Dynamic Graph Representation Learning for Spatio-Temporal Neuroimaging Analysis

Rui Liu, *Member, IEEE*, Yao Hu, *Member, IEEE*, Jibin Wu, *Member, IEEE*, Ka-Chun Wong, Zhi-An Huang, *Member, IEEE*, Yu-An Huang, *Member, IEEE*, and Kay Chen Tan, *Fellow, IEEE*

***Abstract*—Neuroimaging analysis aims to reveal the information-processing mechanisms of the human brain in a noninvasive manner. In the past, graph neural networks (GNNs) have shown promise in capturing the non-Euclidean structure of brain networks. However, existing neuroimaging studies focused primarily on spatial functional connectivity, despite temporal dynamics in complex brain networks. To address this gap, we propose a spatio-temporal interactive graph representation framework (STIGR) for dynamic neuroimaging analysis that encompasses different aspects from classification and regression tasks to interpretation tasks. STIGR leverages a dynamic adaptive-neighbor graph convolution network to capture the interrelationships between spatial and temporal dynamics. To address the limited global scope in graph convolutions, a self-attention module based on Transformers is introduced to extract long-term dependencies. Contrastive learning is used to adaptively contrast similarities between adjacent scanning windows, modeling cross-temporal correlations in dynamic graphs. Extensive experiments on six public neuroimaging datasets demonstrate the competitive performance of STIGR across different platforms, achieving state-of-the-art results in classification and regression tasks. The proposed framework enables the detection of remarkable temporal association patterns between regions of interest based on sequential neuroimaging signals, offering medical professionals a versatile and interpretable tool for exploring task-specific neurological patterns. Our codes and models are available at https://github.com/77YQ77/STIGR/.**

***Index Terms*—Contrastive learning, electroencephalography (EEG), functional near-infrared spectroscopy (fNIRS), graph neural networks, interpretable visualization, magnetic resonance imaging (MRI), self-attention, spatio-temporal dynamics.**

I. Introduction

HUMAN brain is a highly intricate neurobiological system that plays a critical role in coordinating human behavior and cognition. Understanding the structure and function of the human brain network has become an enthralling endeavor with a variety of purposes, including mental disease diagnosis [1], brain–computer interface [2], and neuromorphic computing. Recent advancements have enabled researchers to measure brain activity using various neuroimaging techniques, such as magnetic resonance imaging (MRI), electroencephalography (EEG), functional near-infrared spectroscopy (fNIRS), etc. Functional magnetic resonance imaging (fMRI) detects variations in blood-oxygen-level-dependent signals using magnetic fields, EEG records spontaneous electrical activity of the brain, and fNIRS measures blood flow signals through detectors placed on the scalp [3]. Despite their ability to capture brain activity with high spatial and temporal resolution [4], there is currently no universal method to unify the analysis of complex spatio-temporal dynamics from various neuroimaging data. Especially since different types of neuroimaging data require different analysis tools, it is challenging to integrate these techniques into a cross-platform analysis system [5]. To this end, we propose a cross-platform analysis framework that can be applied to different downstream tasks, such as classification, regression, and interpretation, offering a comprehensive understanding of the human brain network.

Graph Neural Network (GNN), as shown in Fig. 1(B3), is a powerful model for transforming and analyzing graph-structured data while maintaining the graph property (including nodes, edges, and global context) and permutation invariances [7]. The intrinsic graph-structured nature of the brain has led to the widespread use of GNNs to learn the representations of the brain network from neuroimaging data. Recent studies have demonstrated the potential of GNNs in addressing heterogeneity across different neuroimaging data, making the GNN-based graph representation an ideal and general solution for analyzing such data. For example, when GNNs were applied to predict depression based on fMRI, EEG, and fNIRS data, they all focused on functional changes occurring in the prefrontal cortex [5], [8], [9]. Although these

Received 23 April 2024; revised 7 September 2024, 14 November 2024, and 6 January 2025; accepted 12 January 2025. Date of publication 4 February 2025; date of current version 7 March 2025. This work was supported in part by the National Nature Science Foundation of China under Grant 62202399, Grant U21A20512, and Grant 6220239; in part by the Research Grants Council of the Hong Kong, SAR, under Grant PolyU11211521, Grant PolyU15218622, Grant PolyU15215623, and Grant C5052-23G; in part by the Fundamental Research Funds for the Central Universities under Grant G2023KY05102; in part by the General Program of National Natural Science Foundation of China under Grant 62472353; and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515011984 and Grant 2023A151514013. This article was recommended by Associate Editor J.-H. Xue. *(Corresponding authors: Zhi-An Huang; Yu-An Huang.)*

Rui Liu, Yao Hu, Jibin Wu, and Kay Chen Tan are with the Department of Data Science and Artificial Intelligence, The Hong Kong Polytechnic University, Hong Kong, SAR (e-mail: ruiliu@polyu.edu.hk; echo-yao.hu@polyu.edu.hk; jibin.wu@polyu.edu.hk; kaychen.tan@polyu.edu.hk).

Ka-Chun Wong is with the Department of Computer Science, City University of Hong Kong, Hong Kong, SAR (e-mail: kc.w@cityu.edu.hk).

Zhi-An Huang is with the Department of Computer Science, City University of Hong Kong (Dongguan), Dongguan 523000, China (e-mail: huang.za@cityu-dg.edu.cn).

Yu-An Huang is with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710000, China (e-mail: yuanhuang@nwpu.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2025.3531657.

Digital Object Identifier 10.1109/TCYB.2025.3531657

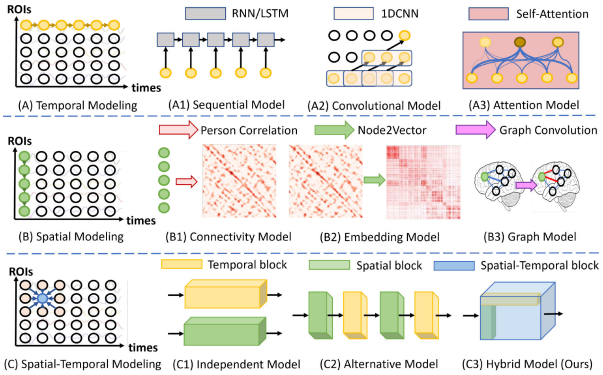

Fig. 1. Overview of different types of spatial and temporal modeling techniques (inspired by [6]). (a) Temporal modeling aims to capture dependencies between activities at different time steps. (A1)–(A3) illustrate commonly used temporal modeling techniques. (b) Spatial modeling focuses on exploring the correlations of activities at different brain regions. (B1)–(B3) illustrate commonly used spatial modeling techniques. (c) Spatio-temporal modeling captures and fuses both spatial and temporal information. Specifically, (C1) represents the independent mode and models spatial and temporal representations separately using different network blocks that work in parallel. (C2) represents the alternative mode, which models the spatial and temporal representations using separate network blocks in sequence. and (C3) represents the hybrid mode, which jointly models spatial and temporal representations using a consolidated network block (e.g., our STIGR).

studies have demonstrated the potential of GNNs in the analysis of neuroimaging data, most of them are grounded in a static manner like spatial modeling methods [Fig. 1(B1) and (B2)] [10]. Growing evidence suggests that functional brain activity can vary rapidly and involve different activated brain regions, providing important information about brain function [11]. Some recent attempts, e.g., Liu et al. [12], have combined dynamic FC analysis with static graph convolution for brain network analysis. Compared with static spatial modeling, the dynamic temporal modeling paradigm can capture the inherent temporal dynamics of brain activity [13], providing a richer and more nuanced representation of the underlying neural processes. As illustrated in Fig. 1(A1)–(A3), sequential models [14], convolutional models [15], and attention models [16] have been utilized to dynamically analyze neuroimaging and have demonstrated a remarkable capability in modeling temporal dependence. Despite notable progress in this field, the aforementioned methods primarily focus on the spatial or temporal dependencies of neuroimaging signals. This oversight neglects the potential benefits of jointly exploring their interplay, which is a critical avenue for elucidating the workings of the brain.

Spatial temporal graph neural network (STGNN) is an emerging branch of GNN to handle time-series data, which has shown great promise in capturing both spatial structure and temporal dynamics in brain activity. However, directly applying STGNN to brain network analysis has some inherent limitations that need to be addressed. First, existing STGNNs and spatio-temporal modeling methods [17], [18], [19], [20], [21] typically model spatial and temporal dependencies independently or alternatively, as illustrated in Fig. 1(C1) and (C2). In the independent mode, distinct temporal and spatial modules are utilized in parallel to capture spatial and temporal dependencies, followed by their combination to comprehend the overall spatio-temporal dependencies within the data. Conversely, in the alternating mode, separate modules are employed to model temporal and spatial dependencies, alternately employing them to extract and incorporate spatio-temporal dependencies. However, both modes fail to comprehensively capture the intricate interplay between spatial and temporal dynamics in complex brain activity. For independent mode, the spatial and temporal module generates exchanges solely at the final stage of late fusion [22], overlooking crucial spatio-temporal interactions, such as how the connectome scales and brain networks evolve over time. For alternative mode, although there is an increased interaction between the spatial and temporal modules [23], this alternation neglects to model the relationship between nodes at previous and subsequent time points (i.e., the spatio-temporal interdependency in Fig. 3), which presents challenges in capturing the inter-relationship between spatial and temporal dynamics [24]. Thus, these models are limited in their ability to comprehensively investigate the interactions between spatial and temporal elements in the dynamic brain network. Second, STGNN typically performs temporal convolutions on the input dynamic graph. However, due to the limited kernel size of the convolution operation, these models are unable to capture long-range dependencies that are beyond the receptive fields of the convolution kernels [15]. Third, traditional STGNNs rely on a fixed adjacency matrix to represent the topological structure of the brain network [25]. However, brain activity is a dynamic process, and the connectivity between brain regions evolves over time. The static representation of brain topology using a fixed adjacency matrix makes it difficult to capture the temporal dependence of different brain regions when performing cognitive functions.

In this work, we propose a novel spatio-temporal interactive graph representation framework (STIGR) (as shown in Fig. 2), dubbed STIGR, for dynamic brain network analysis. STIGR comes with three main components for learning precise and discriminative spatio-temporal graph representations: 1) local dynamic graph convolution network (L-DGCN); 2) global spatio-temporal attentive network (G-STAN); and 3) contrastive-learning-based adjacent matrix learning (CL-AM). L-DGCN is designed to capture the complex inter-relationships between spatial and temporal dynamics by learning intrinsic local dynamic graph representations between adjacent time periods. G-STAN leverages the Transformer architecture to effectively capture long-range dependencies, resulting in enhanced global attentive spatio-temporal representations. By utilizing contrastive learning to adaptively learn the adjacency matrix of L-DGCN, CL-AM can accurately model the connection relationships of dynamic graphs. The major contributions of this work are summarized as follows.

1) *Pioneering Framework for Versatility Neuroimaging Analysis:* We present a promising neuroimaging analysis framework that spans multiple platforms and tasks, enabling efficient learning and representation of intricate spatio-temporal relationships within neuroimaging data. Our framework represents *the first attempt* to hybrid model the dynamic interactions of brain networks, as illustrated in Fig. 1(C3), addressing a critical gap in current research.

