEG clips and patients in the train, validation, and test splits.

We compare our model performance to existing models for seizure detection, including (1) LSTM (Hochreiter & Schmidhuber, 1997), a variant of RNN with gating mechanisms; (2) Dense-CNN (Saab et al., 2020), a densely connected CNN specialized in seizure detection; (3) CNN-LSTM (Ahmedt-Aristizabal et al., 2020); and (4) Dist- and Corr-DCRNN without and with self-supervised pre-training (Tang et al., 2022b). Following prior studies, we use AUROC as the evaluation metric.

Table 5: Number of EEG clips and patients in train, validation, and test splits of TUSZ dataset.

	EEG Clips (% Seizure)	Patients (% Seizure)
Train Set	38,613 (9.3%)	530 (34.0%)
Validation Set	5,503 (11.4%)	61 (36.1%)
Test Set	8,848 (14.7%)	45 (77.8%)

Dreem Open Dataset-Healthy (DOD-H). We use the publicly available Dreem Open Dataset-Healthy (DOD-H) for sleep staging (Guillot et al., 2020). The DOD-H dataset consists of overnight PSG sleep recordings from 25 volunteers. The PSG signals are measured from 12 EEG channels, 1 electromyographic (EMG) channel, 2 electrooculography (EOG) channels, and 1 electrocardiogram channel using a Siesta PSG device (Compumedics). All the signals are sampled at 250 Hz. Following the standard AASM Scoring Manual and Recommendations (Rosen et al., 2018), each 30-s PSG signal is annotated by a consensus of 5 experienced sleep technologists as one of the five sleep stages: wake (W), rapid eye movement (REM), non-REM sleep stages N1, N2, and N3. We randomly split the PSG signals by 60/20/20 into train/validation/test splits, where each split has distinct subjects. Each 30-s PSG signal has a sequence length of 7.5k time steps. The resolution r in GSL layer is set as 10-s (i.e., 2,500 time steps). We treat the PSG graphs as undirected. Table 6 shows the number of 30-s PSG clips, the number of subjects, and the five sleep stage distributions.

Baseline models include existing sleep staging models that achieved state-of-the-art on DOD-H dataset, SimpleSleepNet (Guillot et al., 2020), RobustSleepNet (Guillot & Thorey, 2021), and DeepSleepNet (Supratak et al., 2017), all of which are based on CNNs and/or RNNs. We also include the traditional sequence model LSTM (Hochreiter & Schmidhuber, 1997) as a baseline. For fair comparisons between the baselines and GRAPHS4MER, we trained SimpleSleepNet, RobustSleepNet, and DeepSleepNet using the open sourced code[1] and setting the temporal context to be one 30-s PSG signal. Similar to Guillot & Thorey (2021), we use macro-F1 score and Cohen's Kappa as the evaluation metrics.

Table 6: Number of subjects and 30-s PSG clips in the train, validation, and test splits of DOD-H dataset.

	Subjects	30-s PSG Clips					
		Total	Wake	N1	N2	N3	REM
Train	15	14,823	1,839	925	6,965	2,015	3,079
Validation	5	5,114	480	254	2,480	990	910
Test	5	4,725	718	326	2,434	509	738

ICBEB ECG Dataset. We use the publicly available ECG dataset for the 1st China Physiological Signal Challenge held during the International Conference on Biomedical Engineering and Biotechnology (ICBEB) for ECG classification (Liu et al., 2018). We follow the same data split as described in Strodthoff et al. (2021). Specifically, as the official ICBEB test set is not publicly available, we only use the official ICBEB training set and randomly split it into 10 folds using stratified split, where the first 8 folds are used as the training set, the 9th fold is used as the validation set for hyperparameter tuning, and the 10th fold is used as the held-out test set to report results in this study. In total, there are 6,877 12-lead ECGs ranging between 6-s and 60-s. Following Strodthoff et al. (2021), the ECGs are downsampled to 100 Hz, resulting in sequence lengths ranging from 600 to 6k time steps. To handle variable length ECGs for training in mini batches, we pad the short sequences with 0s. During training and testing, the padded values are masked out so that they are not seen by the models. In the ICBEB dataset, each ECG record is annotated by up to three reviewers. There are 9 classes in total, including one normal and 8 abnormal classes, and each ECG may be associated with more than one abnormal classes (i.e., multilabel classification). The resolution r in the GSL layer is set as the actual ECG sequence lengths to facilitate model training in batches. We treat the ECG graphs as undirected. Table 7 shows the 9 classes and the number of ECGs in each class in the train/validation/test splits.

