Computer Methods and Programs in Biomedicine 219 (2022) 106772

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Collaborative learning of graph generation, clustering and classification for brain networks diagnosis

Wenju Yang[a,b], Guangqi Wen[a,b], Peng Cao[a,b,*], Jinzhu Yang[a,b], Osmar R. Zaiane[c]

[a] College of Computer Science and Engineering, Northeastern University, Shenyang, China
[b] Key Laboratory of Intelligent Computing in Medical Image, Ministry of Education, Northeastern University, Shenyang, China
[c] Alberta Machine Intelligence Institute, University of Alberta, Edmonton, Canada

ARTICLE INFO

Article history:
Received 8 October 2021
Revised 20 March 2022
Accepted 21 March 2022

Keywords:
Autism spectrum disorder
Functional brain networks
Multi-graph clustering
Generative adversarial networks
Graph convolutional networks

ABSTRACT

Purpose: Accurate diagnosis of autism spectrum disorder (ASD) plays a key role in improving the condition and quality of life for patients. In this study, we mainly focus on ASD diagnosis with functional brain networks (FBNs). The major challenge for brain networks modeling is the high dimensional connectivity in brain networks and limited number of subjects, which hinders the classification capability of graph convolutional networks (GCNs). Method: To alleviate the influence of the limited data and high dimensional connectivity, we introduce a unified three-stage graph learning framework for brain network classification, involving multi-graph clustering, graph generation and graph classification. The framework combining Graph Generation, Clustering and Classification Networks (GraphCGC-Net) enhances the critical connections by multi-graph clustering (MGC) with a supervision scheme, and generates realistic brain networks by simultaneously preserving the global consistent distribution and local topology properties. Results: To demonstrate the effectiveness of our approach, we evaluate the performance of the proposed method on the Autism Brain Imaging Data Exchange (ABIDE) dataset and conduct extensive experiments on the ASD classification problem. Our proposed method achieves an average accuracy of 70.45% and an AUC of 72.76% on ABIDE. Compared with the traditional GCN model, the proposed GraphCGC-Net obtains 9.3%, and 10.64% improvement in terms of accuracy and AUC metrics, respectively. Conclusion: The comprehensive experiments demonstrate that our GraphCGC-Net is effective for graph classification in brain disorders diagnosis. Moreover, we find that MGC can generate biologically meaningful subnetworks, which is highly consistent with the previous neuroimaging-derived biomarker evidence of ASD. More importantly, the promising results suggest that applying generative adversarial networks (GANs) in brain networks to improve the classification performance is worth further investigation.

1. Introduction

Autism is increasingly recognized as a common brain disorder with altered brain networks [1,2]. It is a heterogeneous neurodevelopmental disorder characterized by impaired social interaction and repetitive behaviors. Neuroimaging studies have explored functional connectivity (FC) of autism spectrum disorder (ASD) through resting-state functional magnetic resonance imaging (rs-fMRI). FC is defined as a statistical dependency between different brain regions and has been quantified through Pearson's correlation (PC). Compared with structural MRI, rs-fMRI [3] is able to measure hemodynamic changes induced by neuronal activity in the whole brain at a range of time points and has been used as a primary tool to investigate and explore FC. Traditional diagnosis of ASD is typically predominantly performed through the observation individual's behaviors, questionnaires and interviews of a patient. However, these diagnostic methods are time-consuming and prone to a misdiagnosis sometimes. Integrating machine learning algorithms on neuroimages can be established to differentiate the ASD subjects from normal control subjects [4].

Motivated by breakthroughs of deep learning on Euclidean data, efforts have been made to extend convolutional neural networks (CNNs) to non-regular graph data [5,6]. Kipf [7] proposed Graph convolutional network (GCN) as an effective graph embedding model that naturally combines structure information and node features in the learning process. Recently, some researchers have applied GCN on the FBNs to extract latent graph representations [8,9]. However, at the current stage, the brain network classification via GCN models faces two challenges as follow:

[*] Corresponding author at: College of Computer Science and Engineering, Northeastern University, Shenyang, China.

E-mail addresses: caopeng@cse.neu.edu.cn (P. Cao), yangjinzhu@cse.neu.edu.cn (J. Yang).

https://doi.org/10.1016/j.cmpb.2022.106772
0169-2607/© 2022 Published by Elsevier B.V.

Challenge 1: Noisy correlations in brain networks. In brain networks, considering all the correlations may lead to the inclusion of noisy and spurious connections. Noise in brain images is due to measurement errors likely to arise from technological limitations, operator performance, equipment, environment, and other factors. Currently, PC is the simplest and most widely-used method in constructing FBNs. However, it tends to result in brain networks with dense connections. The high dimensional connections could be relatively large and thus not very discriminative, which causes overfitting issues and increases computational complexity. Due to its high dimensionality and high noise level, the FBNs analysis may not be easy to interpret. Moreover, removing weaker (potentially noisy) connections depends on a hard-threshold without enough flexibility.

Challenge 2: Limited training data. Our aim is to treat a brain network as a graph and embed it into a meaningful low-dimensional representation. The embedding learning with GCN requires a large collection of training data. However, the amount of available subjects is usually limited in the clinical application, which hinders the classification performance.

Driven by the two important issues, two questions arise: how to capture the critical structures of brain networks by removing noisy connections, and how to generate high-quality graphs that preserve the local graph structures and the global data distribution. Due to the intrinsically complex structure of brain networks, few generative adversarial network (GAN) models can be directly applied due to: (1) a large number of graph edges, (2) the incapability of preserving the topological structure, (3) the diverse interpolation incompatible with the original distribution, and (4) an unstable optimization. To resolve these issues, we develop a unified three-stage graph learning framework for brain network classification, involving multi-graph clustering, graph generation and graph classification. The framework, GraphCGC-Net, coherently combines the power of Graph Generation, Clustering and Classification Networks. Specifically, we introduce a multi-graph clustering to enhance the important connections and remove the irrelevant connections in the brain network with a supervision scheme. The multi-graph clustering with supervision is able to generate more robust and biologically meaningful functional connectivity subnetworks. Through learning the indicative edges, we improve the interpretability of the model by highlighting the critical functional connections that are significantly beneficial for the prediction. Meanwhile, the clean structure not only reduces the computational complexity, but also facilitates the downstream graph generation and classification task. Moreover, we propose a graph GAN to generate realistic brain networks for improved diagnosis performance. More specifically, we adapt the structure of α-GAN by introducing an additional code discriminator network. Meanwhile, we also employ Wasserstein GAN [10] with Gradient Penalty (WGAN-GP) [11] loss functions to prevent mode collapse. Furthermore, we introduce three constraints: local topological measure constraint, a homeomorphic constraint and dual reconstruction constraint to guarantee the graph generation quality. All of them enable the generator to preserve graph structural properties and enforce the compatibility between the latent sample distances and the corresponding graph sample distances, which are important for the brain networks analysis for neurological disorders.

Our contributions are three folds:

1. We propose a supervised multi-graph clustering, which is able to remove the noisy functional connections from a group level. Moreover, it is capable of identifying the critical structure in a brain network, which actually boosts the performance of the following graph generation and classification procedures.

2. We develop a graph generation model with the proposed multiple regularizations on the graph space and latent embedding space. Our design can stabilize GAN training, alleviate the gradient vanishing and mode collapse issues, for achieving a better approximate data distribution. To the best of our knowledge, the proposed GraphGAN++ is the first generation model on the brain networks for improved classification performance.

3. The proposed framework method is evaluated on the real-world Autism Brain Imaging Data Exchange (ABIDE). Comprehensive experiments have been conducted to evaluate the proposed framework. The experimental results demonstrate that our method not only outperforms several state-of-the-art approaches in the ASD diagnosis, but also is effective in automatically identifying disease-related brain subnetworks in the human brain.

2. Related work

2.1. Brain networks construction

In our work, the FBN is constructed by calculating pairwise linear correlations between fMRI time series of different regions of interest (ROIs) using Pearson's correlation (PC). Given two times series of length L, x and y, the value of PC can be calculated as follows:

$$\rho_{xy} = \frac{\sum_{i=1}^{L}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{L}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{L}(y_i-\bar{y})^2}}, \tag{1}$$

where $\bar{x}$ and $\bar{y}$ are the means of x and y. A correlation matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is obtained by computing all the pairwise correlations, and its values range from 1 to -1, n is the number of time series (or ROIs). The values close to 1 indicate a higher positive correlation whereas the values close to -1 indicate a higher negative correlation.

2.2. Graph convolution networks

Different from the regular structure of Euclidean data, graphs have non-regular spatial structures. Given an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, GCN can automatically use edge information to aggregate node information from the neighboring nodes to generate the $(l+1)$th layer node representations through the following equation:

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \tag{2}$$

where $\mathbf{W}^{(l)}$ is the learnable weight matrix, $\sigma(\cdot)$ denotes a nonlinear operation. $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n$, where $\mathbf{I}_n$ is the identity matrix and $\tilde{\mathbf{D}}$ is the diagonal node degree matrix of $\tilde{\mathbf{A}}$, $\mathbf{H}^{(l)}$ is the feature matrix of the lth layer.

2.3. Generative adversarial network

Recently, GANs [12] have been widely applied in the field of image generation to produce very realistic images in an unsupervised setting. This gained a good deal of interest in the medical imaging research communities [13,14]. In GANs framework, two models are simultaneously trained: a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$. The generator $\mathcal{G}$ captures the data distribution and tries to produce realistic data. It transforms a random vector $\mathbf{z}$ into a sample, where $\mathbf{z}$ usually comes from an easy-to-sample distribution. The generator is trained to generate $\hat{x} = f_g(\mathbf{z})$, which is indistinguishable from the samples in the real distribution $\mathbb{P}_r$. The discriminator $\mathcal{D}$ measures how realistic the input data is. It takes samples as inputs and outputs the probability of whether x comes from the real distribution $\mathbb{P}_r$. In practice, discriminator is trained to produce a lower probability value for the generated data. The learning is done by playing a two-player game, where discriminator tries to distinguish the real sample x and the generated sample $\hat{x}$, while the generator aims to confuse the discriminator by generating $\hat{x}$ similar to x.

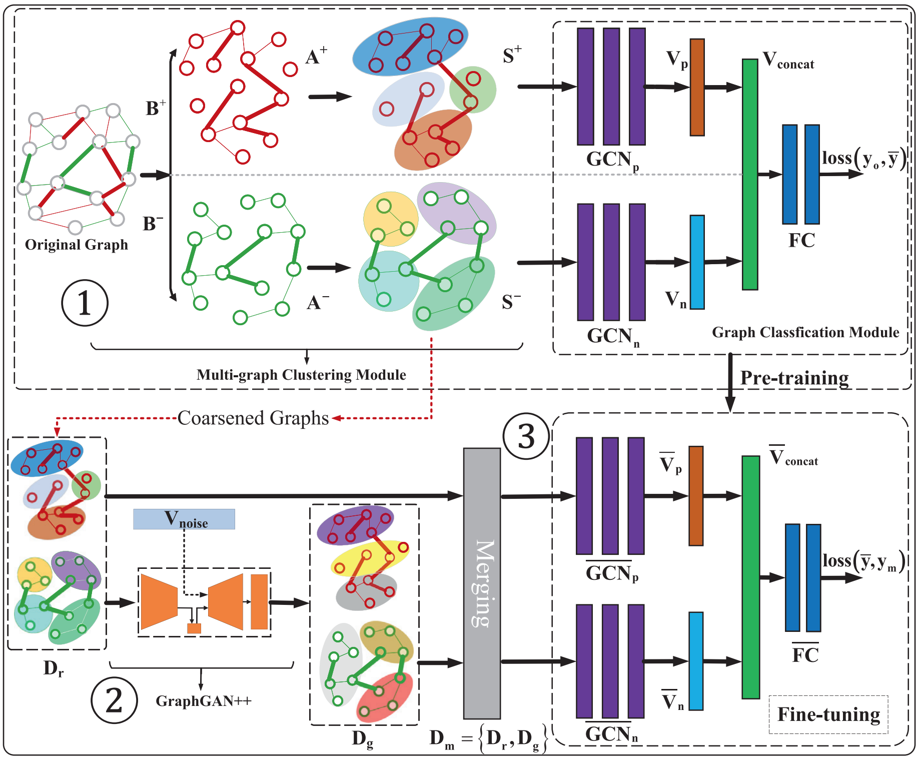


Fig. 1. The detailed pipeline of our proposed GraphCGC-Net for brain network generation and classification: ① we train a supervised multi-graph clustering to construct a coarsened graph with a critical graph structure. ② Based on the coarsened brain networks, we propose a graph GAN model to generate the real-like brain networks under the guidance of the proposed regularizations from the graph space and latent embedding space. ③ Both the generated and original graphs are merged into a mixed training dataset to fine-tune the pre-trained GCN model.