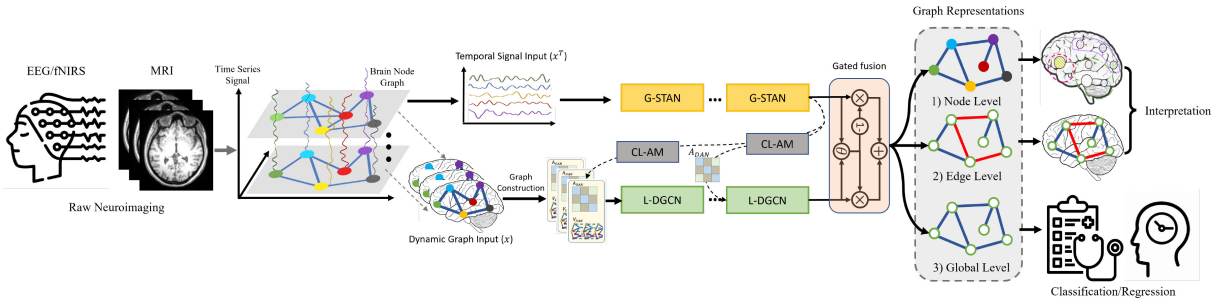

Fig. 2. STIGR framework illustration showcases its core concept as a dynamic spatio-temporal graph representation learning network based on graph convolution. By incorporating graph convolution and global attention mechanisms on the constructed dynamic spatio-temporal graphs, the network effectively learns discriminative graph representations of the neuroimaging data. Specifically, the raw neuroimaging data are first denoised to obtain temporal signal input x^{T} and dynamic graph input x. The temporal signal input x^{T} is then processed by a stack of global spatio-temporal attentive network (G-STAN), extracting global spatio-temporal representations. Simultaneously, the dynamic graph input x is divided into multiple windows, generating different brain node graphs. By utilizing contrastive learning guided by the G-STAN, the CL-AM module generates an adaptive adjacency matrix for the dynamic brain graph. Subsequently, a stack of local dynamic graph convolution networks (L-DGCNs) with the CL-AM module captures local spatio-temporal representations from the dynamic graph input x. The learned graph representations from both perspectives are fused using a gated fusion module, enabling the handling of various downstream tasks at the node, edge, and graph levels.

2) *Innovative Spatio-Temporal Components for Enhanced Learning:* We introduce two novel spatio-temporal modeling components, L-DGCN and G-STAN, which utilize graph convolution and attention mechanisms to learn discriminative spatio-temporal representations from local and global perspectives. In addition, our approach employs contrastive learning to adaptively refine the adjacency matrix of L-DGCN, guided by G-STAN, thereby enhancing the interaction between local and global learning processes.
3) *Exceptional Performance and Interpretability Across Neuroimaging Tasks:* Extensive experiments demonstrate the superior performance of our proposed STIGR in cross-platform/task neuroimaging analysis across a diverse range of modality, including MRI, EEG, and fNIRS. It outperforms State-of-the-Art (SOTA) neuroimaging analysis methods in tasks, such as mental disorder classification, motor imagery classification, and brain age prediction. Furthermore, STIGR facilitates interpretable neuroimaging analysis, allowing the exploration of discriminative patterns at various levels, including nodes, edges, and graphs.

II. Problem Definition

The objective of our research is to address a key problem in neuroimaging analysis: the development of an effective graph representation framework for a set of tasks, such as classification, regression, and interpretation. In this context, we define the input data as a set, $D = \{X, Y\}$, where X represents a series of neuroimaging time-series sequences, and Y represents the corresponding outcomes, such as diagnostic labels or clinical scores. Each sequence in X is of a fixed dimension, $N \times T$, where N denotes the spatial dimension of the neuroimaging data, such as the number of ROIs for fMRI or electrodes/channels for EEG, and T represents the number of time points across all sequences. The goal of our study is to learn a function $f : X \to Y$ that can accurately classify or regress these sequences, thus contributing to our understanding and analysis of neuroimaging data. Our objective is to apply

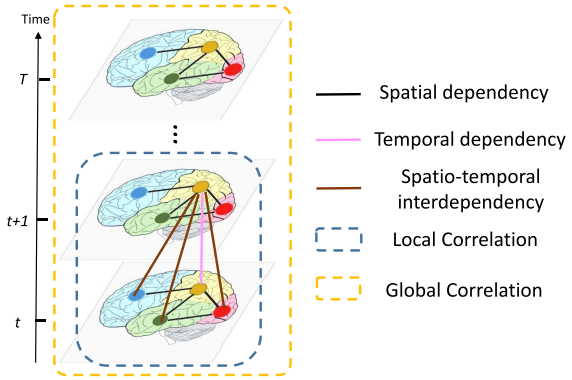

Fig. 3. Illustration of three types of dependencies and two types of correlations in spatio-temporal neuroimaging data: spatial dependency (black lines connecting different ROIs), temporal dependency (pink line representing the influence of a node on itself at the next time point), and spatio-temporal interdependency (brown lines indicating interdependencies between a node and its neighboring nodes at the next time point). Local correlation is shown by the blue dashed rectangle, representing the correlation between a dynamic graph and its temporal and spatial neighbors, encompassing spatial, temporal, and spatio-temporal dependency. Global correlation is depicted by the yellow dashed rectangle, illustrating the correlation between distant graphs.

graph-based analytical techniques to a time-series matrix $X \in \mathbb{R}^{N\times T}$, representing neuroimaging data. To facilitate this, we transform the time-series data into a graph representation. This involves constructing a dynamic graph network $G(X) = \{\mathcal{G}(1), \ldots, \mathcal{G}(T)\}$ where each graph $\mathcal{G}(t)$ at time t consists of a vertex set $V(t) = \{X_1(t), \ldots, X_N(t)\}$ denoting N nodes and an edge set $A(t)$ delineating the connectivity between nodes based on the neighborhood function $\mathcal{N}(i)$. This graph-based representation allows us to apply sophisticated algorithms designed for graph data to a set of tasks (such as classification, regression, and interpretation) for the original time-series datasets. Therefore, this problem can be formulated as an optimization task, which can be defined as follows:

$$f^* = \arg\min_f \mathbb{E}_{X,Y}\big[\mathcal{L}(f(G(X)), Y) + \mathcal{R}(f; A)\big] \tag{1}$$

where $\mathbb{E}$ denotes the expected value of the loss function $\mathcal{L}$ over the space of (X, Y), and $\mathcal{R}$ represents an additional

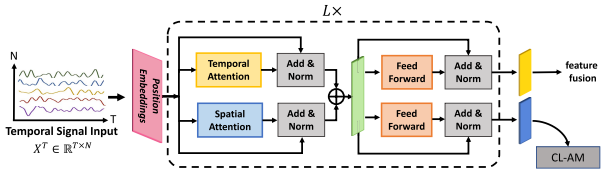

Fig. 4. Proposed G-STAN architecture comprises two parallel attention networks designed to extract global attention information from both spatial and temporal dependences in the data.

regularization term on A. It is worth noting that in this article, the superscript T in x^{T} denotes the transpose operation, whereas the subscript T in X_T denotes a temporal variable.

III. Model Design

Fig. 3 presents a visualization of the interdependencies and correlations associated with dynamic graph representation. These interdependencies encompass spatial, temporal, and spatio-temporal relationships, collectively forming our local correlations. They capture connections between a dynamic graph and its temporal and spatial neighbors, facilitating the analysis of local relationships. In addition, it is crucial to consider the global correlation, which accounts for the interdependence between distant graphs and mitigates the impact of short-term variations in neuroimaging analysis [26].

To effectively incorporate these multiple dependencies and correlations, we present the graph representation learning framework of STIGR, as depicted in Fig. 2. The STIGR comprises three main components: 1) G-STAN; 2) L-DGCN; and 3) CL-AM. The G-STAN and L-DGCN are designed to jointly learn the spatio-temporal graph representation of neuroimaging data at global and local levels, respectively. To better model the node connections between different windows, CL-AM is designed to adaptively extract the cross-window temporal connectivity in L-DGCN based on contrastive learning. The local and global spatio-temporal representations are then adaptively integrated using a gated fusion module to obtain the discriminative graph representation.

A. Global Spatio-Temporal Attentive Network

Capturing global dependency is essential to provide a broader context and understanding of the overall brain dynamics [16]. The Transformer [27] is highly effective for modeling global dependency, which explicitly computes the dependency of tokens at different time steps. However, the vanilla Transformer was designed for sequence data and did not account for the spatial dependencies (e.g., ROIs connectivity) in the brain network. To address this limitation, we propose the G-STAN module based on the Transformer encoder to learn global spatio-temporal representations by decoupling the attention mechanism into temporal and spatial dimensions.

As illustrated in Fig. 4, the G-STAN module takes the temporal signal sequence $X^{\mathsf{T}} \in \mathbb{R}^{T\times N}$ as its input and produces both temporal representations $E_T^{(L)} \in \mathbb{R}^{T\times N}$ and spatial representations $E_S^{(L)} \in \mathbb{R}^{N\times T}$ as its outputs, which serve distinct purposes for feature fusion and CL-based adjacent matrix learning, respectively. Specifically, the input sequence X^{T} is first processed by a position embedding layer to obtain the hidden feature representation $E \in \mathbb{R}^{T\times N}$ that encodes position information. The position embedding is implemented based on [28] and can be formulated as follows:

$$E = \text{PositionEmbedding}\left(X^{\mathsf{T}}\right) = X^{\mathsf{T}} + \text{PE} \tag{2}$$

where the position embedding PE is represented as a learnable table of vectors, which assigns a unique vector representation to each ROI signal based on its position index of the brain. Then, G-STAN utilizes a dual self-attention structure with L layers to separately extract the attention focus from the temporal and spatial dimensions of E. In particular, temporal attention selectively fuses information from different time points of the same node (e.g., ROI of fMRI or channel of EEG), while spatial attention fuses information from different nodes that are collected at the same time point. In the temporal attention block, we learn three matrix representations Q, K, and V (queries, keys, and values) given E as

$$Q = f_Q(E), K = f_K(E), V = f_V(E) \tag{3}$$

where f_Q, f_K, and f_V are the corresponding projection functions of *queries*, *keys*, and values. The dot-product similarity is then used to compare *queries* with *key-value* pairs to obtain the attention distribution on the *values*. If the *queries* and the *keys* are similar (i.e., high attention weight), the corresponding *values* are considered related. The resulting weighted values matrix becomes the output of the attention block. Following [27], multihead attention (MHA) is adopted to extract temporal-level attention, projecting the *queries*, *keys*, and *values* with h different linear projection heads. The output representation of the temporal attention block $\tilde{E}_T$ can be formulated as follows:

$$\tilde{E}_T = \text{MHA}(E) = \langle \text{head}_1, \ldots, \text{head}_h \rangle W^o \tag{4}$$

$$\text{head}_j = \text{Att}_j(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{N}}\right)V \tag{5}$$

where $W^o \in \mathbb{R}^{N\times N}$ represents the linear transformation matrices of the head concatenation. Distinct from conventional temporal convolution layers, G-STAN leverages a self-attention based on the Transformer encoder to capture long-term temporal dependencies in time-series data. Without convolution layers, this self-attention module can learn the significance of each time point in the context of the entire sequence by attending to different positions of the input sequence.