We compare GRAPHS4MER to a wide variety of prior CNNs/RNNs for ECG classification as in Strodthoff et al. (2021): (1) InceptionTime (Ismail Fawaz et al., 2020), (2) XResNet1D (He et al., 2019), (3) ResNet1D (Wang et al., 2017), (4) fully convolutional network (FCN) (Wang et al., 2017), (5) LSTM (Hochreiter & Schmidhuber, 1997), (6) bidirectional LSTM (Hochreiter & Schmidhuber, 1997), and (7) WaveletNN (Strodthoff et al., 2021). Note that these baselines take 2.5-s ECG windows as inputs and aggregate the window-wise predictions at test time, whereas GRAPHS4MER

[1] SimpleSleepNet: https://github.com/Dreem-Organization/dreem-learning-open; RobustSleepNet, DeepSleepNet: https://github.com/Dreem-Organization/RobustSleepNet

takes the entire ECG signal as inputs, which allows modeling long-range temporal dependencies in the entire signals.

Table 7: Number of ECGs in the train, validation, and test splits of ICBEB dataset used in this study. Note that some ECGs are associated with more than one abnormal classes, and thus the total number of ECGs (last row) is not equal to the sum of the ECGs in the individual classes. *Abbreviations*: AFIB, atrial fibrillation; I-AVB, first-degree atrioventricular block; LBBB, left bundle branch block; RBBB, right bundle branch block; PAC, premature atrial contraction; PVC, premature ventricular contraction; STD, ST-segment depression; STE, ST-segment elevated.

	Train	Validation	Test
Normal	734	92	92
AFIB	976	122	123
I-AVB	578	72	72
LBBB	189	24	23
RBBB	1,487	176	194
PAC	493	61	62
PVC	560	70	70
STD	695	87	87
STE	176	22	22
Total	5,499	690	688

C Details of model training procedures and hyperparameters

Model training was accomplished using the AdamW optimizer (Loshchilov & Hutter, 2019) in PyTorch on a single NVIDIA A100 GPU. All experiments were run for three runs with different random seeds. Cosine learning rate scheduler with 5-epoch warm start was used (Loshchilov & Hutter, 2017). Model training was early stopped when the validation loss did not decrease for 20 consecutive epochs. We performed hyperparameter search for all the hyperparameters in Table 8 on the validation set using an off-the-shelf hyperparameter tuning tool[2].

Table 8: Summary of hyperparameters.

Hyperparameters	Tuning Range
Initial learning rate	$[1 \times 10^{-4}, 1 \times 10^{-2}]$
Dropout rate	[0.1, 0.5]
Hidden dimension	{64, 128, 256}
Number of S4 layers	{2, 3, 4}
S4 bidirectionality	{True, False}
Number of GNN layers	{1, 2}
Graph pooling	{max-pool, mean-pool, sum-pool}
Value of κ threshold for graph pruning	[0.01, 0.5]
Value of K in KNN graph	{2, 3}
Weight of KNN graph (ϵ)	[0.3, 0.6]
α, β, γ weights in graph regularization loss	[0, 1]

Model training and hyperparameters for seizure detection on TUSZ dataset. As there are many more negative samples in the dataset, we undersampled the negative examples in the train set during training. We used binary cross-entropy loss as the loss function. The models were trained for 100 epochs with an initial learning rate of 8×10^{-4}. The batch size was 4; dropout rate was 0.1; hidden dimension was 128; number of stacked S4 layers was 4; S4 layers were unidirectional; number of GNN layers was 1; graph pooling was max-pool and temporal pooling was mean-pool; graph pruning was done by setting a threshold of $\kappa = 0.1$, where edges whose edge weights $<= 0.1$ were

[2] https://docs.wandb.ai/guides/sweeps

removed; $K = 2$ for KNN graph and the KNN graph weight ϵ was 0.6; α, β, and γ weights were all set to 0.05. This results in 265k trainable parameters in GRAPHS4MER. Best model was picked based on the highest AUROC on the validation set.