The game between the generator and the discriminator can be defined as a minimax objective function:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[\log(f_d(x))] + \mathop{\mathbb{E}}_{\mathbf{z}\sim\mathbb{P}_\mathbf{z}}[\log(1-f_d(f_g(\mathbf{z})))], \qquad (3)$$

where $\mathbb{P}_\mathbf{z}$ is the simple noise distribution, $f_g(\cdot)$ and $f_d(\cdot)$ are the functions of the generator and discriminator.

3. Architecture overview

In our work, we cast the ASD diagnosis as a classification problem and propose a data augmentation method based on the coarsened graphs. Specifically, we develop a unified three-stage learning framework for brain network classification. It involves multi-graph clustering, graph generation and graph classification. The framework is shown in Fig. 1.

4. Multi-graph clustering for the graph structure learning

In the function brain networks (FBNs), the dimensionality of functional connectivity could be relatively large and thus less discriminative. The noisy connections that are most influenced by experimental need to be removed for further analysis. Moreover, the heterogeneity present in the dataset might compromise the coherence of information between different sites, such as in the large multi-site ABIDE dataset. The existence of heterogeneity results in a lower identification of fMRI image classification, which was demonstrated in several recent imaging studies [15,16]. Therefore, we introduce a multi-graph clustering for graph structure learning that treats clusters as supernodes to remove the noisy connections and eliminate the inconsistency in brain networks data. Our objective is well-motivated: by reducing the noisy correlation edges and enhancing the important correlations, a better brain network can be achieved.

The purpose of multi-graph clustering (MGC) is to improve clustering accuracy by leveraging information from a group-level analysis. However, the MGC [17] is unsupervised so potential inconsistency between the learned graph by MGC and the subsequent classifier may degrade the final diagnosis performance. To overcome this problem, instead of an unsupervised training scheme, MGC guides the graph structure learning process by considering the group-level consistency in the subjects from the multiple sites with class label supervision. Specifically, the weights of functional connections that connect the nodes crossing different clusters are enhanced whereas the connections of the nodes within the same clusters are removed. Moreover, our MGC is an end-to-end model to jointly learn the graph clustering and graph embedding for classification in a supervision scheme, which is beneficial for improving classification performance.

At first, we regard the human brain as a functional system in which the positive and negative correlations are separately utilized for diagnosing brain diseases. However, several methods [18,19] produce a binary graph through a threshold. It may disregard the important information in the positive and negative correlations. Moreover, existing methods [20,21] have suggested that negative correlations should not be simply discarded as they contain useful information that would aid the classification task. Therefore, to avoid losing useful information regarding positive and negative correlations during the graph convolution process, we employ two branches $\mathbf{B}^+$ and $\mathbf{B}^-$ to model the positive correlation $\mathbf{A}^+$ and negative correlation $\mathbf{A}^-$, respectively. Specifically, the FBN can be described by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, which is divided into negative and positive graph data $\mathbf{A}_v^-$ and $\mathbf{A}_v^+$ depending on the FC values. The $\mathbf{A}_v^-$ and $\mathbf{A}_v^+$ are aggregated into coarsened graphs $\mathbf{S}_v^-$ and $\mathbf{S}_v^+ \in \mathbb{R}^{m\times m}$ by

$$\mathbf{S}_v^\kappa = \mathbf{F}^\mathrm{T}\mathbf{A}_v^n\mathbf{F}, \kappa \in (Neg, Pos), \qquad (4)$$

where $\mathbf{F} \in \mathbb{R}^{n \times m}$ is an indicator matrix shared among the graphs. In particular, for node i in graph $\mathbf{A}_v$, a value f_i is assigned indicating how important node i is for the classification task, and the edge between node i and j is associated with a weight w_{ij}. Each item F_{ie} of the indicator matrix $\mathbf{F}$ to be optimized can be interpreted as the membership of node i to the supernode S_e, where $F_{ie} = f_i$ if $i \in S_e$, otherwise $F_{ie} = 0$. During the aggregation of the coarsened graph $\mathbf{S}_v$, the nodes linked to the non-indicative edges are grouped into the same clusters to remove the noisy connections and eliminate graph inconsistencies. The weight of the superedge W^{κ}_{de} between supernode S_d and S_e is defined as:

$$W^{\kappa}_{de} = \sum_{i \in S_d, j \in S_e} f_i \cdot w_{ij} \cdot f_j. \tag{5}$$

Through the training of MGC, we obtain a set of clusters as superedges $\{W^{\kappa}_1, W^{\kappa}_2, \cdots, W^{\kappa}_{m^2}\}$, $\kappa \in (Neg, Pos)$ to form the coarsened graphs $\mathbf{S}_v$. The positive and negative coarsened graphs $\mathbf{S}^+_v$ and $\mathbf{S}^-_v$ are fed into the GCN models to capture the structures embedding $\mathbf{V}^+_v$ and $\mathbf{V}^-_v$. Then we flatten and concatenate the $\mathbf{V}^+_v$ and $\mathbf{V}^-_v$ together, and send them to fully-connected layers to obtain classification scores using the softmax function.

To improve the MGC performance, we incorporate three terms to regularize the learning process. First, we penalize the sum of negative values in the indicator matrix $\mathbf{F}$ to avoid the trainable matrix containing negative values. The regularization is defined as:

$$\mathbf{L}_{\text{neg}} = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\text{Relu}(-F_{ij})\right). \tag{6}$$

Second, the orthogonal constraint $\mathbf{L}_o$ is introduced to penalize the off-diagonal elements of $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ to prevent overlapping of clusters:

$$\mathbf{L}_o = \left\| \mathbf{F}^{\mathrm{T}}\mathbf{F} - \text{diag}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F}\right) \right\|_2. \tag{7}$$

In addition, we introduce a balancing loss $\mathbf{L}_b$ to balance the group size to improve the interpretability of the model. The regularization can be given by:

$$\mathbf{L}_b = \text{var}\left(\text{diag}(\mathbf{F}^{\mathrm{T}}\mathbf{F})\right), \tag{8}$$

where var$(\cdot)$ means variance. Hence, the total loss of MGC can be expressed as follows:

$$\mathbf{L}_{total} = \min_{\mathbf{F},\mathbf{W}} \left(-\frac{1}{N} \sum_{i=1}^{N} y_i \log\left(\hat{y}\right) + \sum_{j=+,-} (\lambda \mathbf{L}^{(j)}_o + \beta \mathbf{L}^{(j)}_b) + \gamma \mathbf{L}_{\text{neg}} + \eta \mathbf{L}_2\right), \tag{9}$$

where a cross entropy loss is used for classification, N is the number of samples, y is the true label and $\hat{y}$ is the predicted probability of the model. $\mathbf{L}_2 = \|\mathbf{F}\|^2_2 + \|\mathbf{W}\|^2_2$ ($\mathbf{W}$ is the weight matrix of GCN) is used to reduce overfitting. λ, β, γ and η are parameters of the constraint terms. With supervision, we can optimize the MGC model via back-propagation and learn a better solution of the cluster indicator matrix $\mathbf{F}$.

5. GraphGAN++: An improved graph GAN model for brain networks generation

GANs [22,23] have been shown to produce very similar graph structure data thanks to adversarial training. However, these approaches are still insufficient for modeling brain network generation due to the following challenges:

C.1 The instability of the GAN training process often leads to the problem of mode-collapse.

C.2 The local topology measure is not preserved during the generation process, which is crucial for brain networks.

C.3 The previous GAN models may be prone to generate inconsistent graphs with respect to graph structure and graph level embedding.

C.4 The global consistent distribution is ignored in the latent embedding space.

In this paper, we introduce a graph GAN method named GraphGAN++ to improve the graph generation quality. Figure 2 illustrates the overall architecture of the GraphGAN++ framework. It involves four modules: the graph encoder, the graph generator, the graph discriminator and the code discriminator. At first, we choose α-GAN [24] for learning latent variables to effectively address the problems of mode collapse. Moreover, we introduce a local topological measure to enable the generator to preserve global structural properties. Furthermore, we propose a latent data distance constraint to enforce the consistency between the latent embedding distances and the corresponding graph distances, which prevents the generator from producing diverse graph samples of which the corresponding latent codes are close to each other.

5.1. Graph encoder and code discriminator

Wasserstein GAN: To effectively address challenge **C.1**, we introduce Wasserstein GAN (WGAN) to prevent mode collapse and achieve a more stable training. WGAN minimizes an approximation of the Wasserstein distance between the real distributions and the generated sample distribution [10]. First, we introduce a code discriminator for learning latent variables to replace the variational inference in VGAE [25]. Second, we use Wasserstein distance to improve the generation quality. Moreover, we employ gradient penalty to stabilize the training process by penalizing the gradient norm of the interpolated samples $\hat{x}$, which can be defined as:

$$\mathbf{L}_{gp} = \mathbb{E}_{\hat{x}}[(\|\nabla_{\hat{x}} f(\hat{x})\|_2 - K)^2], \tag{10}$$

where $f(\cdot)$ is the function of the discriminator. $\mathbf{L}_{gp}$ requires the gradient of the discriminator be at most K to satisfy Lipschitz constraint.

5.1.1. Graph encoder

An encoder network **Enc** is trained to learn a function $E_\theta(\cdot)$ that maps each real sample to a point in the latent space, which consists of two GCN layers. With the coarsened graph as input, the encoder learns the graph embedding $\mathbf{Z}_e$. We introduce the graph convolution processing for each layer as follows:

$$\mathbf{Z}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}}_s \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{Z}^{(l)} \mathbf{W}^{(l)}), \tag{11}$$

where $\tilde{\mathbf{A}}_s$ is the symmetrically normalized adjacency matrix, $\mathbf{Z}^{(l)}$ is the graph embeddings of the lth layer, $\sigma(\cdot)$ denotes the ReLU activation function, and $\mathbf{W}^{(l)}$ is the trainable weight matrix. The loss function of the encoder is:

$$\mathbf{L}^c_{enc} = \min_{E_\theta(\cdot)} \left(-\frac{1}{N} \sum_{i=1}^{N} f^c_w(E_\theta(\mathbf{S}^{(i)}))\right), \tag{12}$$

where N is the number of samples, $f^c_w(\cdot)$ is a function of the code discriminator.