Likewise, the spatial attention block is constructed in a similar way to the temporal attention block, but with the spatial-level Q, K, and V matrices learned over the transpose of the input embedding feature (i.e., $E^{\mathsf{T}} \in \mathbb{R}^{N\times T}$) to output the spatial attention for all nodes at the same time point (denoted as $\tilde{E}_S$). Since the temporal and spatial attention blocks operate independently to calculate $\tilde{E}_T$ and $\tilde{E}_S$, then element-wise addition $\oplus$ is used to capture the interdependencies between spatial and temporal representations, i.e., $\tilde{E} = \tilde{E}_T \oplus \tilde{E}_S$. The output is then passed through two parallel feed-forward networks, allowing our network to extract spatio-temporal interdependencies while exploring modality-specific intradependencies. Following the design principle of

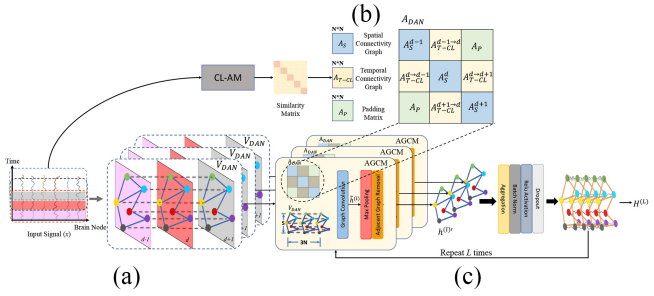

Fig. 5. (a) Input illustration for the dynamic adaptive-neighbor graph generated along the time axis. (b) Adjacency matrix of the dynamic adaptive-neighbor graph in (a). (c) Workflow of L-DGCN and architecture of AGCM.

Transformer, we apply residual connection with layer normalization, abbreviated as *Add&Norm*, after the dual attention and feedforward layers. We stack L spatio-temporal attention layers to successively update the fused embeddings. Finally, G-STAN outputs temporal and spatial representations $E_T^{(L)}$ and $E_S^{(L)}$ for further global-local feature fusion and CL-based adjacent matrix learning, respectively.

B. Local Dynamic Graph Convolution Network

To capture the full range of spatial, temporal, and spatio-temporal interdependencies inherent in dynamic brain activity, we introduce a novel dynamic adaptive-neighbor graph construction ($\mathcal{G}_{\text{DAN}} \in \{V_{\text{DAN}}, A_{\text{DAN}}\}$) to be used in our L-DGCN, as shown in Fig. 5(a) and (b). Specifically, we divide the T time points into D scanning windows with a length of S each ($D = \lfloor T/S \rfloor$). Our graph convolution approach distinguishes itself from standard spatial convolutions by forming connections between nodes across adjacent temporal windows, generating localized spatio-temporal subgraphs adept at capturing short-term dynamics in brain networks. Such a dynamic adaptive-neighbor graph construction extends beyond individual spatial graphs, linking them across time to map intricate spatio-temporal interdependencies.

The scale of $\mathcal{G}_{\text{DAN}}$ is determined by the number of adjacent scanning windows. For example, we set it to 1 as shown in Fig. 5, resulting in $V_{\text{DAN}} = [V[d-1], V[d], V[d+1]] \in \mathbb{R}^{3N \times S}$ [as shown in Fig. 5(a)] and A_{DAN} with the size of $3N \times 3N$ [as shown in Fig. 5(b)]. Inspired by the recent advance in [24], we leverage A_{DAN} to record three types of adjacency matrix $\in N \times N$: 1) spatial connectivity graph matrix; 2) temporal connectivity graph matrix; and 3) padding matrix, as follows.

1) The spatial connectivity graph matrix (A_S) represents the node connections within dynamic subgraphs in each adjacent scanning window and is positioned on the diagonal of A_{DAN}. The construction of A_S is based on the calculation of the Pearson correlation coefficient (PCC). Specifically, the A_S for the dth window is computed as follows:

$$A_S^d(i,j) = \frac{\sum_{k=1}^{S}\left(v_d(i,k) - \overline{v_d(i)}\right)\left(v_d(j,k) - \overline{v_d(j)}\right)}{\sqrt{\sum_{k=1}^{S}\left(v_d(i,k) - \overline{v_d(i)}\right)^2\left(v_d(j,k) - \overline{v_d(j)}\right)^2}} \quad (6)$$

where i and j represent the ith and jth nodes, respectively. Notably, we selected the top 10% of the spatial correlation in A_S to reduce redundant information. In this approach, we define significant FCs as those with a PCC exceeding the threshold $\tau = 0.1$), which are assigned a value of 1, while all other correlations are assigned a value of 0.

2) The temporal connectivity graph matrix (A_{T-CL}), which works in conjunction with the spatial graph matrix A_S, is designed to capture the temporal relationships between nodes across different subgraphs. The CL-AM module plays a pivotal role in adaptively learning the construction of A_{T-CL}, a process that we detail in Section III-B3) for further clarity.
3) The padding matrix (A_P) is included to fill the empty space left after dividing the temporal signal into adjacent scanning windows. It is constructed by setting all its entries to 0 and is used for padding operations.

In this way, $\mathcal{G}_{\text{DAN}}$ can extract the local spatio-temporal correlations of dynamic graph matrices by the following adjacent graph convolution module (AGCM).

We further extend our analysis to the entire time course and employ AGCM to capture hybrid spatio-temporal relations among the entire dynamic brain network, as shown in Fig. 5(c). The L-DCGN module accepts a graph input signal $X \in \mathbb{R}^{N \times T}$ as its input and outputs a learned graph representation $H^{(L)} \in \mathbb{R}^{N \times T}$. Utilizing the proposed dynamic graph construction method, the input graph signal X is segmented into D subgraphs, each denoted as $\mathcal{G}_{\text{DAN}} \in \{V_{\text{DAN}}, A_{\text{DAN}}\}$. Subsequently, for each subgraph $\mathcal{G}_{\text{DAN}}$, a graph convolution operation aggregates the central node's features with those of its neighboring nodes across adjacent windows to form a comprehensive hybrid graph representation. For the lth input graph representation of d subgraph, denoted as $h_d^{(l)} \in \mathbb{R}^{3N \times S}$, the graph convolution operation is applied or performed as

$$\tilde{h_d^{(l)}} = \sigma\left(A_{\text{DAN}}(h_d^{(l)}W + b)\right) \quad (7)$$

where $W \in \mathbb{R}^{S \times S}$ and $b \in \mathbb{R}^S$ are learnable parameters, and σ represents the *ReLU* activation function. After graph convolution, max pooling is performed over the time dimension to aggregate the representation from adjacent windows, resulting in a graph representation $h_d^{(l)'}$ of size $N \times S$. Each scanning window of the spatio-temporal graph input $H^{(l)}$ is processed in parallel by AGCM, generating the local graph representations $h_d^{(l)'}$ (where $d \in [1, D]$) as shown in Fig. 5(c). It should be noted that zero-padding is applied when processing the first and last scanning windows. Then, an aggregation layer is employed to concatenate $h_d^{(l)'}$ as follows:

$$H^{(l)'} = \left\langle h_1^{(l)'}, \ldots, h_D^{(l)'} \right\rangle \in \mathbb{R}^{D \times N \times S} \quad (8)$$

where $\langle . \rangle$ denotes the concatenation operator. Furthermore, a batch normalization layer, *ReLU* function, and a dropout layer are cascaded to generate the final local spatio-temporal graph representation $H^{(l+1)}$ of layer l. The final output after L layers of L-DGCN is represented as $H^{(L)}$.

C. Contrastive-Learning-Based Adjacent Matrix

To effectively capture the dynamic hybrid graph representation of each central node and its neighboring nodes across

adjacent windows, it is essential to model the node connections within and between different windows simultaneously. Generally, the spatial connectivity graph matrix A_S can be calculated using static coherency-based measurements (e.g., PCC) to model the degree of similarity between connectome nodes within the same window. However, modeling node connections between different windows A_{T-CL} is rather challenging due to their complex cross-temporal connectivity characteristics. Therefore, we propose a CL-AM module based on contrastive learning, which contrasts the similarity of learned representations between all scanning windows to model cross-temporal connectivity without label information.

The intuition behind contrastive learning is to pull similar samples closer and push away dissimilar samples, based on their representation. Here, we classify adjacent scanning windows at the same node as "positive" data points, indicating they belong to the same category. In contrast, scanning windows at different nodes are labeled as "negative" data points, indicating they belong to different categories. The representation of scanning windows can be extracted by G-STAN. As a result, G-STAN acts as a shared encoder for the CL-AM module to encode the underlying shared information between different scanning windows.

In addition to the perspective of similarity of features, contrastive learning in our work can also be interpreted from the view of contrastive predictive coding which involves learning representations that predict future data points from past observations [29]. In our CL-AM module, we leverage this principle by using embeddings from one session as context to predict the embeddings of the next session, assuming temporal continuity in data. Our goal is to make the embeddings of the corresponding nodes in adjacent sessions more similar to each other (values close to 1 on the diagonal of the similarity correlation matrix, *SC*) than to the embeddings of other nodes at different time points (values close to 0 for off-diagonal elements). This encourages the model to capture both intrasession node distinctions and the temporal evolution of nodal features. The contrastive loss function plays a key role by minimizing the distance between positive pairs (same node across sessions) and maximizing the distance between negative pairs (different or noncorresponding nodes across sessions). Consequently, the *SC* matrix, derived from these trained embeddings, reflects the degree of similarity between nodes across sessions. Higher similarity indicates a strong spatiotemporal connection, implying predictability of node states over time. Consequently, the final *SC* serves as a proxy for A_{T-CL}, enabling the construction of a dynamic graph that reflects the underlying spatiotemporal connectivity patterns.

As depicted in Fig. 6, the CL-AM module processes the spatial-level representation $E_S^{(l)} \in \mathbb{R}^{N\times T}$ as input and produces a learned similarity matrix SC as output based on contrastive learning. Specifically, the spatial-level representation $E_S^{(l)}$ is divided into D sequences of length S to learn the pair-wise temporal connectivity. For example, to learn $A_{T-CL}^{d-1\to d}$, we first encode the feature representations of the previous window $d-1$ ($E_S^{(d-1)} \in \mathbb{R}^{N\times S}$) and the current window d ($E_S^{(d)} \in \mathbb{R}^{N\times S}$) using the spatial-level attention network of G-STAN. We then feed the representations of the two scanning windows into two

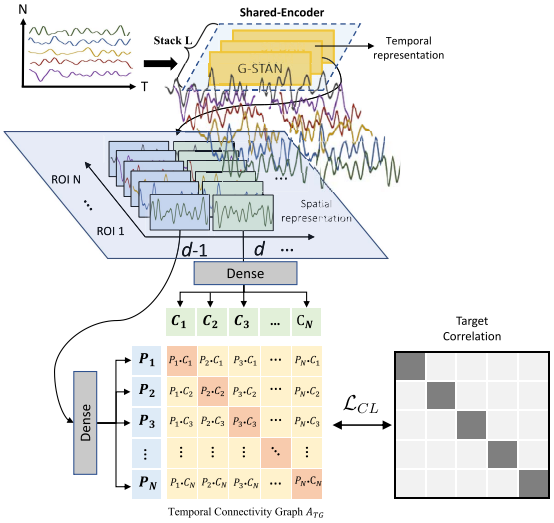

Fig. 6. Structure of the CL-AM, where the spatial network in G-STAN serves as a shared encoder.

different linear projection heads to map the representations to a latent space for similarity calculation. Therefore, we obtain two embedding vectors of N dimensions that represent the regional features of the window $d-1$ and the window d, denoted as P and C, respectively.