Model training and hyperparameters for sleep staging on DOD-H dataset. As the DOD-H dataset is highly imbalanced, we undersampled the majority classes in the train set during training. We used cross-entropy loss as the loss function. The models were trained for 100 epochs with an initial learning rate of 1×10^{-3}. The batch size was 4; dropout rate was 0.4; hidden dimension was 128; number of stacked S4 layers was 4 and S4 layers were unidirectional; number of GNN layers was 1; graph pooling was sum-pool and temporal pooling was mean-pool; graph pruning was done by setting a threshold of $\kappa = 0.1$; $K = 3$ for KNN graph and the weight for KNN graph ϵ was 0.6; α, β, and γ weights were all set to 0.2. This results in 266k trainable parameters in GRAPHS4MER. Best model was picked based on the highest macro-F1 score on the validation set.

Model training and hyperparameters for ECG classification on ICBEB dataset. As each ECG in the ICBEB dataset may be associated with more than one classes, this task is a multilabel classification task. Therefore, we used binary cross-entropy loss as the loss function. The models were trained for 100 epochs with an initial learning rate of 1×10^{-3}. The batch size was 8; dropout rate was 0.1; hidden dimension was 128; number of stacked S4 layers was 4; S4 layers were bidirectional; number of GNN layers was 1; graph pooling was mean-pool and temporal pooling was mean-pool; graph pruning was done by setting a threshold of $\kappa = 0.02$; $K = 2$ for KNN graph and the weight for KNN graph ϵ was 0.6; α, β, and γ weights were 1.0, 0.0, and 0.5, respectively. This results in 299k trainable parameters in GRAPHS4MER for ECG classification. Best model was picked based on the highest macro-AUROC on the validation set. To obtain binarized predictions, we selected the cutoff threshold for each class separately by maximizing the F1-score on the validation set for the respective class.

D VISUALIZATION OF PSG AND ECG GRAPHS

Figure 3 shows mean adjacency matrices for PSG signals in correctly predicted test samples grouped by five sleep stages. We observe that N3 differs from wake stage more than N1 (comparing Figure 3ix to Figure 3vii). This pattern is expected given that N1 is the earliest sleep stage, whereas N3 is the deep sleep stage and is associated with slow brain waves that are not present in other sleep stages (Iber et al., 2007). Moreover, in REM stage (Figure 3ii), the EMG channel has very weak connection to all other channels (red arrow in Figure 3ii), which is expected given that one experiences muscle paralysis in REM stage.

Figure 4 shows mean adjacency matrices for ECG signals in correctly predicted test samples, grouped by ECG classes. We find that the magnitude of differences between adjacency matrices of first-degree atrioventricular block (I-AVB) and normal ECG is small (Figure 4xi), which is expected as I-AVB abnormality involves only small changes in the ECG (a subtle increase in the PR interval of ECG, not involving any morphologic changes in the p-waves or the QRS complexes). In contrast, the magnitude of differences between adjacency matrices of left bundle branch block (LBBB) and normal ECG is much more noticeable (Figure 4xii), particularly in ECG leads V1-V6. This finding is clinically meaningful as ECG signals with LBBB demonstrate a pronounced abnormality in the QRS complexes of the ECG, especially in leads V1-V6.

E EFFECT OF TEMPORAL RESOLUTION

To examine the effect of temporal resolution r on model performance, we show the performance versus different values of r for seizure detection and sleep staging in Figure 5. We observe that smaller value of r tends to result in higher performance, which suggests that capturing dynamically varying graph structures is useful for these tasks. Note that in ECG classification experiments, the temporal resolution r is set as the actual sequence length for each ECG due to variable ECG lengths, and thus ECG classification task is excluded from Figure 5.