5.1.2. Code discriminator

In this section, we adapt the structure of α-GAN and introduce an additional code discriminator **CD** to distinguish whether the latent code comes from a real prior distribution or a code embedding obtained by the graph encoder. This purpose is to enforce the code embedding $\mathbf{Z}_e \sim E_\theta(\mathbf{S};\theta)$ to be normally distributed. Specifically, we treat the code embedding $\mathbf{Z}_e$ as fake and the Gaussian noise $\mathbf{Z}_r$ in the normal distribution as real. Then **Enc** and **CD** play an adversarial game. When the Wasserstein distance between the

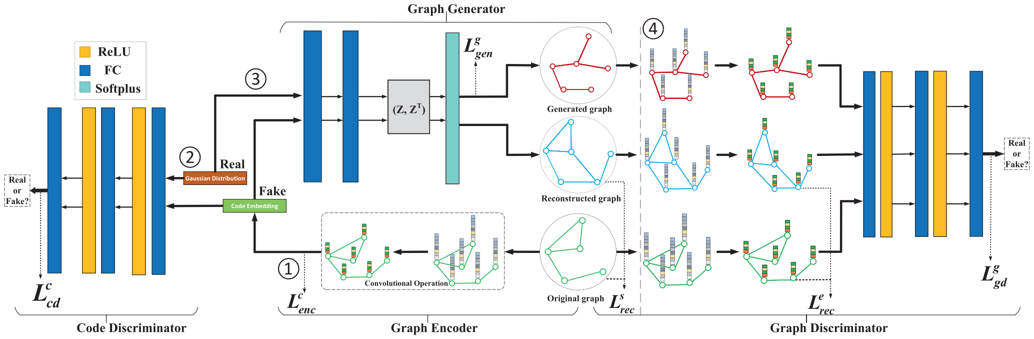


Fig. 2. The architecture of the proposed GraphGAN++. At first, the encoder network **Enc** is trained to learn a parametric mapping: $\mathbf{Z}_e \sim E(\mathbf{S};\theta)$, mapping each real sample $\mathbf{S}$ to a representation $\mathbf{Z}_e$ in the latent space. Then, the generator network **Gen** is trained to learn a function $G_\phi(\cdot)$: mapping each point in the latent space $\mathbf{Z}_e$ to a reconstructed graph $\bar{\mathbf{S}}$, or mapping a random point in the Gaussian distribution $\mathbf{Z}_r$ to a generated graph $\hat{\mathbf{S}}$. These networks are trained in tandem with the graph discriminator and the code discriminator, which learn to discriminate between the real and generated samples, and between the Gaussian noise and the latent points, respectively.

$\mathbf{Z}_e$ and $\mathbf{Z}_r$ is small enough, we consider that the posterior distribution $\mathbf{Z}_e$ and the prior distribution $\mathbf{Z}_r$ are matched. Thus the loss of the code discriminator is formulated as:

$$\mathbf{L}^c_d = \min_{f^c_w(\cdot)}\left(\frac{1}{N}\sum_{i=1}^{N} f^c_w(E_\theta(\mathbf{S}^{(i)})) - \frac{1}{N}\sum_{i=1}^{N} f^c_w(\mathbf{Z}^{(i)}_r) + \mathbf{L}^c_{gp}\right), \tag{13}$$

where $\mathbf{L}^c_{gp}$ is the gradient penalty term of the code discriminator.

5.2. Graph generator and discriminator

5.2.1. Graph generator

The generation process of the generator network involves two parts: mapping each point in the latent space $\mathbf{Z}_e$ to a reconstructed graph $\bar{\mathbf{S}}$, and mapping each point in the normal distribution $\mathbf{Z}_r$ to a generated graph $\hat{\mathbf{S}}$. More specifically, we adopt the link prediction method to generate new edges. Considering the fact that the edges in the coarsened graphs are enhanced, we choose a softplus activation function to guarantee the validity of the generated edges. The generator consists of two fully-connected layers and a link prediction module. For example, the function of link prediction in the generation procedure is expressed as:

$$\begin{aligned} p(\hat{\mathbf{S}} \mid \mathbf{Z}_r) &= \textstyle\prod_{i=1}^{m}\prod_{j=1}^{m} p(s_{ij} \mid z_i, z_j), \\ \text{with } p(s_{ij} &\mid z_i, z_j) = \sigma(\eta(z_i)^T \cdot \eta(z_j)), \end{aligned} \tag{14}$$

where s_{ij} are the edges of $\hat{\mathbf{S}}$, z_i and z_j are the elements of $\mathbf{Z}_r$, $\eta(\cdot)$ is the function of fully-connected layers, $\eta(z_i)^T \cdot \eta(z_j)$ indicates the inner product between latent variables, and σ is the activation function.

Local topological measure constraint: It is a remarkable fact that brain networks have unique topological properties, which are desirable to be preserved when generating brain networks. However, the existing graph GAN models do not consider the local topology measure in the graph space. To solve challenge **C.2**, we incorporate a topological loss function using a centrality metric that enforces the generator to preserve the local topology of the graphs. We calculate the local topology measures during both the generation and reconstruction processes as follows:

$$\mathbf{L}_{tp} = \mathbf{L}^{gen}_{tp} + \mathbf{L}^{rec}_{tp}, \tag{15}$$

where $\mathbf{L}^{gen}_{tp}$ and $\mathbf{L}^{rec}_{tp}$ are the local topology loss for the generation and reconstruction processes, respectively. At first, we set a threshold value μ to remove the weaker edges from both the true and the generated graphs at first. Then, we calculate the absolute difference of the centrality score between the generated graph and the real graph as the topology loss. In our study, we choose betweenness centrality (BC), which is commonly used in graph theory as a local measure. It is worth noting that we attempted multiple graph measures as the topological loss and found BC performs best. We discuss this further in the experiment.

$$C_{bc}(n_i) = \sum_{j,k=1}^{m} sp(j,i,k),\ (j \neq k), \tag{16}$$

Equation (16) denotes the number of shortest paths $sp(j,i,k)$ between nodes j and k that pass through node i. The topology loss in the generation procedure is defined as follows:

$$\mathbf{L}^{gen}_{tp} = \min_{G_\phi(\cdot)}\left(\sum_{i=1}^{N} \ell_{MAE}(C^{\mathbf{S}^{(i)}}_{bc}, C^{\hat{\mathbf{S}}^{(i)}}_{bc})\right), \tag{17}$$

the reconstruction process is the same as the generation for calculating the topology measure loss.

Dual reconstruction loss for consistent generation: To cope with challenge **C.3**, we also introduce a dual reconstruction loss to enforce the generator module to achieve a consistent reconstruction. More specifically, we propose a dual reconstruction scheme to guarantee the generation quality. The dual reconstruction loss of graph generator can be represented by:

$$\mathbf{L}_{rec} = \mathbf{L}^s_{rec} + \mathbf{L}^e_{rec}, \tag{18}$$

where $\mathbf{L}^s_{rec}$ and $\mathbf{L}^e_{rec}$ indicate the graph structure and graph embedding reconstruction loss, respectively.

1. More specifically, to ensure that the reconstructed graphs $\bar{\mathbf{S}}$ can preserve the same structure with the original graphs, we choose l_1 loss as the graph structure reconstruction loss:

$$\mathbf{L}^s_{rec} = \min_{G_\phi(\cdot)}\left(\frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{S}^{(i)} - \bar{\mathbf{S}}^{(i)}\right|\right). \tag{19}$$

2. Moreover, to ensure that both the reconstructed graph and the original graph can be represented by the same graph embedding by the graph encoder $E_\theta(\cdot)$, we define the reconstruction loss as follows:

$$\mathbf{L}^e_{rec} = \min_{G_\phi(\cdot)}\left(\frac{1}{N}\sum_{i=1}^{N}(E_\theta(\mathbf{S}^{(i)}) - E_\theta(\bar{\mathbf{S}}^{(i)}))^2\right). \tag{20}$$

In summary, the overall loss of the generator can be defined as:

$$\mathbf{L}^g_{gen} = \mathbf{L}_{adv} + \lambda_{tp}\mathbf{L}_{tp} + \lambda_{rec}\mathbf{L}_{rec}, \tag{21}$$

where $\mathbf{L}_{tp}$ and $\mathbf{L}_{rec}$ are the topology measure loss in Eq. (15) and the dual reconstruction loss in Eq. (18), λ_{tp} and λ_{rec} are the hyperparameters, and $\mathbf{L}_{adv}$ is the adversarial loss:

$$\mathbf{L}_{adv} = \min_{G_\phi(\cdot)}\left(-\frac{1}{N}\sum_{i=1}^{N} f^g_w(G_\phi(\mathbf{Z}^{(i)}_r)) - \frac{1}{N}\sum_{i=1}^{N} f^g_w(G_\phi(\mathbf{Z}^{(i)}_e))\right), \tag{22}$$

where $f^g_w(\cdot)$ is a function of the graph discriminator.

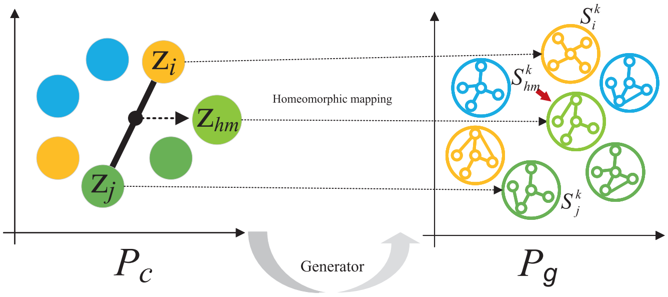


Fig. 3. Schematic diagram of the proposed homeomorphic mapping.

5.2.2. Graph discriminator

Typically, the graph discriminator distinguishes the generated graph data from the true graph data. In this paper, we expect the generated graphs $\hat{\mathbf{S}}$ and reconstructed graphs $\bar{\mathbf{S}}$ to follow the same structures as the original graphs $\mathbf{S}$. In an ideal case, $\mathbf{L}_{rec}$ should be zero. However, this is not the case for the discriminator model that uses fully connected networks. It will mislead the generator to waste its capacity in reducing the total loss, instead of maintaining the same graph structures.

To deal with this deficiency, we devise a GCN-based graph discriminator $\mathbf{D}$ that learns to enforce the intrinsic structural similarity between $\bar{\mathbf{S}}$, $\hat{\mathbf{S}}$ and $\mathbf{S}$. Specifically, the $\mathbf{D}$ includes a two-layer GCN and a three-layer full connected network. The loss function of the graph discriminator is defined as:

$$L_d^g = \min_{f_w^g(\cdot)} \left(\frac{1}{N} \sum_{i=1}^{N} f_w^g\big(G_\phi(\mathbf{Z}_e^{(i)})\right) + \frac{1}{N} \sum_{i=1}^{N} f_w^g\big(G_\phi\big(\mathbf{Z}_r^{(i)}\big)\big)$$
$$- \frac{2}{N} \sum_{i=1}^{N} f_w^g\big(\mathbf{S}^{(i)}\big) + \lambda L_{gp}^g), \tag{23}$$

where $\mathbf{L}_{gp}^g$ is the gradient penalty term of the graph discriminator.

Homeomorphic mapping for consistent constraint: Moreover, we propose a homeomorphic mapping for consistent constraint, which enforces the compatibility between latent sample distances in the embedding code space and the corresponding graph sample distances in the graph space. The consistent constraint also prevents the generator from producing diverse graph samples of which the corresponding latent codes are close to each other, i.e. mode collapse. To improve the mapping $\mathbf{Z} \to \mathbf{S}$, the homeomorphic mapping ensures that interpolation between samples in the latent code space leads to a better semantic interpolation in the graph space with a consistency constraint.

Specifically, we incorporate a homeomorphic mapping to the model, as shown in Fig. 3. We define $\mathbf{z}_{hm}$ sampled uniformly along straight lines between a pair of points vector $\mathbf{z}_i$ and $\mathbf{z}_j$ in the graph embedding space $\mathbb{P}_c$. It can be defined as follows:

$$\mathbf{z}_{hm} = \epsilon \mathbf{z}_i + (1 - \epsilon)\mathbf{z}_j, \tag{24}$$

where $\epsilon \sim \mathbf{U}[0, 1]$. The scheme allows our GAN model to generate the neighborhood samples by restricting the generated $G_\phi(\mathbf{z}_{hm})$, $G_\phi(\mathbf{z}_i)$ and $G_\phi(\mathbf{z}_j)$ in the graph space $\mathbb{P}_g$. The homeomorphic mapping loss that we add to the **Gen** and **D** is defined as:

$$L_{gen}^{hm} = \min_{G_\phi(\cdot)} (-\frac{1}{N} \sum_{i=1}^{N} f_w^g(G_\phi(\mathbf{Z}_{hm}^{(i)}))), \tag{25}$$

$$L_d^{hm} = \min_{f_w^g(\cdot)} (\frac{1}{N} \sum_{i=1}^{N} f_w^g(G_\phi(\mathbf{Z}_{hm}^{(i)})) - \frac{1}{N} \sum_{i=1}^{N} f_w^g(\mathbf{S}^{(i)}) + L_{gp}^{hm}). \tag{26}$$

5.3. Learning

In summary, our algorithm alternately updates the parameters of graph encoder, graph generator, graph discriminator and code discriminator. The total loss of our GraphGAN++ is defined by:

$$\mathbf{L}_{\text{GAN}} = \mathbf{L}_{enc}^c + \lambda_1 \mathbf{L}_d^c + \lambda_2 \mathbf{L}_{gen}^g + \lambda_3 \mathbf{L}_d^g. \tag{27}$$

We jointly train the proposed model using the following parameter updating rules for each training batch:

$$\theta_{\text{Enc}} \overset{+}{\leftarrow} -\nabla_{\theta_{\text{Enc}}} \mathbf{L}_{enc}^c,\ \theta_{\text{CD}} \overset{+}{\leftarrow} -\nabla_{\theta_{\text{CD}}} \mathbf{L}_d^c,\ \theta_{\text{Gen}} \overset{+}{\leftarrow} -\nabla_{\theta_{\text{Gen}}} \mathbf{L}_{gen}^g,\ \theta_{\text{D}} \overset{+}{\leftarrow} -\nabla_{\theta_{\text{D}}} \mathbf{L}_d^g,$$

where λ_1, λ_2, λ_3 are tunable hyper-parameters.