The CL-AM module is trained to predict which of the $N \times N$ possible pairings actually occurred by taking the embeddings of N nodes in two windows as samples for contrastive learning. The CL-AM maximizes the similarity of the previous and current embeddings of the N positive pairings while minimizing the similarity of the embeddings of the rest $\{N^2 - N\}$ negative pairings. The total number of A_{T-CL} that needs to be calculated in the learning process is denoted as M. Given the ith and jth nodes, the contrastive loss function of mth A_{T-CL} is calculated as follows:

$$\mathcal{L}_{CL}^m = \sum_i (1 - \mathrm{SC}_{ii})^2 + \lambda \sum_i \sum_{i\neq j} \mathrm{SC}_{ij}^2 \tag{9}$$

where λ is a positive constant that trades off the importance of the positive pairs (first term) and negative pairs (second term). The similarity coefficient SC_{ij} is computed between P_i and C_j as follows:

$$\mathrm{SC}_{ij} = \frac{P_i C_j}{\|P_i\|_2 \cdot \|C_j\|_2} \tag{10}$$

where $\|.\|_2$ is l^2-norm. By optimizing $\mathcal{L}_{CL}^m$, the temporal connectivity graph matrix $A_{T-CL}^{d-1\to d}$ can be well represented by the learned *SC* for dynamic graph convolution. The total contrastive loss is defined as $\mathcal{L}_{CL} = \sum_{m=1}^{M} \mathcal{L}_{CL}^m$ for the processing of L-DGCN, where m indicates the total number of A_{T-CL} needed to calculate in the learning process. The overall contrastive loss $\mathcal{L}_{CL}$ can be regarded as a regularization term in (1) that serves to optimize A_{T-CL}.

D. Overall Architecture

Algorithm 1 outlines the program flow of STIGR. First, we reshape the input dynamic graph signal $X \in \mathbb{R}^{N\times T}$

Algorithm 1 Pseudocode of the STIGR

Require: Preprocessed data X, and label Y
Ensure: Predicted probabilities of testing set Y_{pred}^{te}

$[X^{tr}, X^{te}, Y^{tr}, Y^{te}] \leftarrow$ Split(X, Y)
Initialize STIGR.
for $e = 1, \ldots,$ *epochs* **do**
  // Global Spatio-Temporal Attentive Network
  $E =$ PositionEmbedding $(X^{tr\mathsf{T}})$
  $E_T{}^{(0)} = E$; $E_S{}^{(0)} = E^{\mathsf{T}}$
  for $l = 0, \ldots, L-1$ **do** // l: # of attention layers
    $\tilde{E}^{(l)} =$ MHA$(E_S^{(l)}) \oplus$ MHA$(E_T^{(l)})$
    $E_T^{(l+1)} =$ AddNorm(FFN$(\tilde{E}^{(l)})$, $\tilde{E}^{(l)}$)
    $E_S^{(l+1)} =$ AddNorm(FFN$(\tilde{E}^{(l)\mathsf{T}})$, $\tilde{E}^{(l)\mathsf{T}}$)
  end for
  // Local Dynamic Graph Convolution Network
  $H^{(0)} = X^{tr}$
  for $l = 0, \ldots, L-1$ **do** // l: # of L-DGCN layers
    $A_{T-CL}^{(l)} =$ CL-AM$(E_S{}^{(l)})$
    $A_{DAN}^{(l)} = \left\langle A_S, A_{T-CL}^{(l)}, A_P \right\rangle$
    $H^{(l+1)} =$ L-DGCN$(H^{(l)}, A_{DAN}^{(l)})$
  end for
  $R = (1-\theta)E_T^{(L)} \oplus \theta H^{(L)}$ // Fused representation
  // Tower networks for specific tasks
  $output \leftarrow$ Tower(R)
  $Loss \leftarrow \alpha\mathcal{L}(output, Y^{tr}) + (1-\alpha)\mathcal{L}_{CL}$
  STIGR.update($Loss$)
end for
$Y_{pred}^{te} \leftarrow$ STIGR.predict (X^{te})

Note: The $\langle . \rangle$ represents the construction of the adjacency matrix for the DAN graph.

into $X^{\mathsf{T}} \in \mathbb{R}^{T\times N}$ and pass it through the L layer G-STAN module. This module learns the global spatial and temporal representations $E_S^{(L)}$ and $E_T^{(L)}$, respectively. Based on the learned $E_S^{(L)}$, we then employ the CL-AM module to adaptively learn the temporal connectivity matrix A_{T-CL} to construct the adjacent matrix A_{DAN}. The dynamic graph signal X and learned adjacent matrix A_{DAN} are subsequently fed into the L layer L-DGCN module to derive the local spatio-temporal representation $H^{(L)}$. Upon obtaining the final outputs $E_T^{(L)}$ and $H^{(L)}$, the gated fusion module is used to combine the local and global spatio-temporal representations as $R = (1-\theta)E_T^{(L)} \oplus \theta H^{(L)}$, where θ is a learnable parameter. The fused feature embeddings R are then directed to different output networks for various downstream tasks. For classification, the output tower network is composed of a global average pooling layer that aggregates node-level features to represent the entire graph at a global level and two dense layers that perform binary classification. The contrastive loss acts as a regularization effect on the representation learning process, resulting in the overall optimization function in (1)

$$f^* = \arg\min_f \mathbb{E}_{X,Y}\left[\alpha\mathcal{L}(f(G(X)), Y) + (1-\alpha)\mathcal{L}_{CL}\right] \quad (11)$$

where α is a tunable parameter that balances the weights of the losses.

IV. Experiments and Results

A. Data Acquisition

ABIDE I & II[1]: ABIDE I comprises 1035 valid resting-state fMRI samples aggregated from 17 different brain imaging sites, including 505 autism spectrum disorder (ASD) subjects and 530 typical controls (TCs). ABIDE II dataset is also a multisite dataset containing 1113 resting-state fMRI from 19 different sites, comprising 521 ASD participants and 592 TCs.

ADHD-200[2]: ADHD-200 includes 939 preprocessed resting-state fMRI samples from 8 imaging sites, involving 358 children and adolescents with ADHD and 581 TCs.

COBRE [3]: The COBRE dataset consists of 146 resting-state fMRI samples involving 72 schizophrenia (SCZ) and 74 TCs.

BCICIV_2a[4]: BCICIV_2a is a motor imagery dataset with training and testing sets based on EEG recordings collected from nine subjects. The dataset includes four different motor imagery tasks, using a 22-channel EEG.

BCIC2015[5]: BCIC2015 is an EEG dataset obtained from 26 healthy participants who completed an error detection task based on the P300-Speller paradigm. The objective of this experiment is to utilize the analysis of 55 channels of EEG signals following the receipt of feedback to identify instances.

fNIRS-BCI[6]: fNIRS-BCI consists of fNIRS recordings obtained from a total of eight individuals, including three males and five females. During the experiments, a 52-channel fNIRS system was used to record changes in oxygenated hemoglobin and deoxygenated hemoglobin in the prefrontal cortex.

The performance of the experiments was evaluated in terms of accuracy (ACC), sensitivity (SEN), and specificity (SPE). A more detailed description of the model and parameters settings can be found in the supplementary Section I.

B. Model Comparison

In this section, we evaluate the performance of STIGR on fMRI, EEG, and fNIRS datasets for preliminary classification tasks, and compare it with SOTA methods, as shown in Fig. 7. Since most SOTA methods do not provide open-source code, we ensured fairness by directly comparing the performance of our models with the results reported in the respective papers. For fMRI datasets, we achieve competitive accuracy of 72.73(±3.37)% (SEN: 76.08%, SPE: 68.84%, and AUC: 79.03%) on ABIDE I, 72.53(±3.59)% (SEN: 73.00%, SPE: 71.56%, and AUC: 77.59%) on ABIDE II, 76.15(±3.74)% (SEN: 79.47%, SPE: 72.35%, and AUC: 81.75%) on ADHD-200, and 88.42(±4.9)%(SEN: 89.92%, SPE: 87.29%, and AUC: 92.44%) on COBRE, respectively. Notably, these datasets are cross-site and collected from different medical institutions with varying scanning equipment and protocols, resulting in significant data heterogeneity that can significantly affect the accuracy of the analysis [36].

[1] http://fcon_1000.projects.nitrc.org/indi/abide
[2] http://fcon_1000.projects.nitrc.org/indi/adhd200/
[3] http://fcon_1000.projects.nitrc.org/indi/retro/cobre.html
[4] https://www.bbci.de/competition/iv/
[5] https://www.kaggle.com/c/inria-bci-challenge
[6] http://bnci-horizon-2020.eu/database/data-sets

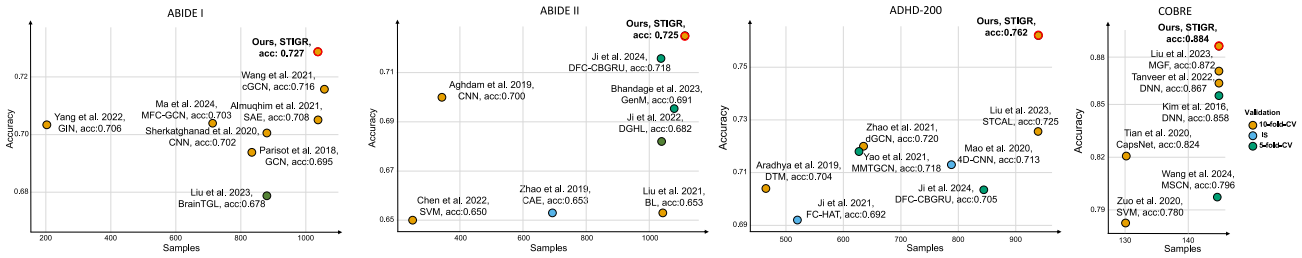

Fig. 7. Performance comparison with the SOTA methods on fMRI datasets of ABIDE I, ABIDE II, and ADHD-200 (AUCs are stated in main text).