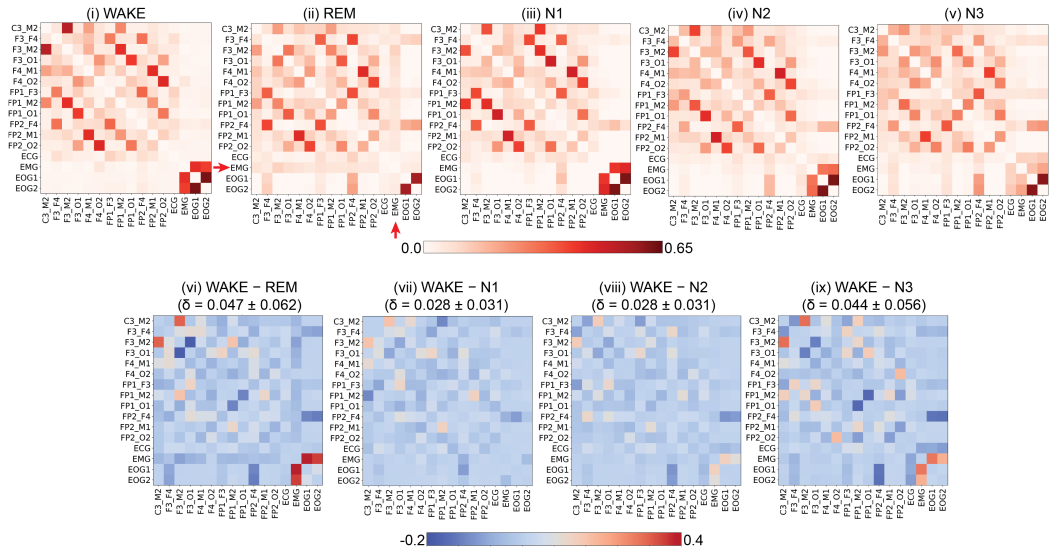

Figure 3: **(i)–(v)** Mean adjacency matrices for PSG signals for five sleep stages in correctly predicted test samples. **(vi)–(ix)** Difference between non-WAKE stages and WAKE. δ indicates the mean and standard deviation of the absolute values of the differences between non-WAKE stages and WAKE. Red arrow indicates EMG channel in REM stage that has weak connection to all other channels. Self-edges (i.e., diagonal) are not shown here.

F Effect of KNN graph in GSL

To examine the effect of adding KNN graph in GSL (see Section 2) on model performance, we show the performance without versus with KNN graph in Figure 6. We observe that adding KNN graph in GSL tends to result in better performance.

Additionally, instead of using a fixed KNN graph weight, we decay the KNN graph weight according to a cosine annealing scheduler (Loshchilov & Hutter, 2017). Figure 7 shows the model performance with and without decay (original GraphS4mer). Decaying KNN graph weight tends to result in decreased performance than using a fixed KNN graph weight (i.e., original GraphS4mer).

G Effect of end-to-end training

In our experiments in Tables 1–3, all of the layers in GraphS4mer are trained end-to-end. Here, we examine the effect of end-to-end training. Specifically, for each task, we pre-trained S4 on the task, extracted S4 embeddings from the pre-trained S4, and trained GSL and GNN layers using the S4 embeddings as inputs. Figure 8 shows the model performance with pre-trained S4 (frozen) and original end-to-end training. Pre-training and freezing S4 decrease the model performance by a large margin, suggesting that end-to-end training is needed to learn S4 embeddings that are useful for learning graph structures and graph representations.