5.4. Final classification

The final training dataset $\mathbb{D}_m$ contains two sources: the original graph datasets $\mathbb{D}_r$ and the generated graph dataset $\mathbb{D}_g$.

$$\mathbb{D}_m = \left\{ \{\mathbf{S}_r^{(i)}\}_{i=1}^{N_r}; \{\mathbf{S}_g^{(j)}\}_{j=1}^{N_g} \right\}, \tag{28}$$

where N_r and N_g indicate the number of real and generated samples, both the $\mathbf{S}_r^{(i)}$ and the $\mathbf{S}_g^{(j)}$ are coarsened graphs. The generated graphs $\mathbb{D}_g$ are used to augment the original graph datasets $\mathbb{D}_r$. The mixed datasets are used to fine-tune the GCN model to improve the classification performance.

6. Experiment

We evaluate our GraphCGC-Net on the task of graph classification to answer the following three questions:

Q1 Is multi-graph clustering beneficial for the following GAN and GCN?

Q2 How does our GAN behavior compare with the previous GAN models?

Q3 Can the generated graph data enrich the limited dataset and improve classification performance?

6.1. Datasets and pre-processing

6.1.1. Datasets

The Preprocessed Connectomes Project (PCP) released preprocessed versions of ABIDE-I using several pipelines, and we use the data processed through Configurable Pipeline for Analysis of Connectomes (CPAC). Moreover, we choose the functional preprocessed data based on Craddock 200 (CC200) functional parcellation [26], which was segmented into 200 regions of interest (ROIs). ABIDE I includes 505 ASD subjects and 535 normal controls (NC) from 17 international sites [2]. We selected the same 871 subjects as the previous works [27], including 403 ASD subjects and 468 NCs. The details of ABIDE I are shown in Table 1.

6.1.2. Combating batch effects when combining batches (ComBat)

The parameter settings of scanners are different for each site. Therefore, we choose ComBat [16] to eliminate the negative impact of multi-site harmonization on the performance of classification models. ComBat suggests that the differences introduced in the imaging features can be normalized by adjusting the location and scale (L/S) of each site. We choose y_{ijv} to represent the connectivity values of the vth functional connectivity metric of jth scanner at ith site as:

$$y_{ijv} = \alpha_v + \mathbf{I}_{ij}\boldsymbol{\beta}_{\bar{v}} + \gamma_{iv} + \delta_{iv}\epsilon_{ijv}, \tag{29}$$

where α_v is the overall feature value for the connectivity value v between two brain regions, $\mathbf{I}_{ij}$ is the specific covariance matrix of biological information, such as age and gender, $\boldsymbol{\beta}_v$ is the regression coefficients corresponding to $\mathbf{I}_{ij}$, the terms γ_{iv} and δ_{iv} indicate

Table 1
Detail of the ABIDE I. SD is the standard deviation, M and F denote the male and female.

SITE	ASD		NC		Scanner
	Age(SD)	Gender	Age(SD)	Gender	
CALTECH	27.4(10.3)	M 15, F 4	28.0(10.9)	M 14, F 4	Siemens Trio
CMU	26.4(5.8)	M 11, F 3	26.8(5.7)	M 10, F 3	Siemens Verio
KKI	10.0(1.4)	M 16, F 4	10.0(1.2)	M 20, F 8	Philips Achieva
LEUVEN	17.8(5.0)	M 26, F 3	18.2(5.1)	M 29, F 5	Philips Intera
MAX MUN	26.1(14.9)	M 21, F 3	24.6(8.8)	M 27, F 1	Siemens Verio
NYU	14.7(17.1)	M 65, F 10	15.7(6.2)	M 74, F 26	Siemens Allegra
OHSU	11.4(2.2)	M 12, F 0	10.18(1.1)	M 14, F 0	Siemens Trio
OLIN	16.5(3.4)	M 16, F 3	16.7(3.6)	M 13, F 2	Siemens Allegra
PITT	19.0(7.3)	M 25, F 4	18.9(6.6)	M 23, F 4	Philips Allegra
SBL	35.0(10.4)	M 15, F 0	33.7(6.6)	M 15, F 0	Philips Intera
SDSU	14.7(1.8)	M 13, F 1	14.2(1.9)	M 16, F 6	GE MR750
STANFORD	10.0(1.6)	M 15, F 4	10.0(1.6)	M 16, F 4	GE Signa
TRINITY	16.8(3.2)	M 22, F 0	17.1(3.8)	M 25, F 0	Philips Achievao
UCLA	13.0(2.5)	M 48, F 6	13.0(1.9)	M 38, F 6	Siemens Trio
UM	13.2(2.4)	M 57, F 9	14.8(3.6)	M 56, F 18	GE Signa
USM	23.5(8.3)	M 46, F 0	21.3(8.4)	M 25, F 0	Siemens Trio
YALE	12.7(3.0)	M 20, F 8	12.7(2.8)	M 20, F 8	Siemens Trio

the location and site parameters of site i for value v, and the error term ϵ_{ijv} follow a normal distribution. The final ComBat-adjusted connectivity values can be defined as:

$$y_{ijv}^{\text{combat}} = \frac{y_{ijv} - \hat{\alpha}_v - \mathbf{I}_{ij}\hat{\boldsymbol{\beta}}_v - \gamma_{iv}^*}{\delta_{iv}^*} + \hat{\alpha}_v + \mathbf{I}_{ij}\hat{\boldsymbol{\beta}}_v, \tag{30}$$

where $\hat{\alpha}$, $\hat{\boldsymbol{\beta}}$, γ_{iv}^* and δ_{iv}^* are the empirical Bayes estimators for the parameters α, β, γ_{iv} and δ_{iv}.

6.2. The result of our proposed GraphCGC-Net

6.2.1. Performance of the graph generation with respect to the graph level

We compared our proposed model with several current state-of-the-art generative model approaches: α-GAN, ARVGE and VAE-GAN, which are most related to our work. **ARVGE** [28] is an auto-encoder model that regularizes the latent codes by adversarial training and forces the latent codes to match a specific prior distribution. α-**GAN** [24] combines the variational lower bound on the data likelihood with the density ratio trick, allowing the model to replace variational inference in a GAN-like fashion and to better capture the connection between VAE and GAN. **VAEGAN** [29] combines VAE with GAN using the KL divergence regularization terms to minimize the distance between the prior and the posterior of the latent code.

To show the comparison between the quality of graphs generated by our model and other comparable methods in terms of reality, we employ the trained generative models to generate graph data at first. Then we set a threshold value μ to remove the weaker edges from both the original and generated graphs, and compare the differences between them in terms of the following graph statistical properties:

1. **Degree property metrics**: largest connected component (LCC), characteristic path length (CPL), power law coefficient (PLC), clustering coefficient (CC).
2. **Edge distribution metrics**: relative edge distribution entropy (REDE) and node strength (NS).
3. **Centrality measures metrics**: closeness centrality (CC), betweenness centrality (BC) and degree centrality (DC).

The quantitative results achieved by different methods are reported in Table 2. The first row is the values of the real graphs and the rest is the absolute values of differences between the graphs generated by each generative model and the real graphs.

Firstly, it can be seen that compared with ARVGE, VAEGAN and α-GAN, GraphGAN++ obtains an improvement in terms of graph degree, edges and centrality attributes. The improvement demonstrates that our GAN framework presents a better generation performance than the state-of-the-art graph GAN methods due to the following advantages:

1. A more stable training process that guarantees the samples generated from the prior space are more realistic;
2. A local topology measure that preserves the graph structure;
3. A dual reconstruction loss that ensures a consistent generation of graph structure in the graph space.

6.2.2. Performance of the graph generation with respect to the data distribution

To intuitively understand the graph generation quality of GraphGAN++, we apply t-SNE [30] on both the original and generated graphs. To measure how close the distributions of generated and the original samples are on the 2-dimensional space, we evaluate the data distribution of each class in the original and generated samples through visualization. t-SNE is a non-linear dimensionality reduction algorithm used to visually explore high-dimensional data. Figure 4 shows the scatter plot results of t-SNE, with each point representing a graph sample.

From Fig. 4, we can observe that the distributions of the graphs generated by VAEGAN and α-GAN are inconsistent with the original data distribution. For ARVGE, the generated samples from the normal control class are mixed with the ASD class, which could confuse the following classifier model. Compared to the competing models, the results demonstrate that our GraphGAN++ model is able to synthesize graphs that are consistent with the original graph distribution and provides a better class discriminative ability.

6.2.3. Classification performance evaluation of the multiple graph GAN models

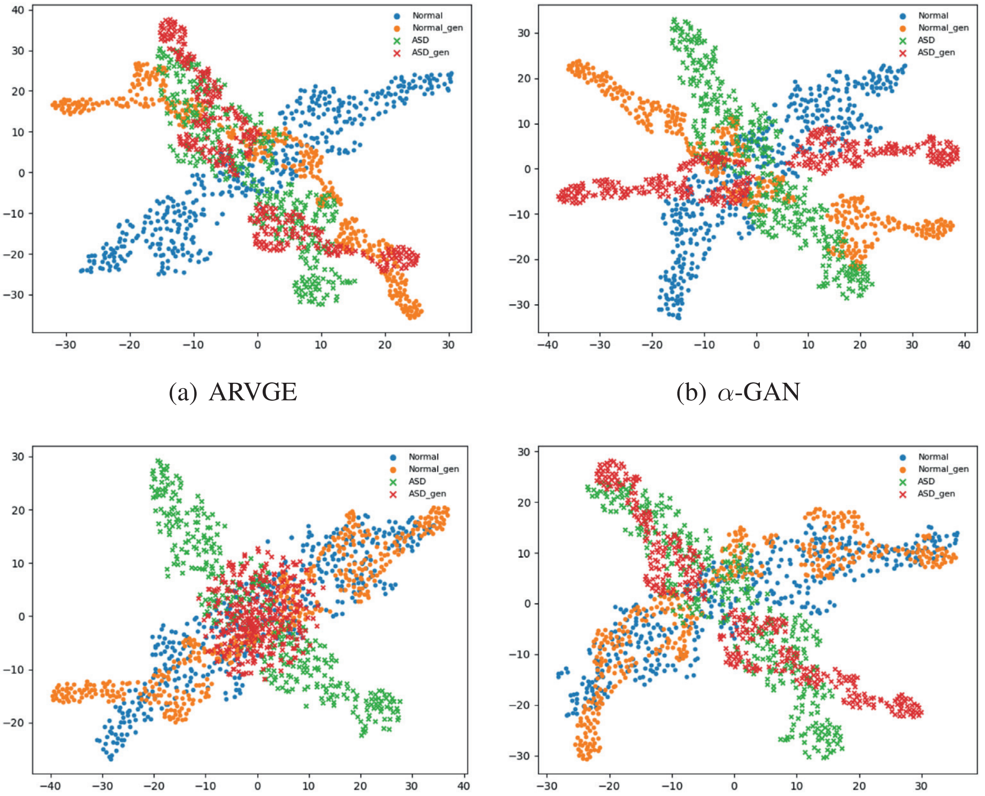
Data augmentation is typically utilized to increase the size of the effective training datasets when training deep neural networks for supervised learning tasks. In addition to the comparisons with respect to graph measure characteristics, we conduct rigorous experiments to verify whether the generated graphs can improve the classification performance. We compared our GraphGAN++ method with the following graph generation methods:

α **-GCNGAN** [31] is a graph generation model that solves the problems of mode collapse by combining GCN with α-GAN to generate graphs.