TABLE I
PERFORMANCE EVALUATION ON BCICIV_2A DATASET

Section A: Performance comparison with the SOTA methods on BCICIV_2a dataset											
Method	task	A01	A02	A03	A04	A05	A06	A07	A08	A09	mean
Amin et al. 2019 [30]	L/R	78.10	61.10	79.88	69.08	94.09	71.49	91.70	95.85	94.10	81.71
	L/F	85.07	71.18	80.25	86.11	87.14	80.92	82.67	83.02	91.71	83.12
	L/T	88.94	79.88	87.54	85.11	86.47	78.06	87.51	92.35	98.86	86.75
	R/F	88.88	74.31	75.00	83.67	85.79	72.25	79.89	78.15	78.47	79.60
	R/T	98.24	75.71	91.31	77.04	82.24	76.03	86.48	84.36	87.52	84.33
	F/T	80.25	85.79	84.70	77.77	73.73	80.60	85.81	80.58	92.72	83.55
	mean	86.58	74.66	83.11	79.80	86.58	76.23	85.68	85.72	90.23	83.18
Bang et al. 2021 [31]	L/R	90.26	68.47	95.46	78.47	91.32	72.59	94.80	97.22	94.10	86.96
	L/F	97.91	83.31	90.31	87.16	81.92	81.22	97.91	89.29	95.14	89.35
	L/T	94.78	72.23	95.52	83.65	85.79	76.75	97.92	93.74	98.61	88.78
	R/F	95.84	77.02	93.05	89.62	88.23	78.84	97.56	89.58	84.41	88.24
	R/T	97.55	73.28	96.18	85.81	87.84	74.35	96.19	88.57	93.73	88.17
	F/T	76.07	75.05	82.67	77.76	75.71	78.82	85.11	91.00	90.28	81.39
	mean	92.07	74.89	92.20	83.74	85.14	77.09	**94.92**	91.57	**92.71**	87.15
Gaur et al. 2021 [32]	L/R	86.81	64.58	95.83	67.36	68.06	67.36	80.56	97.22	92.36	80.02
	L/F	97.22	63.89	93.06	82.64	65.97	70.83	97.92	84.2	96.53	83.64
	L/T	97.22	65.97	94.44	88.19	74.31	72.22	93.75	92.36	97.22	86.19
	R/F	97.22	80.56	93.06	89.58	70.83	64.58	93.75	88.19	84.03	84.64
	R/T	100	66.67	94.44	86.81	65.97	71.53	93.75	89.58	78.47	83.49
	T/F	69.44	73.61	69.44	62.50	68.06	70.14	79.86	82.64	85.42	72.99
	mean	91.32	69.21	90.05	79.51	68.87	69.44	89.93	88.43	89.70	81.83
Das et al. 2022 [33]	L/R	95.14	58.33	97.92	79.92	84.03	65.28	85.42	95.83	93.75	83.96
	L/F	97.92	77.78	95.14	85.42	68.06	63.19	99.31	93.75	95.14	86.19
	L/T	97.92	68.75	95.14	86.11	80.56	75.69	97.22	95.14	97.92	88.27
	R/F	99.31	80.56	95.83	85.42	77.78	55.56	100.0	90.97	85.42	85.65
	R/T	100.0	69.44	96.53	81.94	82.64	70.14	97.22	89.58	72.22	84.41
	T/F	75.69	79.86	79.86	68.75	61.11	66.67	88.19	86.81	88.19	77.24
	mean	94.33	72.45	93.40	81.26	75.70	66.08	94.56	92.01	88.77	84.28
Ours	L/R	94.44	77.08	98.26	83.33	88.27	81.25	76.39	96.53	91.32	86.42†
	L/F	97.22	89.58	96.88	88.19	80.21	84.72	96.53	92.01	97.22	91.40†
	L/T	97.57	81.60	94.44	89.58	82.99	82.99	94.10	93.40	98.61	90.59†
	R/F	97.57	88.58	96.88	91.32	79.17	80.91	97.22	95.14	84.38	90.24†
	R/T	98.96	82.64	98.96	84.38	82.99	82.99	96.53	92.01	90.63	90.01†
	T/F	86.81	90.82	86.48	79.86	78.13	80.56	92.71	89.58	89.93	86.04†
	mean	**95.43**	**87.43**	**95.31**	**86.11**	**80.44**	**82.24**	92.25	**93.11**	92.02	**89.28**†
Section B: Performance results on BCICIV_2a for four-class classification task											
Method	Metrics	A01	A02	A03	A04	A05	A06	A07	A08	A09	mean
Hou et al. 2022 [34]	Acc	80.90	64.24	85.76	**65.63**	44.79	54.17	**84.38**	84.03	80.56	71.61
	Kappa	0.745	0.523	0.810	**0.542**	0.264	0.389	**0.792**	0.787	0.741	0.621
Siviero et al. 2023 [35]	Acc	**83.34**	56.14	**88.82**	60.34	**57.93**	**68.93**	60.61	77.36	**91.45**	71.67
	Kappa	**0.780**	0.420	**0.850**	0.470	**0.440**	**0.590**	0.480	0.700	**0.890**	0.620
Ours	Acc	80.90	**64.79**	83.86	63.54	48.61	56.80	**84.38**	**85.88**	80.56	**72.15**
	Kappa	0.745	**0.545**	0.788	0.510	0.285	0.405	**0.792**	**0.798**	0.741	**0.623**

Based on the Wilcoxon rank-sum test with Holm p-value correction (α = 0.05), the † indicates the marked method is significantly better than the compared methods.

TABLE II
PERFORMANCE COMPARISON WITH THE SOTA METHODS ON FNIRS-BCI DATASET IN TERMS OF ACC, SEN, AND SPE

Method	Validation	Acc	Sen	Spe
Panagiotis et al. 2018 [37]	CV	92.52	-	-
Lu et al 2020 [38]	5-fold CV	95.30	-	-
Wang et al. 2022 [39]	5-fold CV	92.97	93.06	93.55
Han et al. 2022 [40]	5-fold CV	95.26	96.63	95.33
Ours	5-fold CV	**97.51**(±3.7)	**97.63**	**96.13**

Note: These evaluation metrics are calculated by averaging the results of five-fold cross-validation across eight subjects.

As such, training a model with more samples from these datasets does not necessarily lead to improved accuracy. In the context of multisite fMRI datasets, the hybrid spatio-temporal graph representation extracted by STIGR exhibits robust generalization capabilities across diverse sites and equipment, surpassing deep learning techniques, such as CNNs, stacked autoencoders (SAE), and graph isomorphism networks (GINs). These methods often struggle to improve accuracy when the sample size exceeds 800 due to the data heterogeneity issue. In addition, our model consistently demonstrates robust performance across different data scales. We observe this trend when evaluating our model on datasets of varying sizes, from the small-scale COBRE dataset with 146 samples to the larger ABIDE dataset with 1035 samples. In all cases, our model outperforms SOTA methods, underscoring its effectiveness and resilience across different data scales. Besides, it is worth noting that in the ABIDE I dataset, BrainTGL [12] employed a sliding window strategy similar to ours, partitioning the fMRI signal into distinct segments for constructing graph networks. While BrainTGL failed to achieve optimal performance due to its neglect of spatio-temporal relationship modeling in fMRI, it represents a commendable endeavor in integrating dynamic FC with GNNs.

For EEG data, STIGR continues to outperform current SOTA methods on both BCICIV_2a and BCIC2015 datasets. Specifically, STIGR achieves an average accuracy of 89.28% across subjects on BCICIV_2a and 75.53% accuracy in four-fold CV on BCIC2015. The classification results of BCICIV_2a dataset for each subject and six binary classification tasks are shown in Section A of Table I. It is shown that the STIGR model provides the highest average accuracy on the six motor imagery classification tasks for seven of the nine subjects and outperforms the runner-up [31] by 2.13% in overall average accuracy. To provide a more comprehensive evaluation of our approach on the BCICIV_2a dataset, we also include results for the four-class classification task. Given the unavailability of source code for the compared methods, we aim to benchmark our proposed model against another two recent methods [34], [35] under the same experimental conditions. The results for the four-class classification task are tabulated in Section B of Table II. The performance across different subjects illustrates the variability inherent in subject-specific EEG patterns. Despite these variations, our method demonstrates competitive performance, achieving the highest average accuracy and kappa value across subjects. In the case of the BCIC2015 dataset, which is used for competition on Kaggle, we compare our STIGR model with the winning algorithm. The winning algorithm, which does not utilize leaky features, achieves an accuracy of 72.94 (±3.7)%. In contrast, STIGR achieves better accuracy of 75.53(±4.2)% in four-fold CV, marking a 2.59% improvement. To further validate the generalization ability of our STIGR model, we also conduct intersubject evaluation on BCICIV_2a dataset and compared

TABLE III
PERFORMANCE COMPARISON WITH OTHER REPRESENTATIVE STGNN ARCHITECTURES IN NEUROIMAGING FIELD

Section A: Performance comparison with STGNNs architectures on fMRI datasets												
Method	ABIDE I				ABDIE II				ADHD-200			
	ACC(STD)	SEN	SPE	AUC	ACC(STD)	SEN	SPE	AUC	ACC(STD)	SEN	SPE	AUC
STGCN	68.2(±4.7)	71.5	64.1	72.8	68.4(±4.0)	70.3	66.1	74.1	71.4(±4.4)	77.5	65.1	78.6
STAGIN	69.5(±4.1)	74.6	64.1	75.6	68.8(±4.0)	71.3	64.9	75.1	73.2(±3.2)	75.0	66.0	77.2
BrainGNN	68.7(±3.9)	69.8	65.8	73.4	67.6(±4.3)	70.7	64.1	73.7	72.9(±3.9)	78.9	61.3	77.8
BrainGSLs	69.7(±3.7)	70.7	64.8	74.7	68.2(±4.7)	70.9	65.1	74.5	72.1(±3.5)	77.6	64.2	76.8
STIGR(ours)	**72.7**†(±3.4)	**76.1**	**68.8**†	**79.0**†	**72.5**†(±3.6)	**73.0**	**71.6**†	**77.6**†	**76.2**†(±3.7)	**79.5**	**72.4**†	**81.8**†
Section B: Performance comparison with STGNNs architectures on EEG & fNIRS-BCI datasets												
Method	BCICIV_2a						BCIC2015			fNIRS-BCI		
	L/R	L/F	L/T	R/F	R/T	T/F	ACC(STD)	SEN	SPE	ACC	SEN	SPE
STGCN	78.6	80.9	82.4	79.4	82.4	79.5	70.9(±4.8)	54.9	79.8	94.3(±4.1)	94.0	95.1
STAGIN	79.4	81.7	83.6	80.6	83.5	79.6	72.1(±4.1)	51.9	80.9	94.7(±4.0)	93.9	95.8
BrainGNN	81.8	83.6	86.2	82.4	83.8	80.9	70.1(±5.2)	55.2	77.5	95.0(±3.5)	94.7	97.0
BrainGSLs	80.2	80.6	84.2	81.2	82.7	79.6	71.4(±4.9)	54.3	81.2	95.2(±3.4)	94.2	96.3
STIGR(ours)	**86.4**†	**91.4**†	**90.6**	**90.2**†	**90.0**†	**86.0**†	**75.5**†(±4.2)	**57.8**	**86.8**†	**97.5**†(±3.7)	**96.1**	**97.6**

Based on the Wilcoxon rank-sum test with Holm p-value correction (α = 0.05), the † indicates the marked method is significantly better than the compared methods.

TABLE IV
ABLATION STUDY OF OUR STIGR MODEL

Model	Component			ABIDE I		ADHD-200		BCICIV_2a	
	L-DGCN	G-STAN	CL-AM	Accuracy	Δ	Accuracy	Δ	Accuracy	Δ
(a)	–	–	–	68.2(±4.7)	–	71.4(±4.0)	–	80.5	–
(b)	–	✓	–	69.4(±3.6)	+1.2	72.4(±4.2)	+1.0	84.3	+3.8
(c)	✓	–	–	69.9(±3.9)	+1.7	73.2(±3.8)	+1.8	85.2	+4.7
(d)	✓	✓	–	71.5(±3.5)	+3.3	74.9(±3.3)	+3.5	87.2	+6.7
(e)	✓	✓	✓	**72.7**(±3.4)	**+4.5**	**76.2**(±3.6)	**+4.8**	**89.3**	**+8.8**

Δ refers to the relative performance improvement over the baseline model. ✓represents the given function participates in the training process.

its performance with other SOTA methods, as detailed in supplementary II.