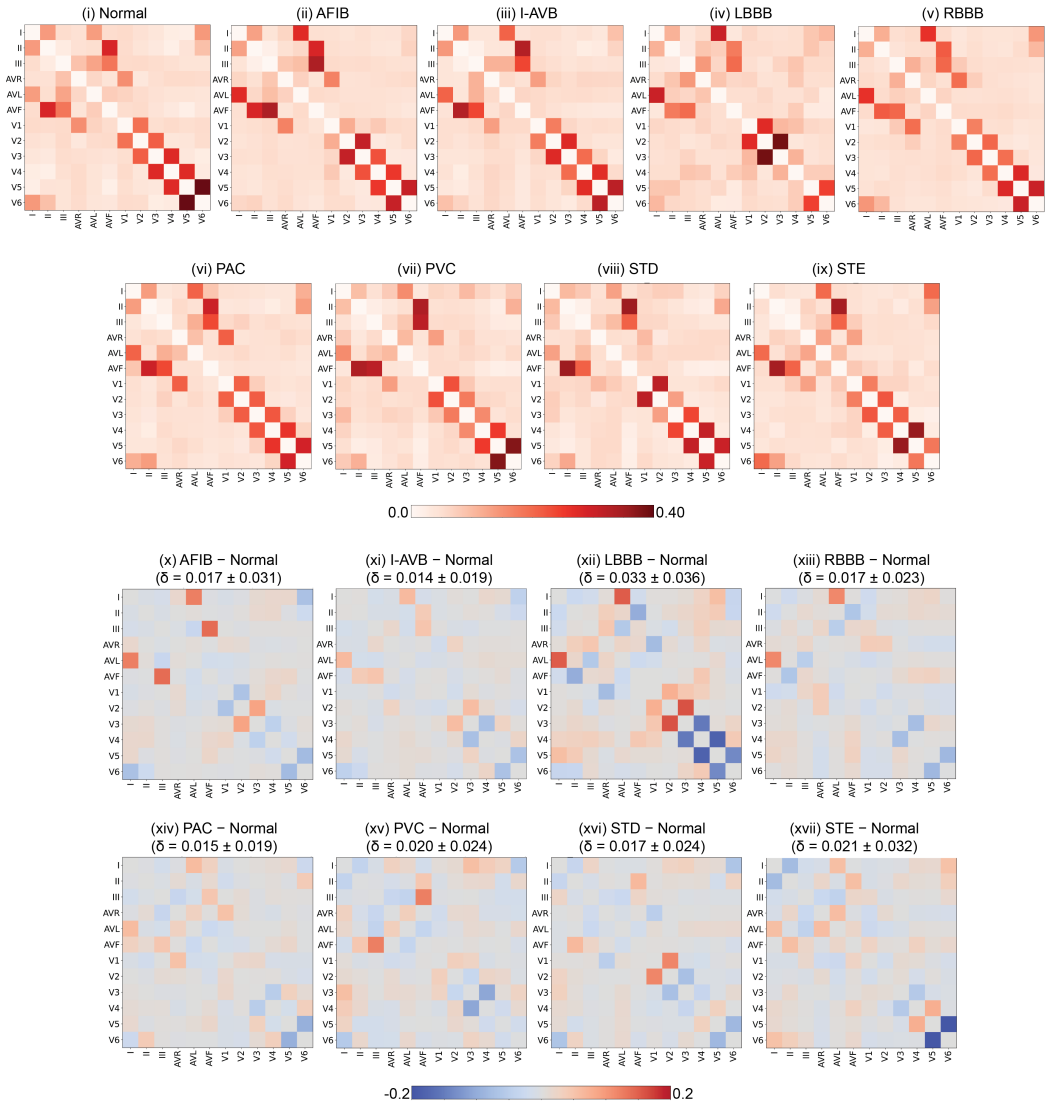

Figure 4: **(i)–(ix)** Mean adjacency matrices for ECG signals for nine ECG classes in correctly predicted test samples. **(x)–(xvii)** Difference between adjacency matrices of abnormal ECG classes and normal class. δ indicates the mean and standard deviation of the absolute values of the differences between the abnormal class and the normal class. Self-edges (i.e., diagonal) are not shown here. *Abbreviations*: AFIB, atrial fibrillation; I-AVB, first-degree atrioventricular block; LBBB, left bundle branch block; RBBB, right bundle branch block; PAC, premature atrial contraction; PVC, premature ventricular contraction; STD, ST-segment depression; STE, ST-segment elevated.