CONDGEN [32] is an end-to-end model that combines the power of GCN and VAEGAN to collapse the node features

Table 2
Performance evaluation of compared algorithms in terms of graph statistical properties.

Models	Degree				Edges		Centrality		
	LCC	CPL	PLC	CC	REDE	NS	BC	CC	DC
Real Graph	8	1.592	2.542	0.158	0.753	7.711	0.880	0.028	0.831
ARVGE [28]	**1.610**	0.311	0.842	0.670	0.110	5.010	0.442	0.046	0.473
α-GAN [24]	1.746	0.329	0.885	0.646	0.122	4.846	0.447	0.051	0.470
VAEGAN [29]	1.753	0.311	**0.795**	0.579	0.089	4.794	0.427	0.027	0.451
GraphGAN++	1.762	**0.280**	0.821	**0.375**	**0.059**	**4.508**	**0.413**	**0.025**	**0.432**



(a) ARVGE

(b) α-GAN

(c) VAEGAN

(d) GraphGAN++

Fig. 4. The embedded t-SNE representation of synthetic graphs generated by the various competing GAN models.

Table 3
Classification performance comparison of various methods. We compare the multiple GAN models on the original brain networks and the coarsened brain networks, respectively.

Model	Graph	ACC(%)	AUC(%)	SEN(%)	F1(%)	PREC(%)
EigenGCN [33]	Original	58.00	56.40	62.40	59.96	57.70
CONDGEN [32]	Original	59.80	57.70	61.40	60.90	60.40
α-GCNGAN [31]	Original	61.40	57.40	**76.20**	51.70	58.60
α-GAN [24]	Coarsened	66.42	68.40	66.35	66.38	66.62
ARVGE [28]	Coarsened	67.82	68.40	65.00	65.00	65.00
VAEGAN [29]	Coarsened	67.94	71.16	66.67	65.75	64.86
GraphCGC-Net	Coarsened	**70.45**	**72.76**	70.47	**70.39**	**70.38**

into permutation-invariant graph latent code and guarantee the permutation-invariant generation of the graphs.

We choose **EigenGCN** [33] as the baseline classification of the original graphs without generation. EigenGCN is a unified GCN model that integrates a pooling operator, EigenPooling, to utilize the node features and local structures during the pooling process. From the results in Table 3, we can draw the following conclusions:

1. It is clearly observed that the synthetic graph data generated by different models provide additional contributions to the classification performance. The graph-based generative models working on the original brain networks are also helpful for the classification model, but the improvement is limited. The reason is due to the fact that generative models are negatively influenced by the noisy edges in the original graph data. Although the original brain networks can be reduced by a threshold value, the binarization of graph data results in that the useful information is lost in the original brain networks. In contrast, the coarsened graphs by MGC can better preserve the functional connectivity information and achieve better classification performance.

2. The results prove our hypothesis that the GAN models based on the coarsened graphs can improve the classification performance. Compared with EigenGCN, our GraphCGC-Net obtained an accuracy improvement of 12.45%.

3. It is obvious that the graph samples generated by GraphGAN++ provide more sufficient information for the classification compared with other graph generation models.

6.3. Comparison with the state-of-the-art methods for ASD diagnosis

To verify the effectiveness of our GraphCGC-Net on the ASD classification, we compare our method with several recent state-of-the-art methods:

ASD_DiagNet [8] is a joint learning model combing an auto-encoder and a single layer perceptron to improve the quality of extracted FC features and to optimize the parameters of the classifier.

DAE [34] is an auto-encoder-based model that used two denoising auto-encoders to reduce the dimensionality and eliminate noise from the correlation matrices, and the weight parameters used for denoising were further used to initialize the parameters for the classifier.

S **-GCN** [35] is a framework that uses the siamese graph convolutional neural network to learn the structural similarity between two graphs in a supervised manner.

ST **-GCN** [36] is a framework for analyzing rs-fMRI data that utilizes spatio-temporal graph convolutional to learn the importance of graph edges to gain insight into the FCs.

BrainGNN [9] is an interpretable graph neural network-based framework for fMRI classification tasks that jointly learns ROI clustering and the downstream whole-brain fMRI prediction.

MVS-GCN [19] is a multi-view graph convolutional network-based model using graph structure learning and multi-task graph embedding learning to diagnose brain disorders and to identify the critical subnetworks.

1D_CNN [37] is a simple method to transform the high-dimensional rs-fMRI data into the time series and use 1D-convolutional neural networks for classification.

3DCNN_1D [38] is a model that uses 3DCNNs to extract voxel-level features from the full 4-D fMRI data and use 1D_CNNs to process the spatio-temporal features for classification.

3DCNN_C_LSTM [38] is an end-to-end model that can extract local and global spatio-temporal features from the full 4-D fMRI data using 3DCNNs and 3D-convolutional long short-term memory (LSTM).

To more comprehensively evaluate our model, we compare GraphCGC-Net with two groups of state-of-the-art neural network-based methods, involving GCN-based methods (EigenGCN, BrainGNN, *ST*-GCN and *S*-GCN) and autoencoder based methods ASD_diagNet and DAE. We also compare GraphCGC-Net with two traditional machine learning approaches (SVM, RF) that use the flattened PC data (i.e., $\frac{n\times(n-1)}{2}$, where n is the amount of ROIs) as features for classification. In particular, we compare with the CNN-based approaches, which involve extracting the voxel-level features from 4D fMRI image data without converting to FBNs. Experimental results of the 10-fold cross validation are reported in Table 4, and the ROC curves are plotted in Fig. 5. We highlight the following observations:

1. It can be observed that the GraphCGC-Net model for ASD classification achieves the best classification performance with an accuracy of 70.45%, which is about 5.24%, 3.2%, 9.3% and 11.19% improvement in ACC compared to the previous traditional machine learning models, autoencoder models, GCN-based models and CNN-based models, respectively.

2. The autoencoder-based and traditional machine learning methods are dedicated to flatten FC information and work on extracting global features in brain networks. However, these methods have an insufficient feature representation capability and ignore the spatial structure information of FBNs. The CNN-based approaches use a combination of CNN and LSTM to obtain local and global spatio-temporal information from the 4D fMRI images. They perform worse since multi-site data are obtained from different scanners and with different acquisition parameters. The heterogeneity of the data hinders the advantages of CNN-based approaches. Moreover, the reason why 1D_CNN performs better than 3DCNN is that the 3DCNN-based model cannot obtain representative voxel-level features from the heterogeneous data.

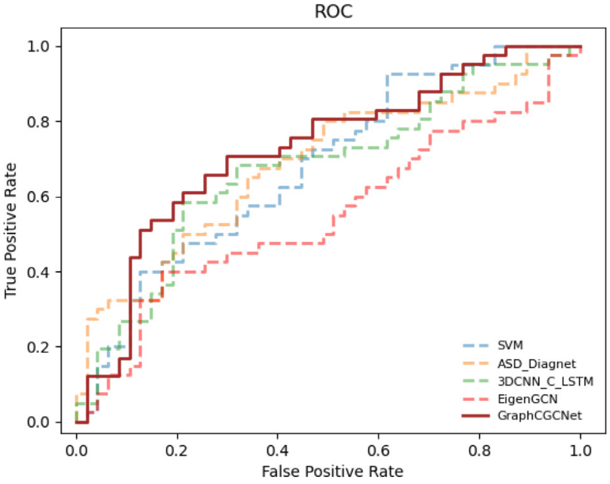


Fig. 5. The ROC of multiple computing methods.

3. Moreover, the graph representation learning methods with deep learning generally obtain worse prediction results than the traditional machine methods. The reason is that the complex and inconsistent graph structure in brain networks limits the classification performance of the GCN models. Another reason is that training GCN requires a sufficiently large training data. The limited data hinders the embedding learning of GCN. Overall, the results suggest that GCN classification with the augmentation data generated by our GraphGAN++ can enable more accurate performance compared to the competing methods.

6.4. Discussion

6.4.1. The influence of positive and negative correlation in MGC

At first, we explore the influence of positive correlation and negative correlation in the classification. Specifically, we input positive and negative graphs as a separate branch into the GCN model to explore the separate importance of positive and negative connections in the FBNs. In addition, the binarized single graph was input into the MGC classification model by setting a threshold value $\mathbf{T}_r$ using Eq. (31) and then verifying whether the threshold behavior would lose information, where $w_{i,j}$ are the edge weights. The results are summarized in Table 5:

$$w_{i,j} = \begin{cases} 1, & \text{if } \left|w_{i,j}\right| \geq \mathbf{T}_r \\ 0, & \text{if } \left|w_{i,j}\right| < \mathbf{T}_r \end{cases} \tag{31}$$

1. We can find that the MGC using the positive correlations are superior to the model with the negative ones, resulting in a 2.30% increase in ACC. The observation shows that the positive correlations are more critical for the classification tasks.

2. Moreover, the MGC results when the positive and negative correlations are learned by one branch (Pos_Neg_nosplit) shows that mixing positive and negative correlations together can negatively affect the classification performance. More importantly, we can observe that the performance of the original MGC

Table 4
Comparison with the state-of-the-art methods for ASD diagnosis.

Category	Model	ACC(%)	AUC(%)	SEN(%)	F1(%)	PREC(%)	CV
ML	SVM	68.76	67.98	57.56	62.87	69.66	10-CV
	RF	61.66	60.35	42.96	50.42	62.54	10-CV
Auto-encoder	ASD_DiagNet [8]	69.00	68.50	61.05	64.54	68.85	10-CV
	DAE [34]	65.50	66.89	58.33	63.81	**70.43**	10-CV
CNN	1D_CNN [37]	64.00	-	-	64.00	-	5-CV
	3DCNN_1D [38]	54.00	-	-	50.00	-	5-CV
	3DCNN_C_LSTM [38]	59.78	67.77	48.64	52.74	57.60	10-CV
GCN	MVS-GCN [19]	69.89	69.11	70.18	-	-	10-CV
	S-GCN [35]	64.73	64.33	67.54	63.61	60.12	10-CV
	BrainGNN [9]	61.65	60.79	61.84	61.31	60.79	10-CV
	ST-GCN [36]	53.12	56.88	56.51	53.24	50.33	10-CV
	EigenGCN [33]	56.34	59.50	53.20	56.49	60.21	10-CV
Our method	GraphCGC-Net	**70.45**	**72.76**	**70.47**	**70.39**	70.38	10-CV

Table 5
The influence of positive (Pos) and negative (Neg) correlations. The Pos/Neg_split denotes that the positive and negative correlations are learned by individual branches, and Pos/Neg_nosplit means that two correlations are not split.

Method	Pos/Neg	ACC(%)	AUC(%)	SEN(%)	F1(%)	PREC(%)
No thresholding	Pos	62.69	61.68	61.68	61.24	62.82
	Neg	60.39	59.02	59.02	56.88	57.98
	Pos/Neg_split	**65.32**	**68.95**	**65.22**	**65.13**	**65.31**
	Pos/Neg_nosplit	61.99	65.32	62.10	62.06	62.02
Thresholding	$\mathbf{T}_r = 0.5$	60.13	61.82	59.72	59.48	60.13
	$\mathbf{T}_r = 0.6$	59.47	63.09	59.21	59.00	59.39

Table 6
The effectiveness of each regularization in the MGC model.