Regarding the fNIRS data, Table II shows that previous studies have achieved an average accuracy ranging from 92.52% to 95.30% across eight subjects on the fNIRS-BCI dataset. The proposed STIGR model achieves the highest average accuracy, sensitivity, and specificity of 97.51%, 97.63%, and 96.13%, respectively. In addition, in comparison to the study [39] that also incorporates Transformer in its classification framework, STIGR demonstrates a significant accuracy improvement of 4.54%. Generally speaking, the proposed model demonstrates competitive classification performance while maintaining strong robustness and generalization capabilities across various types of neuroimaging datasets.

In addition to comparing with the SOTA methods, we also evaluate the effectiveness of STIGR against other representative STGNN architectures. To our knowledge, no STGNN-based methods have been proposed specifically for these neuroimaging datasets. Therefore, we compare STIGR with several representative STGNN architectures de for neuroimaging data, including STGCN [18], STAGIN [17], BrainGNN [41], and BrainGSLs [42]. The comparison results, as shown in Table III, indicate that STIGR consistently outperforms all competitors, dominating 100% (12/12) of evaluation metrics. Notably, nine of these metrics demonstrate statistical significance by the Wilcoxon rank-sum test at $p \leq 0.05$ on fMRI datasets. On the EEG and fNIRS datasets, STIGR also exhibits strong generalization, again dominating 100% (12/12) of the evaluation metrics and achieving statistical significance on eight of them. Since STGCN, STAGIN, and BrainGSLs were originally proposed for fMRI data, their performance on the EEG datasets, especially BICCIV_2a, significantly lags behind that of our proposed method. For the fNIRS-BCI dataset, while all compared STGNNs achieve similar accuracy of around 95%, our STIGR outperforms them by 2.3%. Overall, we conclude that STIGR achieves the highest average accuracy across these datasets, maintaining an appropriate balance between specificity and sensitivity compared to other representative STGNN architectures. To further establish the reliability of our proposed model and the fairness of our comparative study, we also conduct a leave-one-site-out test and a model complexity analysis, detailed in supplementary Sections III and IV, respectively.

C. Ablation Study

To evaluate the effectiveness of each proposed module (L-DGCN, G-STAN, and CL-AM) on STIGR's performance, ablation studies are conducted on the ABIDE I, ADHD-200, and BCICIV_2a dataset. We select STGCN [18] as our baseline [Model (a)], primarily because it is the first spatio-temporal graph convolutional network introduced in neuroimaging research, and its straightforward structure allows for easy customization. The STGCN with G-STAN is referred to as Model (b). Subsequently, we replace STGCN in baseline with L-DGCN as Model (c), and then extended the L-DGCN architecture to incorporate G-STAN, creating Model (d). The complete STIGR model, which includes all components, is referred to as Model (e). The experimental results are tabulated in Table IV.

To our surprise, the inclusion of only G-STAN [Model (b)] and the L-DGCN component [Model (c)] in our model resulted in significant improvements over the baseline. Specifically, the model with G-STAN [Model (b)] enhances the baseline performance by 1.2%, 1.0% and 3.8% in mean accuracy on ABDE I, ADHD-200, and BCICIV_2a datasets, respectively. This demonstrates the critical role of G-STAN and L-DGCN in capturing complex spatio-temporal relations for effective feature extraction in dynamic neuroimaging. That is to say, L-DGCN can serve as a powerful backbone model for extracting dynamic FC correlations within adjacent scanning windows. In addition, by introducing the G-STAN module on top of Model (c), we observe additional accuracy improvements of 1.6%, 1.7%, and 2.0% for the ABIDE I, ADHD-200, and BCICIV_2a datasets in Model (d). The performance improvement can be attributed to our G-STAN module, which supplements the missing global spatio-temporal representation of L-DGCN through its globally self-attention mechanism. Thus, our model is designed to obtain dynamic graph representations through L-DGCN for short-term (adjacent graphs)

TABLE V
EFFECTIVENESS ANALYSIS OF DIFFERENT ATLAS, LEARNABLE COEFFICIENTS, AND NEIGHBOR STEP

Section A: Learned and tunable coefficient values for models trained on different datasets					
Coefficient	ADHD-200	ABIDE I	COBRE	BCICIV_2a	fNIRS-BCI
θ (in gated fusion)	0.43	0.57	0.53	0.62	0.51
α (in final loss function)	0.72	0.68	0.57	0.74	0.62

Section B: Effectiveness analysis of threshold τ in adjacent matrix construction.					
Model	Threshold (τ)	ABIDE I		ADHD-200	
		Acc	Auc	Acc	Auc
STIGR	0	72.1(±4.0)	78.2	75.3(±3.7)	81.0
	0.1	**72.7**(±3.4)	**79.0**	**76.2**(±3.6)	**81.8**
	0.2	71.8(±3.2)	78.8	75.5(±3.9)	80.9
	0.3	71.9(±3.9)	77.5	74.8(±3.6)	81.2
	0.5	71.2(±3.4)	76.5	74.5(±3.8)	81.1

Section C: Effectiveness analysis of different neighbor step on ABIDE I and ADHD-200 dataset.					
Model	Neighbor step (k)	ABIDE I		ADHD-200	
		Acc	Auc	Acc	Auc
STIGR	1	**72.7**(±3.4)	**79.0**	**76.2**(±3.6)	**81.8**
	2	71.9(±3.5)	78.0	75.1(±4.0)	80.7
	3	71.2(±3.2)	77.5	74.5(±3.6)	80.5

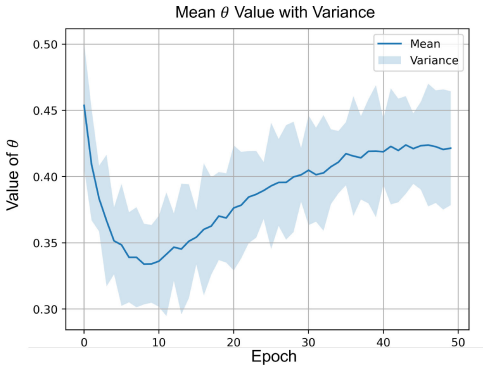

Fig. 8. Variation of learnable coefficient θ during the training process on ADHD-200.

analysis, while G-STAN extracts long-term spatio-temporal dependencies (global attention). Since the CL-AM module is proposed based on L-DGCN and G-STAN, we cannot evaluate its effectiveness by eliminating the other two components. However, we can assess the impact of the generated A_{T-CL} by directly comparing the performance of Model (d) and Model (e). The introduction of the CL-AM module results in significant accuracy improvements of 1.2%, 1.3%, and 2.1% for the ABIDE I, ADHD-200, and BCICIV_2a datasets, respectively. This finding underscores the importance of adjacent matrix construction in graph convolution and highlights the potential of contrastive learning in modeling representation similarity. Overall, Model (e), which integrates all three components (L-DGCN, G-STAN, and CL-AM), consistently outperforms the baseline model, achieving accuracy improvements of 4.5%, 4.8%, and 8.8% on the ABIDE I, ADHD-200, and BCICIV_2a datasets, respectively. Consequently, we conclude that all components clearly contribute to the performance enhancement of STIGR.

D. Parameter Sensitivity Analysis

To evaluate the effectiveness of the proposed components and novel loss function, we have tabulated the values of the learned coefficients θ (in gated fusion) and α (in the final loss function) in Section A of Table V. As anticipated, it is evident that neither θ nor α converge to the extremes of 0.0 or 1.0 across different datasets. Instead, it tends to remain around the mid-value of 0.5, which suggests that both the L-DGCN and G-STAN components jointly play an equitable role in augmenting the model performance across various datasets. In addition to the analysis of θ across datasets, we conduct specific experiments on the ADHD-200 dataset to further examine the behavior of θ within a unified dataset. The results are clearly presented in the accompanying Fig. 8. From the findings, it can be observed that the mean value of θ initially starts around 0.5 and gradually decreases to 0.33, eventually converging at 0.42. This trajectory of θ during training offers valuable insight into the parameter's adaptation within a unified dataset. Besides, it is also worth mentioning that the coefficient of the loss function, α, consistently remains above 0.5, suggesting that the binary cross-entropy classification loss plays a more crucial role than the contrastive loss. We perform visualizations by varying the parameter α from 0 to 1 [Fig. 9(a)], showcasing the performance differences observed with the different L_{CL} loss weights. The accuracy remains relatively stable as the weight of L_{CL} is increased, with a sharp trend toward decreasing performance as alpha approaches from 0.2 to 0 (i.e., full L_{CL} contribution). The model performance across different datasets shows a peak in accuracy when the weight of L_{CL} is around 0.3. This suggests that a certain amount of L_{CL} is beneficial for enhancing accuracy, but too much weight on L_{CL} could lead to performance degradation. These insights are critical for fine-tuning the balance between the contrastive loss and other components of the loss function.

We also perform a parameter sensitivity analysis on two specific aspects: 1) the number of layers L and 2) the length of scanning windows S, specifically on the ABIDE I and ADHD-200. To evaluate the impact of the number of layers L, we explore settings to vary L from 1 to 4. Regarding the length of scanning windows S, we test the model under three different configurations: S = 15, $S = 30$ TR, and a hybrid setting, $S =$ 15/30 TR. In the case of $S = 15/30$ TR, where we use 15 TR for the first layer and 30 TR for the second. The experimental results are presented in Fig. 9. With respect to the number of layers, we observe an enhancement in performance with an increase in L up to 2 layers, after which the performance noted a marginal decline. Regarding the length of scanning windows, the variations in performance across the different session lengths were minimal, suggesting that the model performance is not heavily dependent on this hyperparameter. Interestingly, the hybrid setting of $S = 15/30$ TR exhibited a performance improvement over the singular settings. This can be attributed to its alignment with the hierarchical nature of GNNs' message-passing algorithms [43], which progressively aggregate information from closer to farther neighbors.

In addition, to examine the impact of the adjacency matrix construction in graph modeling, we conduct a series of experiments within our L-DGCN module. By conducting experiments with τ values ranging from 0 to 0.5, we have gathered the experimental results that clarify the relationship between threshold settings and model performance, as shown in Section B of Table V. It is worth noting that when $\tau = 0$, it indicates that all FC values were utilized for matrix construction. Upon analyzing the result, we identify that increasing τ initially improved the performance of our model and eventually reached a plateau with a slight decline.

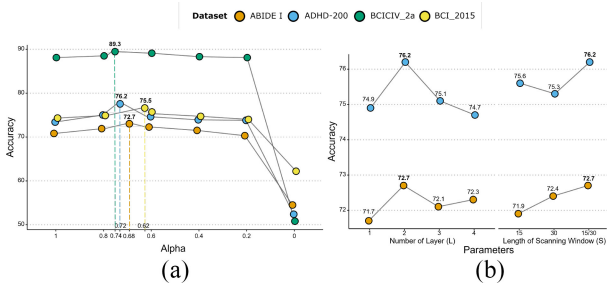

Fig. 9. Performance comparison with different hyperparameters. (a) Performance comparison with different Alpha. (b) Performance comparison with different L and S.