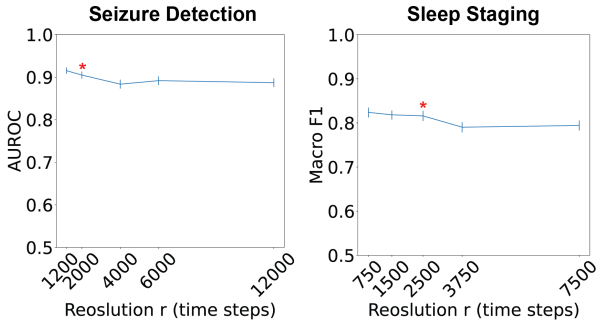

Figure 5: Model performance versus temporal resolution using the run with median performance. For convenience, we assume that temporal resolution is chosen so that the sequence length is divisible by the resolution. Asterisk indicates the temporal resolution used to report results for GRAPHS4MER in Tables 1–2 and Table 4. Error bars indicate 95% confidence intervals obtained using bootstrapping with 5,000 replicates with replacement.

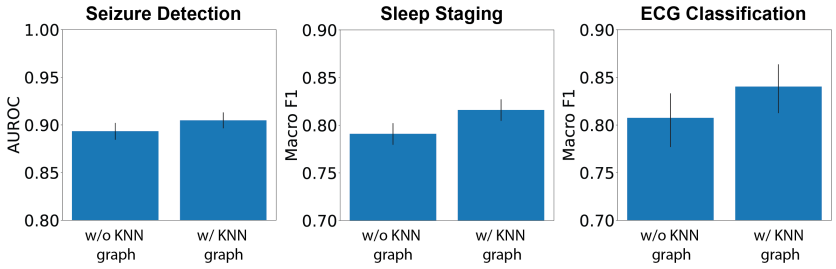

Figure 6: Model performance without and with KNN graph in GSL using the run with median performance. Error bars indicate 95% confidence intervals obtained using bootstrapping with 5,000 replicates with replacement.

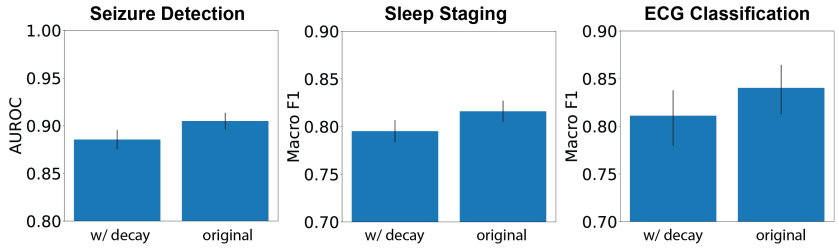

Figure 7: Model performance with and without KNN graph weight decay using the run with median performance. Error bars indicate 95% confidence intervals obtained using bootstrapping with 5,000 replicates with replacement.

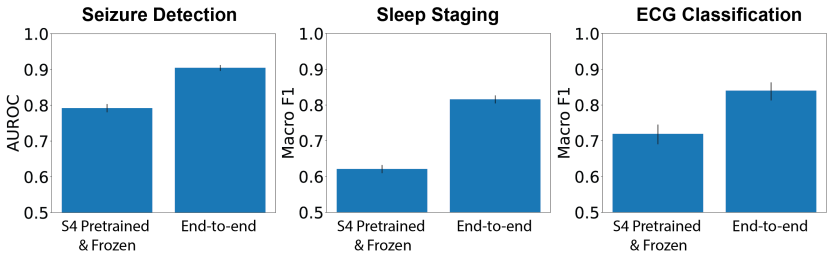

Figure 8: Model performance with pre-trained S4 versus original end-to-end training. Error bars indicate 95% confidence intervals obtained using bootstrapping with 5,000 replicates with replacement.