Model	ACC(%)	AUC(%)	SEN(%)	F1(%)	PREC(%)
MGC -w only CE	59.87	64.95	60.14	60.39	60.51
MGC-w/o $\mathbf{L}_n$	62.10	67.85	62.39	62.45	62.64
MGC-w/o $\mathbf{L}_b$	62.47	66.98	62.97	63.14	63.71
MGC -w/o $\mathbf{L}_o$	63.58	68.92	64.19	64.97	**65.66**
MGC -w/o $\mathbf{L}_2$	64.30	63.91	64.37	63.89	63.92
MGC	**65.13**	**68.95**	**65.22**	**65.32**	65.31

Table 7
The effectiveness of each component in our GraphGAN++ model.

Component	ACC(%)	AUC(%)	SEN(%)	F1(%)	PREC(%)
GraphGAN++ w/o DR	67.05	69.11	66.81	66.83	66.88
GraphGAN++ w/o WD	68.18	68.34	67.88	67.92	68.03
GraphGAN++ w/o LT	69.23	71.16	68.65	68.71	69.16
GraphGAN++ w/o HM	67.81	69.91	66.78	66.20	69.53
GraphGAN++	**70.45**	**72.76**	**70.47**	**70.39**	**70.38**

(Pos/Neg_split) performs much better, which indicates that the positive and negative correlations should be considered individually.

3. Additionally, we set a simple threshold to binarize the absolute value of the edge weights. We observe that the average performance of MGC decreases by 5.52% with respect to ACC, indicating that generating a binary graph by thresholding can ignore the information of positive and negative correlations.

6.4.2. The effectiveness of regularization terms in MGC

In this section, we perform several experiments to compare the prediction accuracy without different regularization terms to show the impact of each regularization term. As shown in Table 6, the results demonstrate that all regularizations are complementary and effective. We can observe that adding $\mathbf{L}_b$ and $\mathbf{L}_n$ constraints is more helpful for MGC. In addition, removing $\mathbf{L}_o$ from the MGC leads to ACC decreases 1.55%, which may be due to the lack of constraints to control the overlap size between the clusters. It also indicates that we should control the overlap of ROIs between different regions, because it is beneficial for the classification tasks.

6.4.3. The impact of the different components in our proposed GraphGAN++

To demonstrate the effectiveness of each component in GraphGAN++, we removed each one from the GraphGAN++ to investigate the relative contribution in the classification. Specifically, the main components of GraphGAN++ include the dual reconstruction (DR) mechanism, the Wasserstein distance measure (WD), the local topology constraint (LT) and the homeomorphic mapping (HM).

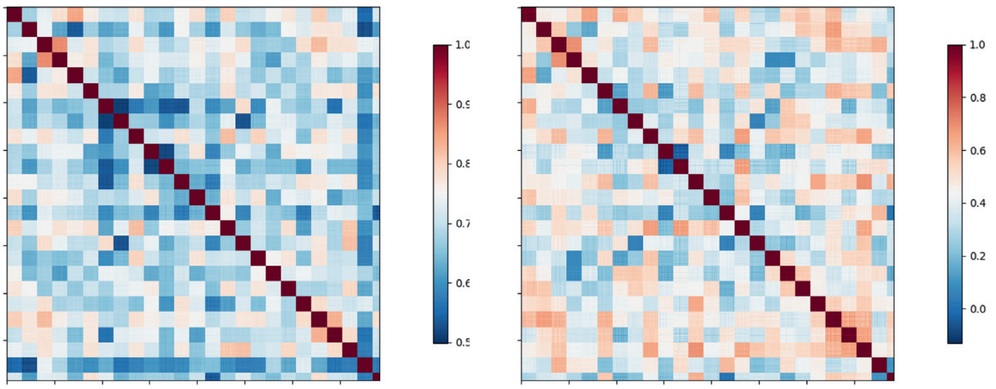
The results in Table 7 show that the DR component contributes to the effectiveness and robustness of the whole model. Specifically, we find that removing the DR component from GraphGAN++ leads to an optimization difficulty. Moreover, the worst classification result demonstrates that the dual reconstruction mechanism is helpful to preserve the same structure with the original graphs. In addition, it can be observed that removing any component leads to a performance decrease. GraphGAN++ consistently outperforms the other variants, indicating the effectiveness of combining the proposed components together.

6.4.4. The impact of the code discriminator

To understand the performance of our code discriminator through the Wasserstein distance instead of the KL divergence, we compare the latent embedding generated by our model and the one uses the KL divergence term. We plot the covariance matrix in Fig. 6 with the latent codes. We see that the two methods have different side effects: the latent codes obtained using the code discriminator in our model are decorrelated, this is expected; while the ones obtained using the empirical KL are entangled. The code discriminator achieves better disentangling, which guarantees the diversity of the generated graphs with less correlated embedding.

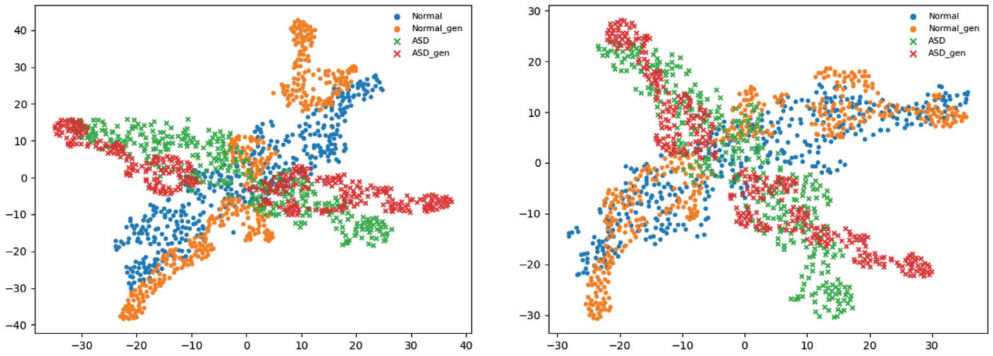
6.4.5. The impact of the topology loss

In order to explore the useful topology information, we attempt two other different measures including: closeness centrality (CC) and degree centrality (DC). For an undirected graph $\mathbf{S}$ with m nodes, the closeness centrality directly relates to the cardinality of



(a) Code discriminator (b) KL approximation

Fig. 6. Covariance matrices with the latent embedding from GraphGAN++ model using a code discriminator and the empirical KL approximation.



(a) GraphGAN++ without homeomorphic mapping (b) GraphGAN++ with homeomorphic mapping

Fig. 7. The influence of the homeomorphic mapping by observing the embedded *t*-SNE representation of graphs.

Table 8
The impact of different measures on preserving the most critical edges.

Model	$\text{MAE}_{\text{edges}}$	MAE_{BC}	MAE_{CC}	MAE_{DC}
-	3.776	0.125	0.596	0.250
BC	**0.777**	**0.082**	0.082	0.186
CC	1.513	0.094	0.136	**0.152**
DC	2.741	0.075	**0.075**	0.192

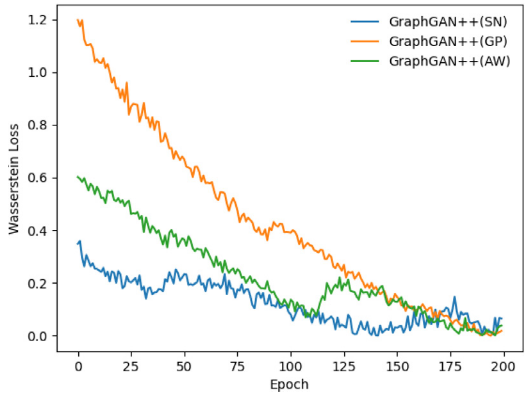


Fig. 8. The loss curves of generators in different generation models.

the shortest path between two nodes. The CC is defined as:

$$C_{cc}(n_i) = \frac{1}{\sum_{j=1}^{m} d(i,j)}, \tag{32}$$

where $d(i,j)$ defines the distance between two nodes i and j. The DC measures the degree of connectivity doc_{ij} between a node i and the other $m-1$ nodes, as follows:

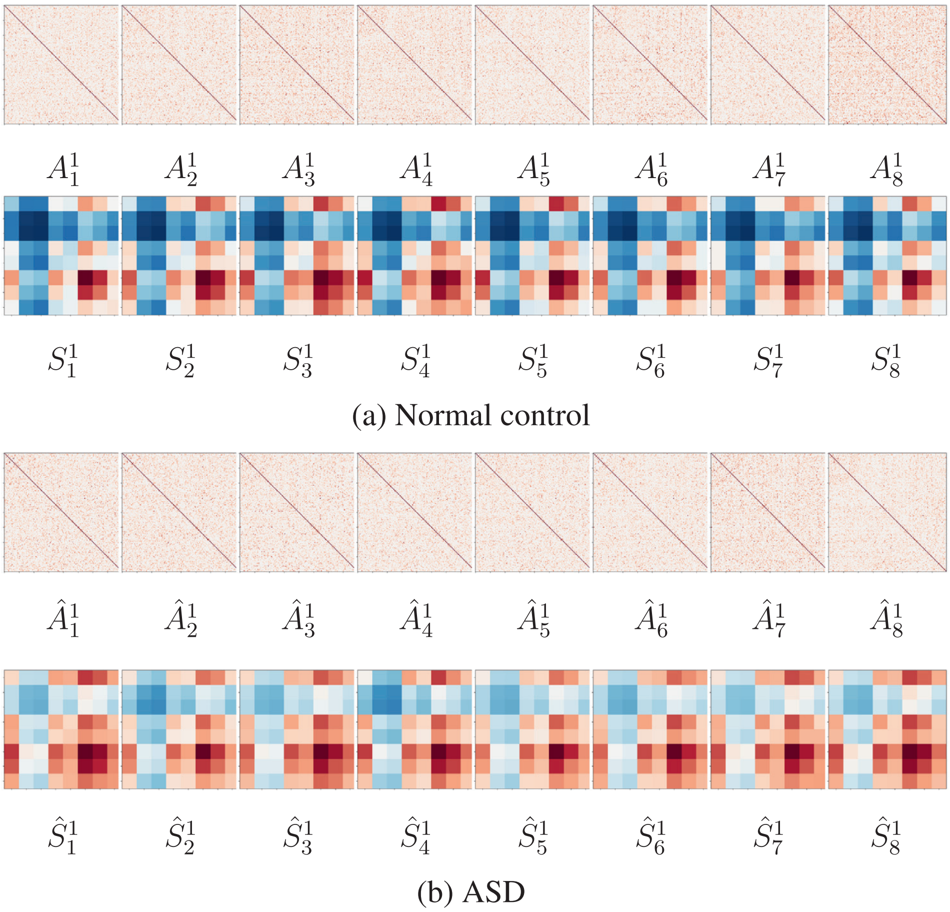
$$C_{dc}(n_i) = \sum_{j=1}^{m} doc_{ij}(i \neq j). \tag{33}$$

To evaluate the performance of the multiple topological loss, we calculated the mean absolute error (MAE) of the edges between the real and generated graphs, as well as the MAE of centrality score between the generated and real graphs.

Table 8 shows the incorporation of the topology measures can effectively improve the quality of the generated graphs. Moreover, BC is generally better than the two other measures on graph generation, which implies that considering the frequency of nodes on the shortest path during the graph generation is more effective for topology preservation.

6.4.6. The impact of the proposed homeomorphic mapping

The purpose of homeomorphic mapping is to achieve smooth interpolations of the generated samples. To evaluate the properties of the generated graphs, the number of the generated samples is set to the number of the original training datasets. Figure 7 illustrates the embedded representation of the generated and true data with and without distribution consistency mechanism. It can be observed that the distribution consistency mechanism is able to



(a) Normal control

(b) ASD

Fig. 9. Original graphs and coarsened graphs of the selected samples. The first row represents the original graphs, while the second one is the coarsened graphs.

Table 9
The effectiveness of each discriminator in GraphGAN++. The first row is the values of the real graphs and the rest are the absolute values of the difference between the generated graphs and the real graphs.