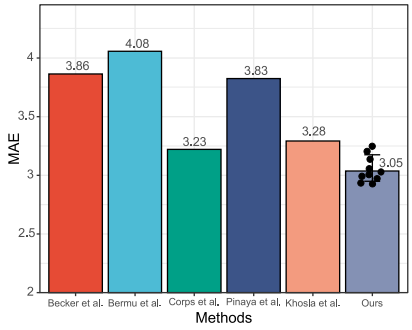

Fig. 10. Performance comparison of brain age prediction on the ABIDE I.

However, peak performance for both the ABIDE I and ADHD-200 datasets was observed at $\tau = 0.1$. These findings validate our proposition that a moderate τ effectively balances the elimination of redundant information with the retention of pertinent data, thereby enhancing the efficiency of our model. Generally, variations in performance across different τ levels are minimal, demonstrating the model's capability to effectively handle both scenarios of information redundancy at $\tau = 0$ and information scarcity at $\tau = 0.5$.

In this section, we delve into how varying neighbor steps affect our framework's performance. By adjusting the neighbor steps, denoted as k, when constructing the dynamic adaptive-neighbor graph (V_{DAN}), we explore different construction scenarios. V_{DAN} includes nodes within a range of k steps from the central node d, represented as $V_{\text{DAN}} = [V[d-k], V[d], V[d+k]]$. We assess k from 1 to 3 in simulation experiments, tabulated in Section C of Table V. Results show that $k = 1$ yields the best performance on ABIDE I and ADHD-200 datasets. Increasing k leads to a slight decrease in Acc and Auc on both datasets, indicating that prioritizing immediate neighbors effectively extracts relevant information. This is because finer time granularity at $k = 1$ allows for a precise representation of temporal dynamics and stronger correlations. However, as k increases, granularity decreases, limiting the capacity to capture fine-grained temporal relationships. Therefore, a smaller k is preferable for achieving higher performance on neuroimaging datasets. We also assess the efficacy of employing various atlases. Detailed results are provided in supplementary Section IV.

E. Brain Age Prediction

Accurately predicting brain age is crucial in neuroimaging analysis, as it adds in the early detection of vital indicators of an individual's brain health [44]. Many recent neuroimaging studies utilize deep learning perform age prediction and classification tasks concurrently to improve model generalization across various applications [45]. However, these studies typically require a complete separation of the two tasks during model training, necessitating training each task independently with specific modifications to the model architecture. Leveraging the powerful representation capabilities of STIGR, we aim to overcome this limitation by directly utilizing graph-level representations learned from previous classification tasks. STIGR is trained on classification tasks as a feature extractor to obtain high-quality graph representations. These representations are then fed into a three-layer fully connected network for training and subsequently used for age prediction. This approach enables training a simple regression head network instead of the entire model, as done in previous studies. As depicted in Fig. 10, our method achieves competitive performance compared to recent research on brain age prediction in the ABIDE I dataset [44], [45], [46], with a mean absolute error (MAE) of 3.05($\pm$0.28) years. It is worth noting that this comparison is not conducted under identical conditions, as these methods utilize different data volumes and validation approaches. Our method outperforms the runner-up with a decrease in MAE of 0.18 years, showcasing the effectiveness of leveraging learned graph representations to reduce training time for regression tasks while achieving superior results. We also conduct brain age prediction on ABIDE II and ADHD-200 with MAEs of 3.69($\pm$0.39) and 2.62($\pm$0.11), respectively. In summary, STIGR is a versatile framework that can effectively learn graph representations for different tasks.

F. Interpretation Analysis

In addition to the global property of graph subjects explored in previous experiments, the rich information encoded in the node and edge features of the learned graph representation remains crucial for a comprehensive neuroimaging analysis. As shown in Fig. 2, an ideal graph representation learning framework should support a variety of graph tasks, spanning node-level, edge-level, and global-level analyses. Therefore, we delve into STIGR's capacity to learn informative node-level and edge-level representations for model interpretation. This approach allows us to directly probe the model's focus on specific brain regions and their connections, offering a biologically interpretable and scalable window into its decision-making process. The fused graph representation R (feature map), generated by the gated fusion layer through forward propagation, serves as a representation that preserves essential input patterns [47]. The magnitude of the activations in R corresponding to each ROI/electrode can be directly interpreted as a measure of its importance. Higher magnitudes suggest greater relevance to the downstream task. This is because graph convolutions and attention mechanisms amplify the signal from important nodes [48], leading to higher activations

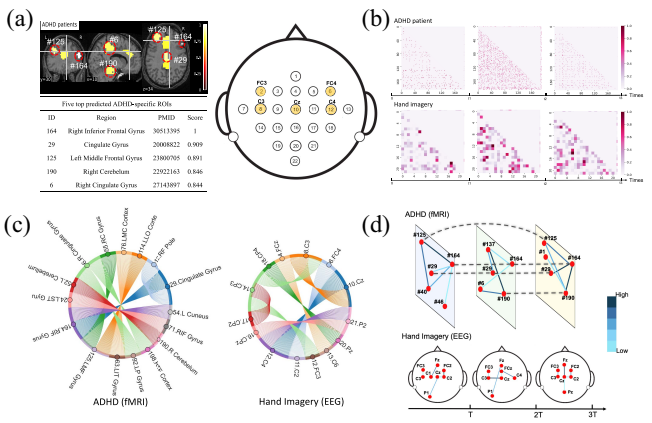

Fig. 11. (a) and (b) visualize the discriminative ROIs (fMRI)/channels (EEG) obtained from static and dynamic node-level representations, respectively. (c) and (d) illustrate the visualization of discriminative connections obtained from static and dynamic edge-level representations.

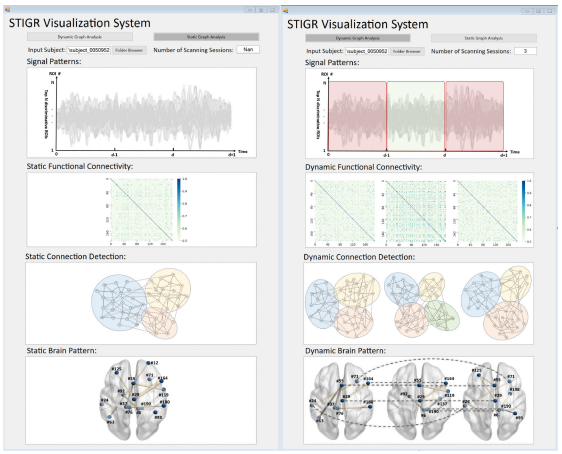

Fig. 12. Example demonstration of the STIGR visualization system, showcasing its potential in aiding the analysis of individual-level neurological patterns for ADHD.

in R. Mathematically, this process can be formulated as

$$\text{Score}_{j,c} = \frac{1}{N_c}\sum_{i=1}^{N_c} R_{i,j}^{c} \tag{12}$$

where N_c is the number of subjects in class c and $R_{i,j}^{c}$ represents the activation of the jth ROI/electrode for the ith subject in class c, obtained as the output of our well-trained model without the tower network. To illustrate the practical potential of the learned representations for model interpretation, we conduct case studies on two datasets: 1) ADHD-200 for fMRI and 2) BCICIV_2a for EEG. For detailed interpretability analysis of the other datasets, please refer to the supplementary Section VI.

In static node-level analysis, attention scores for each ROI are computed by aggregating their saliency values from feature maps. Using the weighted sum, we determine the relative importance of each ROI and visualize the top-5 ROIs/channels with the highest attention scores [Fig. 11(a)]. The left panel displays ADHD-related ROIs inferred by the learned static node-level representation. Manual investigation validated the five predicted ADHD-specific ROIs: *Right Inferior Frontal Cortex* (164), *Cingulate Gyrus* (29), *Left Middle Frontal Gyrus* (125), *Right Cerebelum* (190), and *Right Cingulate Gyrus* (6), associated with neurological manifestations of ADHD. For example, Rubia et al. [49] reported dysfunction in *right inferior frontal cortex* (164) of ADHD subjects, a key center of cognitive control. Similarly, the right panel shows the top five channels essential for detecting changes in EEG signal during the motor imagery task (hand versus feet): *C3*, *C4*, *FC3*, *FC4*, and *Cz*, aligning with prior studies [50]. In the dynamic node-level analysis, we partition the feature map into three time segments. The mean FC matrices are calculated for each segment to measure the correlations between nodes in the neuroimaging data. These matrices are then visualized to track dynamic changes in node-level significance over time, as illustrated in Fig. 11(b). In particular, this visualization demonstrates that the significant ROIs and channels identified in the static analysis retain their significance in the dynamic analysis. This consistency reinforces the reliability of our findings, validating the credibility of results in both static and dynamic node-level analyses.

In addition, in the static edge-level analysis, we derive the mean FC using the obtained feature maps, quantifying the overall connectivity between different ROIs. The top-10 FC connections with the highest values are visualized, shown in Fig. 11(c). We find that the strongest connectivity was observed between the right inferior frontal cortex region (164) and the cingulate gyrus (29) in participants with ADHD, which is consistent with a recent study [49] that reported increased positive functional connectivity between these two regions in adolescents with ADHD. For the EEG task of hand imaging, the connections between *C3* and *Cz*, as well as between *C4* and *Cz*, were found to be the most significant, in agreement with the results reported in [51]. Interestingly, all five of the top ADHD-specific ROIs and hand-specific channels in the static node-level representation were also identified in the top 10 discriminative FC-connected regions in the static edge-level representation, indicating strong consistency.

Similarly to dynamic node-level analysis, we also partition the obtained feature map into three distinct time segments. The corresponding FC matrices are calculated and specific top-5 FC connections are identified that exhibited significance in the ADHD/hand imagery group but not in the control (TC/foot imagery). In Fig. 11(d), we present the dynamic edge connection patterns and demonstrate the efficiency of exploring the evolution of these dynamic interaction brain networks over time to identify reproducible dynamic edge connection patterns as potential neuroimaging biomarkers. Specifically, our statistical analysis revealed that only for this particular pattern [the upper panel of Fig. 11(d)], 35.6% (129/362) of ADHD subjects in the ADHD-200 dataset exhibited a similar dynamic FC pattern. Furthermore, in the BCICIV_2a dataset, 40.3% (58/144) of the samples showed a similar dynamic connection pattern, as illustrated in the lower panel of Fig. 11(d). These findings suggest that the dynamic activity patterns identified by STIGR have great potential to advance

understanding of brain activity mechanisms. To validate the effectiveness of our graph representation-based interpretation analysis, we also compare the results with the traditional layer-wise relevance propagation method in the supplementary material.

Finally, we have developed a visualization system for subject-level diagnosis, integrating the analysis approaches detailed above. Illustrated in Fig. 12, our trained STIGR effectively captures significant nodes and edges from static and dynamic viewpoints, allowing exploration of ADHD-associated neurological patterns when selecting an input subject (an ADHD subject sample is shown). Specifically, we explore FC connectome at the node-level, community detection, and edge-level connection patterns within this system. In particular, these individual-level results align substantially with the ADHD-specific findings highlighted in Fig. 11. Our visualization system is designed to improve the interpretation of learned graph representations, offering medical profession-also an interpretable tool to explore task-specific neurological patterns at the individual level. In essence, this framework supports informed and data-driven medical decision-making.