Method	Degree				Edge		Centrality		
	LCC	CPL	PLC	CC	REDE	NS	BC	CC	DC
Real Graph	8	1.592	2.542	0.158	0.753	7.711	0.880	0.028	0.831
GraphGAN++ w/o CD	0.732	0.539	1.718	0.690	0.227	2.869	0.025	0.149	0.178
GraphGAN++ w/o GD	0.411	0.514	3.932	0.372	0.206	1.489	0.028	0.105	0.151
GraphGAN++	**0.330**	**0.280**	**0.821**	**0.375**	**0.059**	**1.197**	**0.024**	**0.025**	**0.143**

synthesize graphs that are consistent with the original graph distribution thanks to the proposed homeomorphic mapping, which shows the advantage of our method to improve the modeling distribution of complex data.

6.4.7. The effectiveness of the discriminators in GraphGAN++

We remove each discriminator from GraphGAN++ to demonstrate the effectiveness of the graph discriminator (GD) and code discriminator (CD) and to validate the reality of the generated graphs. Specifically, we remove the code discriminator and adopt the KL divergence term to minimize the distribution between the prior and the posterior of the latent code. Moreover, we binarize the generated graphs by thresholding and measure the impact of these two discriminators on the model generation performance in terms of graph statistics.

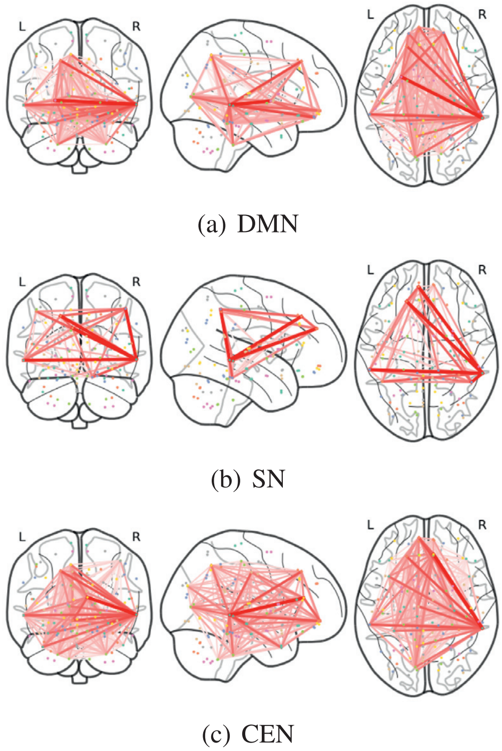
Table 9 shows that the incorporation of the code discriminator and the graph discriminator can effectively enhance the quality of the generated graphs. Moreover, we can conclude that the adversarial training approach can present a better performance when forcing the latent code distribution to match a specific prior distribution.

6.4.8. Convergence performance evaluation of different training strategies

GAN models are prone to issues in training such as instability and lack of convergence. To explore the stability and convergence of GraphGAN++, we choose two other generation algorithms [39,40] that also focus on the discriminative models to rectify the instability problems. We then replace the Lipschitz constant of GraphGAN++ with the constraint or loss function proposed in both algorithms, and compare the convergence behavior.

SNGAN [39] introduces a spectral normalization (SN) constraint $\mathbf{C}_{sn}$ to normalize the weight matrix and stabilize the training of the discriminator network.

AWGAN [40] introduces an adaptive weighted (AW) loss $\mathbf{L}_{aw}$ function to adaptively select weights for training a discriminator in the direction of gradient that benefits model's stability (Table 10).



(a) DMN

(b) SN

(c) CEN

Fig. 10. The top 3 subnetworks identified by our MGC.

From Fig. 8, we can find that all the losses of three generators decrease and approach to zero, which indicates that our GraphGAN++ is able to converge and has stability in training, regardless of any strategy of stabilizing the training process of GANs.

Table 10
The details of models' discriminator.

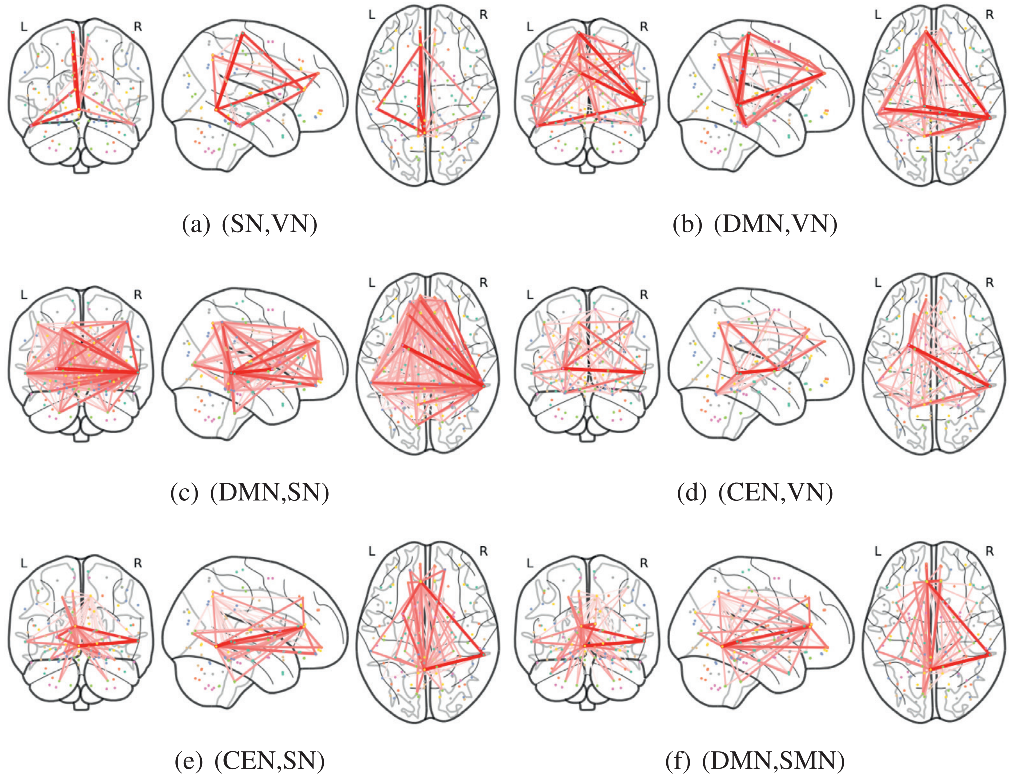
Model	Mechanism
GraphGAN++(GP)	$\mathbf{L}_{gp}$
GraphGAN++(AW)	$\mathbf{L}_{aw}$
GraphGAN++(SN)	$\mathbf{C}_{sn}$

6.5. Interpretability

In this paper, we evaluate the effectiveness of multi-graph clustering. At first, we visualize two groups of samples with ASD and normal controls to explore the differences between the coarsened graphs and the original graphs. In Fig. 9, we can observe that each subject of the original graphs is very irregular. Through the multi-graph clustering, the inconsistency of all subjects is mitigated and the indicative edges are highlighted. Moreover, by comparing the results in Fig. 9(a) and (b), we can notice that the discriminability between ASD subjects and NCs is enhanced on the coarsened graphs.

In addition, to enhance the interpretability of MGC, we exploit the corresponding relationship between the brain regions in the CC200 atlas and the functional subnetworks to identify the subnetworks that are most informative for predictive targets. At first, the label of each brain region in CC200 was aligned according to the maximum overlap with the AAL atlas. Then, we cluster the connectivity values belonging to the same brain regions defined in the AAL template. Specifically, given a subnetwork Sn_p, we calculate the average indicator factor of an edge within it. The subnetwork Sn_p score is calculated as:

$$score_p = \frac{2}{\mid R_p \mid^2} \sum_{i,j \in Sn_p \wedge i \in S_e \wedge j \in S_d \wedge e \neq d} f_i \cdot f_j, \qquad (34)$$



(a) (SN,VN)

(b) (DMN,VN)

(c) (DMN,SN)

(d) (CEN,VN)

(e) (CEN,SN)

(f) (DMN,SMN)

Fig. 11. The top 6 inter-subnetworks identified by our MGC.

Table 11
The top 3 subnetworks and the top 6 inter-subnetworks selected and the corresponding scores optimized by MGC.

Subnetworks name	Scores	Inter-subnetworks name	Scores
DMN	0.0023	(SN, VN)	0.0018
SN	0.0019	(DMN, VN)	0.00067
CEN	0.0017	(DMN,SN)	0.00065
-	-	(CEN,VN)	0.00061
-	-	(CEN, SN)	0.00056
-	-	(DMN, SMN)	0.00053

where f_i and f_j indicates the importance of node i and j, S_e and S_d are the supernodes obtained by MGC, R_p is the brain regions belonging to the subnetwork p. Therefore, the $score_p$ can be explained as the sum of the products of the importance of any two nodes belonging to the subnetwork Sn_p but different cluster. Moreover, we also explore the important inter-subnetwork correlations, the score of two subnetworks Sn_p and Sn_q are computed as follows:

$$score_{pq} = \frac{2}{\mid R_p \mid\mid R_q \mid} \sum_{i \in Sn_p, j \in Sn_q \wedge i \in S_e \wedge j \in S_d \wedge e \neq d} f_i \cdot f_j, \tag{35}$$

where the $score_{pq}$ can also be explained as the sum of the products of the importance of any two nodes belonging to the subnetwork Sn_p and Sn_q.

In the experiment, we evaluate six networks including the default mode network (DMN), the central executive network (CEN), the salience network (SN), the auditory network (AN), the somatomotor network (SMN) and the visual network (VN). Through analyzing the scores calculated by Eq. (34) and Eq. (35), the top subnetworks and inter-subnetworks selected by our model are shown in Table 11. We can find that the top three subnetworks are DMN, SN and CEN, as shown in Fig. 10. It is consistent with the previous neuroscience research that ASD is associated with the altered functional connectivity of the CEN, DMN and SN, which are thought to be central to the symptomatology of ASD [41]. In addition, we obtain the top six inter-subnetworks including (SN, VN), (DMN, VN), (DMN, SN), (CEN, VN), (CEN, SN), (DMN,SMN), as shown in Fig. 11. This is also consistent with the discoveries that the triple networks including CEN, SN and DMN are dysfunctional in the ASD [42]. The results also demonstrate that visual engagement difficulties are an early signature of ASD, and VN play a central role in initiating the transition between SN, CEN and DMN. It is a process necessary for attention and flexible cognitive control [43]. In particular, it has been reported that the FC of VN and SMN is disrupted in ASD, and the subjects who are less sensitive to visual feedback during motor learning will display more severe autistic traits [44].

7. Conclusion

Recently, GCN-based deep learning methods have made a massive breakthrough in the field of brain networks. However, training an accurate GCN model for brain networks faces several challenges, including high dimensional and noisy correlation in the brain networks and limited labeled training data. Indeed, a large amount of labeled data is the essential cornerstone to reach this success. However, in the field of medical imaging, it is challenging or often impossible to obtain a sufficient amount of data for training. Consequently, to preserve the continuous information in brain networks data, we remove the noisy connections through a multi-graph clustering approach at first. In addition, we propose a novel GraphGAN++ model, which can generate realistic brain networks with coarsened structure starting with a small amount of training samples. By combining the strengths of multi-graph clustering and GraphGAN++ components, the proposed GraphCGC-Net significantly improved upon the state-of-the-art results in ASD classification on benchmark datasets. In the future work, we will evaluate our method on the other brain disorders, such as major depressive disorder. Moreover, the proposed generative model is unsupervised during adversarial learning, which may lack sufficient discriminative ability for graph classification. We will construct an end-to-end framework that simultaneously trains the GAN and GCN, thus the classification loss will help the GAN to produce a more realistic brain networks. Furthermore, tremendous progress has been made in neural architecture search (NAS) approaches for automatically searching the network architectures. Several efforts have yielded significant results in the search architectures for CNNs to achieve better performance [45–47], and we will leverage the NAS methods to design more efficient architectures for the GANs and GCNs to improve the capability of ASD diagnosis.

Declaration of Competing Interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in or the review of the manuscript entitled, “Collaborative Learning of Graph Generation, Clustering and Classification for Brain Networks diagnosis”.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (No.62076059) and the Science Project of Liaoning Province (2021-MS-105).

References

[1] G.S. Bajestani, M. Behrooz, A.G. Khani, M. Nouri-Baygi, A. Mollaei, Diagnosis of autism spectrum disorder based on complex network features, Comput. Methods Programs Biomed. 177 (2019) 277–283, doi:10.1016/j.cmpb.2019.06.006.

[2] A.S. Heinsfeld, A.R. Franco, R.C. Craddock, A. Buchweitz, F. Meneguzzi, Identification of autism spectrum disorder using deep learning and the ABIDE dataset, NeuroImage Clin. 17 (November 2016) (2018) 16–23, doi:10.1016/j.nicl.2017.08.017.

[3] T. Iidaka, Resting state functional magnetic resonance imaging and neural network classified autism and control, Cortex 63 (2015) 55–67.

[4] M. Khosla, K. Jamison, G.H. Ngo, A. Kuceyeski, M.R. Sabuncu, Machine learning in resting-state fMRI analysis, Magn. Reson. Imaging 64 (June) (2019) 101–121 arXiv preprint arXiv:1812.11477, doi:10.1016/j.mri.2019.05.031.

[5] Y. Zhao, C. Dong, G. Zhang, Y. Wang, X. Chen, W. Jia, Q. Yuan, F. Xu, Y. Zheng, EEG-Based Seizure detection using linear graph convolution network with focal loss, Comput. Methods Programs Biomed. 208 (2021) 106277, doi:10.1016/j.cmpb.2021.106277.

[6] J. Shi, R. Wang, Y. Zheng, Z. Jiang, H. Zhang, L. Yu, Cervical cell classification with graph convolutional network, Comput. Methods Programs Biomed. 198 (2021) 105807, doi:10.1016/j.cmpb.2020.105807.

[7] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, 2017, pp. 1–14. arXiv preprint arXiv:1609.02907.

[8] T. Eslami, V. Mirjalili, A. Fong, A.R. Laird, F. Saeed, et al., ASD-DiagNet: a hybrid learning approach for detection of autism spectrum disorder using fMRI data, Front. Neuroinform. 13 (2019) arXiv preprint arXiv:1904.07577, doi:10.3389/fninf.2019.00070.

[9] X. Li, Y. Zhou, S. Gao, N. Dvornek, M. Zhang, J. Zhuang, S. Gu, D. Scheinost, L. Staib, P. Ventola, J. Duncan, BrainGNN: interpretable brain graph neural network for fMRI analysis, bioRxiv (2020), doi:10.1101/2020.05.16.100057.

[10] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: 34th International Conference on Machine Learning, ICML 2017, vol. 1, 2017, pp. 298–321.

[11] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, et al., Improved training of wasserstein GANs, in: Advances in Neural Information Processing Systems, vol. 2017-Decem, 2017, pp. 5768–5778. 1704.00028. https://github.com/igul222/improved_wgan_training.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144 1406.2661, doi:10.1145/3422622.

[13] T. Pang, J.H.D. Wong, W.L. Ng, C.S. Chan, Semi-supervised GAN-based radiomics model for data augmentation in breast ultrasound mass classification, Comput. Methods Programs Biomed. 203 (2021) 106018, doi:10.1016/j.cmpb.2021.106018.

[14] T. Shen, K. Hao, C. Gou, F.Y. Wang, Mass image synthesis in mammogram with contextual information based on GANs, Comput. Methods Programs Biomed. 202 (2021) 106019, doi:10.1016/j.cmpb.2021.106019.

[15] M. Ingalhalikar, S. Shinde, A. Karmarkar, A. Rajan, D. Rangaprakash, G. Deshpande, et al., Functional connectivity-based prediction of Autism on site harmonized ABIDE dataset, IEEE Trans. Biomed. Eng. (2021), doi:10.1109/TBME.2021.3080259.

[16] M. Yu, K.A. Linn, P.A. Cook, M.L. Phillips, M. McInnis, M. Fava, M.H. Trivedi, M.M. Weissman, R.T. Shinohara, Y.I. Sheline, et al., Statistical harmonization corrects site effects in functional connectivity measurements from multi-site fMRI data, Hum. Brain Mapp. 39 (11) (2018) 4213–4227, doi:10.1002/hbm.24241.

[17] W. Tang, Z. Lu, I.S. Dhillon, Clustering with multiple graphs, in: Proceedings - IEEE International Conference on Data Mining, ICDM, 2009, pp. 1016–1021, doi:10.1109/ICDM.2009.125.

[18] B. Jie, D. Zhang, C.Y. Wee, D. Shen, Topological graph kernel on multiple thresholded functional connectivity networks for mild cognitive impairment classification, Hum. Brain Mapp. 35 (7) (2014) 2876–2897, doi:10.1002/hbm.22353.

[19] G. Wen, P. Cao, H. Bao, W. Yang, T. Zheng, O. Zaiane, MVS-GCN: a prior brain structure learning-guided multi-view graph convolution network for autism spectrum disorder diagnosis, Comput. Biol. Med. (2022) 105239.

[20] A. Kazeminejad, R.C. Sotero, The importance of anti-correlations in graph theory based classification of autism spectrum disorder, Front. Neurosci. 14 (2020), doi:10.3389/fnins.2020.00676.

[21] A. Sólon, A. Rosa, R.C. Craddock, A. Buchweitz, F. Meneguzzi, NeuroImage: clinical identification of autism spectrum disorder using deep learning and the ABIDE dataset, Neuroimage Clin. 17 (June 2017) (2018) 16–23, doi:10.1016/j.nicl.2017.08.017.

[22] A. Bessadok, M.A. Mahjoub, I. Rekik, Brain graph synthesis by dual adversarial domain alignment and target graph prediction from a source graph, Med. Image Anal. 68 (2021), doi:10.1016/j.media.2020.101902.

[23] G. Kwon, C. Han, D. shik Kim, Generation of 3D brain MRI using auto-encoding generative adversarial networks, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11766, LNCS, 2019, pp. 118–126, doi:10.1007/978-3-030-32248-9_14. 1908.02498.

[24] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, S. Mohamed, Variational approaches for auto-encoding generative adversarial networks, CoRR (2017) 1706.04987.

[25] T.N. Kipf, M. Welling, Variational graph auto-encoders, arXiv preprint arXiv:1611.07308(2016).

[26] R.C. Craddock, G.A. James, P.E. Holtzheimer, X.P. Hu, H.S. Mayberg, A whole brain fMRI atlas generated via spatially constrained spectral clustering, Hum. Brain Mapp. 33 (8) (2012) 1914–1928, doi:10.1002/hbm.21333.

[27] A. Ashourvan, S. Gu, M.G. Mattar, J.M. Vettel, D.S. Bassett, A. Abraham, M.P. Milham, A. Di Martino, R.C. Craddock, D. Samaras, B. Thirion, G. Varoquaux, M. Giulia, T. Aw, D.V.D. Ville, Deriving reproducible biomarkers from multi-site resting-state data: an autism-based example, Neuroimage 157 (October 2016) (2016) 1–37 1609.01015.

[28] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, C. Zhang, et al., Adversarially regularized graph autoencoder for graph embedding, in: IJCAI International Joint Conference on Artificial Intelligence, vol. 2018-July, 2018, pp. 2609–2615, doi:10.24963/ijcai.2018/362. 1802.04407.

[29] A.B.L. Larsen, S.K. Sønderby, H. Larochelle, O. Winther, et al., Autoencoding beyond pixels using a learned similarity metric, in: 33rd International Conference on Machine Learning, ICML 2016, vol. 4, 2016, pp. 2341–2349. 1512.09300. https://www.researchgate.net/publication/288889700

[30] C.R. García-Alonso, L.M. Pérez-Naranjo, J.C. Fernández-Caballero, Multiobjective evolutionary algorithms to identify highly autocorrelated areas: the case of spatial distribution in financially compromised farms, Ann. Oper. Res. 219 (1) (2014) 187–202, doi:10.1007/s10479-011-0841-3.

[31] S. Zhang, P. Cao, L. Dou, J. Yang, D. Zhao, An auto-encoding generative adversarial networks for generating brain network, in: The Fourth International Symposium on Image Computing and Digital Medicine, 2020, pp. 14–18.

[32] C. Yang, P. Zhuang, W. Shi, A. Luu, P. Li, et al., Conditional structure generation through graph variational generative adversarial nets, Advances in Neural Information Processing Systems, vol. 32, 2019.

[33] A. Brasoveanu, M. Moodie, R. Agrawal, Graph convolutional networks with eigenpooling, CEUR Workshop Proceedings, vol. 2657, CEUR-WS, 2020, pp. 1–9, doi:10.1145/nnnnnnn.nnnnnnn.

[34] A.S. Heinsfeld, A.R. Franco, R.C. Craddock, A. Buchweitz, F. Meneguzzi, Identification of autism spectrum disorder using deep learning and the ABIDE dataset, Neuroimage Clin. 17 (August) (2018) 16–23, doi:10.1016/j.nicl.2017.08.017.

[35] S.I. Ktena, S. Parisot, E. Ferrante, M. Rajchl, M. Lee, B. Glocker, D. Rueckert, Metric learning with spectral graph convolutions on brain connectivity networks, Neuroimage 169 (December) (2018) 431–442, doi:10.1016/j.neuroimage.2017.12.052.

[36] S. Gadgil, Q. Zhao, E. Adeli, A. Pfefferbaum, E.V. Sullivan, K.M. Pohl, Spatio-temporal graph convolution for functional MRI analysis (2020).

[37] A. El Gazzar, L. Cerliani, G. van Wingen, R.M. Thomas, Simple 1-D convolutional networks for resting-state fMRI based classification in autism, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019a, pp. 1–6.

[38] A. El Gazzar, M. Quaak, L. Cerliani, P. Bloem, G. van Wingen, R.M. Thomas, A hybrid 3DCNN and 3DC-LSTM based model for 4D spatio-temporal fMRI data: an ABIDE autism classification study, OR/MLCN@ MICCAI, 2019b.

[39] T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks, arXiv preprint arXiv:1802.05957(2018).

[40] V. Zadorozhnyy, Q.S. Cheng, Q. Ye, Adaptive weighted discriminator for training generative adversarial networks, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 4779–4788.

[41] K.E. Lawrence, L.M. Hernandez, H.C. Bowman, N.T. Padgaonkar, E. Fuster, A. Jack, E. Aylward, N. Gaab, J.D. Van Horn, R.A. Bernier, et al., Sex differences in functional connectivity of the salience, default mode, and central executive networks in youth with ASD, Cereb. Cortex 30 (9) (2020) 5107–5120.

[42] V. Menon, Large-scale brain networks and psychopathology: a unifying triple network model, Trends Cogn. Sci. 15 (10) (2011) 483–506, doi:10.1016/j.tics.2011.08.003. https://www.sciencedirect.com/science/article/pii/S1364661311001719

[43] M.V. Lombardo, L. Eyler, A. Moore, M. Datko, C.C. Barnes, D. Cha, E. Courchesne, K. Pierce, Default mode-visual network hypoconnectivity in an autism subtype with pronounced social visual engagement difficulties, Elife 8 (2019) e47427.

[44] M.B. Nebel, A. Eloyan, C.A. Nettles, K.L. Sweeney, K. Ament, R.E. Ward, A.S. Choe, A.D. Barber, J.J. Pekar, S.H. Mostofsky, Intrinsic visual-motor synchrony correlates with social deficits in autism, Biol. Psychiatry 79 (8) (2016) 633–641.

[45] D. O'Neill, B. Xue, M. Zhang, Evolutionary neural architecture search for high-dimensional skip-connection structures on densenet style networks, IEEE Trans. Evol. Comput. 25 (6) (2021) 1118–1132, doi:10.1109/TEVC.2021.3083315.

[46] Y. Xue, P. Jiang, F. Neri, J. Liang, A multi-objective evolutionary approach based on graph-in-graph for neural architecture search of convolutional neural networks, Int. J. Neural Syst. 31 (9) (2021a), doi:10.1142/S0129065721500350.

[47] Y. Xue, Y. Wang, J. Liang, A. Sowik, A self-adaptive mutation neural architecture search algorithm based on blocks, IEEE Comput. Intell. Mag. 16 (2021b) 67–78.