V. Conclusion

In this article, we have proposed STIGR, an effective graph representation learning framework for capturing discriminative spatio-temporal graph representations of neuroimaging data. STIGR combines both local and global spatio-temporal dynamics through two major components: 1) L-DGCN and 2) G-STAN. L-DGCN performs graph convolution operations while simultaneously exploring spatio-temporal dependencies by considering the temporal connection of dynamic graphs; G-STAN is developed to learn the missing global spatio-temporal representation and fuse it with L-DGCN to enhance the overall dynamic graph representation ability. In addition, we have designed a CL-AM module based on contrastive learning to adaptively learn the adjacent node matrix for L-DGCN.

Extensive experiments across three neuroimaging datasets (fMRI, EEG, and fNIRS) demonstrate STIGR's superiority over current SOTA methods on benchmark datasets. Ablation studies confirm the contributions of the proposed components. The learned graph representations are versatile and applicable to various downstream tasks, including classification, regression, and interpretation at node, edge, and graph levels. Overall, our framework offers a promising solution for effective graph representation learning in multiple spatio-temporal neuroimaging contexts. In the future, we expect to combine STIGR with multimodality learning to explore the feasibility of multimodal graph representations in the context of neuroimaging.

References

[1] C. Lian, M. Liu, Y. Pan, and D. Shen, "Attention-guided hybrid network for dementia diagnosis with structural MR images," *IEEE Trans. Cybern.*, vol. 52, no. 4, pp. 1992–2003, Apr. 2022.

[2] P. Zhong, D. Wang, and C. Miao, "EEG-based emotion recognition using regularized graph neural networks," *IEEE Trans. Affect. Comput.*, vol. 13, no. 3, pp. 1290–1301, Jul.–Sep. 2020.

[3] R. Liu, Z.-A. Huang, Y. Hu, Z. Zhu, K.-C. Wong, and K. C. Tan, "Attention-like multimodality fusion with data augmentation for diagnosis of mental disorders using MRI," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 7627–7641, Jun. 2024.

[4] D. Ahmedt-Aristizabal, M. A. Armin, S. Denman, C. Fookes, and L. Petersson, "Graph-based deep learning for medical diagnosis and analysis: Past, present and future," *Sensors*, vol. 21, no. 14, p. 4758, 2021.

[5] Y. Kong et al., "Multi-stage graph fusion networks for major depressive disorder diagnosis," *IEEE Trans. Affect. Comput.*, vol. 13, no. 4, pp. 1917–1928, Oct.–Dec. 2022.

[6] S. Wein, A. Schüller, A. M. Tomé, W. M. Malloni, M. W. Greenlee, and E. W. Lang, "Forecasting brain activity based on models of spatiotemporal brain dynamics: A comparison of graph neural network architectures," *Netw. Neurosci.*, vol. 6, no. 3, pp. 665–701, 2022.

[7] X. Cao et al., "scPriorGraph: Constructing biosemantic cell–cell graphs with prior gene set selection for cell type identification from scRNA-seq data," *Genome Biol.*, vol. 25, no. 1, p. 207, 2024.

[8] C. Dai et al., "Brain EEG time-series clustering using maximum-weight clique," *IEEE Trans. Cybern.*, vol. 52, no. 1, pp. 357–371, Jan. 2020.

[9] Q. Yu, R. Wang, J. Liu, L. Hu, M. Chen, and Z. Liu, "GNN-based depression recognition using spatio-temporal information: A fNIRS study," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 10, pp. 4925–4935, Oct. 2022.

[10] J. Wang, Q. Wang, H. Zhang, J. Chen, S. Wang, and D. Shen, "Sparse multiview task-centralized ensemble learning for ASD diagnosis based on age-and sex-related functional connectivity patterns," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 3141–3154, Aug. 2019.

[11] Z.-A. Huang, R. Liu, Z. Zhu, and K. C. Tan, "Multitask learning for joint diagnosis of multiple mental disorders in resting-state fMRI," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 8161–8175, Jun. 2024.

[12] L. Liu et al., "BrainTGL: A dynamic graph representation learning model for brain network analysis," *Comput. Biol. Med.*, vol. 153, Feb. 2023, Art. no. 106521.

[13] P. Yang et al., "Fused sparse network learning for longitudinal analysis of mild cognitive impairment," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 233–246, Jan. 2021.

[14] R. Liu, Z.-A. Huang, M. Jiang, and K. C. Tan, "Multi-LSTM networks for accurate classification of attention deficit hyperactivity disorder from resting-state fMRI data," in *Proc. 2nd Int. Conf. Ind. Artif. Intell. (IAI)*, 2020, pp. 1–6.

[15] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 2, 2021, p. 4.

[16] R. Liu, Z.-A. Huang, Y. Hu, Z. Zhu, K.-C. Wong, and K. C. Tan, "Spatial–temporal co-attention learning for diagnosis of mental disorders from resting-state fMRI data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 8, pp. 10591–10605, Aug. 2024.

[17] B.-H. Kim, J. C. Ye, and J.-J. Kim, "Learning dynamic graph representation of brain connectome with spatio-temporal attention," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 4314–4327.

[18] S. Gadgil, Q. Zhao, A. Pfefferbaum, E. V. Sullivan, E. Adeli, and K. M. Pohl, "Spatio-temporal graph convolution for resting-state fMRI analysis," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2020, pp. 528–538.

[19] Y. Li, Y. Liu, Y.-Z. Guo, X.-F. Liao, B. Hu, and T. Yu, "Spatio-temporal-spectral hierarchical graph convolutional network with semisupervised active learning for patient-specific seizure prediction," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 12189–12204, Nov. 2022.

[20] Q. Zhu et al., "Spatio-temporal graph hubness propagation model for dynamic brain network classification," *IEEE Trans. Med. Imag.*, vol. 43, no. 6, pp. 2381–2394, Jun. 2024.

[21] D. Yang et al., "A novel Spatio-temporal hub identification in brain networks by learning dynamic graph embedding on Grassmannian manifolds," *IEEE Trans. Med. Imag.*, early access, Nov. 19, 2024, doi: 10.1109/TMI.2024.3502545.

[22] F. Hu, L. Zhang, X. Yang, and W.-A. Zhang, "EEG-based driver fatigue detection using spatio-temporal fusion network with brain region partitioning strategy," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 8, pp. 9618–9630, Aug. 2024.

[23] L. Chen et al., "An explainable spatio-temporal graph convolutional network for the biomarkers identification of ADHD," *Biomed. Signal Process. Control*, vol. 99, Jan. 2025, Jan. 106913.

[24] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 914–921.

[25] R. Liu, Z.-A. Huang, Y. Hu, L. Huang, K.-C. Wong, and K. C. Tan, "Spatio-temporal hybrid attentive graph network for diagnosis of mental disorders on fMRI time-series data," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 8, no. 6, pp. 4046–4058, Dec. 2024.

[26] M. Gadgil, E. Peterson, J. Tregellas, S. Hepburn, and D. C. Rojas, "Differences in global and local level information processing in autism: An fMRI investigation," *Psychiatry Res.*, vol. 213, no. 2, pp. 115–121, 2013.

[27] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–15.

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[29] A. Van Den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[30] S. U. Amin, M. Alsulaiman, G. Muhammad, M. A. Bencherif, and M. S. Hossain, "Multilevel weighted feature fusion using convolutional neural networks for EEG motor imagery classification," *IEEE Access*, vol. 7, pp. 18940–18950, 2019.

[31] J.-S. Bang, M.-H. Lee, S. Fazli, C. Guan, and S.-W. Lee, "Spatio-spectral feature representation for motor imagery classification using convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 7, pp. 3038–3049, Jul. 2022.

[32] P. Gaur, H. Gupta, A. Chowdhury, K. McCreadie, R. B. Pachori, and H. Wang, "A sliding window common spatial pattern for enhancing motor imagery classification in EEG-BCI," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.

[33] K. Das and R. B. Pachori, "Electroencephalogram based motor imagery brain computer interface using multivariate iterative filtering and spatial filtering," *IEEE Trans. Cogn. Develop. Syst.*, vol. 15, no. 3, pp. 1408–1418, Sep. 2023.

[34] Y. Hou, T. Chen, X. Lun, and F. Wang, "A novel method for classification of multi-class motor imagery tasks based on feature fusion," *Neurosci. Res.*, vol. 176, pp. 40–48, Mar. 2022.

[35] I. Siviero, G. Menegaz, and S. F. Storti, "Functional connectivity and feature fusion enhance multiclass motor-imagery brain–computer interface performance," *Sensors*, vol. 23, no. 17, p. 7520, 2023.

[36] P. Khan, P. Ranjan, and S. Kumar, "Data heterogeneity mitigation in healthcare robotic systems leveraging the Nelder–Mead method," in *Artificial Intelligence for Future Generation Robotics*. Amsterdam, The Netherlands: Elsevier, 2021, pp. 71–82.

[37] P. C. Petrantonakis and I. Kompatsiaris, "Single-trial NIRS data classification for brain–computer interfaces using graph signal processing," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 9, pp. 1700–1709, Sep. 2018.

[38] J. Lu, H. Yan, C. Chang, and N. Wang, "Comparison of machine learning and deep learning approaches for decoding brain computer interface: An fNIRS study," in *Proc. Int. Conf. Intell. Inf. Process.*, 2020, pp. 192–201.

[39] Z. Wang, J. Zhang, X. Zhang, P. Chen, and B. Wang, "Transformer model for functional near-infrared spectroscopy classification," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 6, pp. 2559–2569, Jun. 2022.

[40] J. Han, J. Lu, J. Lin, S. Zhang, and N. Yu, "A functional region decomposition method to enhance fNIRS classification of mental states," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 11, pp. 5674–5683, Nov. 2022.

[41] U. Mahmood, Z. Fu, V. Calhoun, and S. Plis, "Attend to connect: End-to-end brain functional connectivity estimation," in *Proc. Workshop Geom. Topol. Represent. Learn.*, 2021, pp. 1–8.

[42] G. Wen et al., "Graph self-supervised learning with application to brain networks analysis," *IEEE J. Biomed. Health Inform.*, vol. 27, no. 8, pp. 4154–4165, Aug. 2023.

[43] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.

[44] B. G. Becker, T. Klein, C. Wachinger, and A. D. N. Initiative, "Gaussian process uncertainty in age estimation as a measure of brain abnormality," *NeuroImage*, vol. 175, pp. 246–258, Jul. 2018.

[45] C. Bermudez et al., "Anatomical context improves deep learning on the brain age estimation task," *Magn. Reson. Imag.*, vol. 62, pp. 70–77, Oct. 2019.

[46] J. Corps and I. Rekik, "Morphological brain age prediction using multi-view brain networks derived from cortical morphology in healthy and disordered participants," *Sci. Rep.*, vol. 9, no. 1, pp. 1–10, 2019.

[47] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.

[48] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability methods for graph convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10772–10781.

[49] K. Rubia et al., "Functional connectivity changes associated with fMRI neurofeedback of right inferior frontal cortex in adolescents with ADHD," *NeuroImage*, vol. 188, pp. 43–58, Mar. 2019.

[50] X. Ma, S. Qiu, and H. He, "Multi-channel EEG recording during motor imagery of different joints from the same limb," *Sci. Data*, vol. 7, no. 1, p. 191, 2020.

[51] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, "Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b," *Front. Neurosci.*, vol. 6, p. 39, Mar. 2012.