Multi-View Brain Network Analysis with Cross-View Missing Network Generation

Gongxu Luo
Department of Machine Learning
MBZUAI
Abu Dhabi, United Arab Emirates
gongxu.luo@mbzuai.ac.ae

Chenyang Li
Dept. of Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, Canada
chenyang.li@uwaterloo.ca

Hejie Cui
Department of Computer Science
Emory University
Atlanta, GA, United States
hejie.cui@emory.edu

Lichao Sun
Dept. of Computer Science and Engineering
Lehigh University
Bethlehem, PA, United States
lis221@lehigh.edu

Lifang He
Dept. of Computer Science and Engineering
Lehigh University
Bethlehem, PA, United States
lih319@lehigh.edu

Carl Yang*
Department of Computer Science
Emory University
Atlanta, GA, United States
j.carlyang@emory.edu

***Abstract*—Parkinson's Disease (PD), one of the most common neurological disorders, has long been a challenge in public health clinical diagnosis as well as scientific understanding. Recently, there has been an upsurge of interest in brain network analysis which benefits the understanding of brain functions and early detection of neurological disorders extensively. Multi-view brain networks with different connectivity patterns among regions of interests (ROIs) can be constructed to reflect different and complementary perspectives of the brain connectivity profile. However, the extraction of such multi-view brain networks relies on the availability of multiple neuroimaging modalities and heavy data preprocessing, which often leads to severe missing data in either view. The cross-view missing issue hinders the pragmaticality of multi-view representation learning and downstream analysis. In this work, we formulate the novel problem of cross-view brain network generation and propose CroGen, a graph generative model that can generate the missing view when only one view is given. Specifically, GroGen leverages the potential correlation between diverse views of brain networks of the same individuals. Moreover, we design a pre-train schema to make full use of the labeled individuals with only single views of brain networks. Extensive experiments on real-life Parkinson's Progression Markers Initiative (PPMI) cohort demonstrate the supreme effectiveness of CroGen over baselines on both cross-view generation tasks and downstream PD classification.**

***Index Terms*—multi-view network analysis, cross-view network generation, brain network-based disease classification**

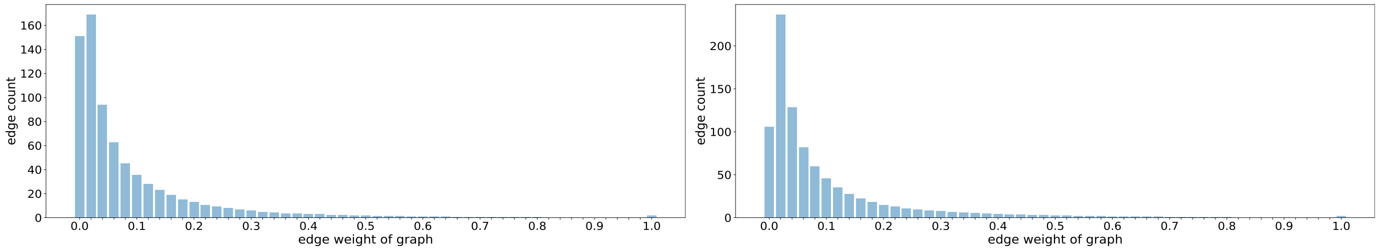
I. Introduction

According to the statistics of the World Health Organization (WHO), disabilities and deathes due to PD are increasing faster than other neurological diseases [1]. Brain is an organ with extremely complex nervous systems. Understanding the relationship between fundamental mechanisms of brain functions and diseases is beneficial for the early diagnosis of PD and other neurological disorders [2]–[5]. Therefore, brain network analysis has attracted increasing attention in both machine learning and neuroscience communities. Aided by the rapid advances in modern neuroimaging technology, e.g., functional Magnetic Resonance Imaging (fMRI), Diffusion Tensor Imaging (DTI), and CT scans, human brains can be represented by graph data, known as brain networks. specifically, nodes in brain networks denote the Region-of-Interests (ROIs), while the rich link structures model the connection strength among ROIs. Such brain networks generated from various neuroimaging modalities have provided invaluable insights into the underlying PD mechanism [6], [7].

In recent years, the appetite for data motivates biomedical researchers to explore new methods to utilize rich information through deep learning [8], [9]. Considering the incomplete information in single modalities of brain network data, multi-view representation learning methods come to rescue and have achieved great success in brain network analysis for PD by utilizing the mutual complement between different views [10]–[14]. However, despite the success of multi-view learning methods, collecting the multi-view networks is a non-trivial task, sometimes even impossible. Moreover, limitations such as poor quality of sample views and restricted access to specific equipments lead to a severe missing view problem, which impedes the pragmaticality of multi-view brain network analysis. This brings up a natural and challenging question: Can we learn to generate the missing view based on the given view to better support downstream analysis?

In this work, we formulate and study a novel problem of cross-view graph generation to deal with the missing view phenomenon in multi-view brain network analysis with a case study on PD. Specifically, we propose to learn the correlation patterns between the two views of structural brain networks based on a small set of individuals with both views available, so as to generate the missing views for individuals with only one view of data. To further leverage the single-view individuals, we design a schema to pre-train the model on single views of data, and then fine-tune the model on both views of data.

*Corresponding author



(a) The distribution of edge weights in V_1.

(b) The distribution of edge weights in V_2.

Fig. 1. The weight distribution of brain networks. The horizontal axis denotes the value range of the edge weights. The ordinate counts the number of edges whose weights fall within the interval.

The missing view scenario is first simulated by splitting the Parkinsons Progression Markers Initiative (PPMI) dataset with ratio 6:2:2, where 60% individuals have the complete pairs of two views while the other two 20% are single individuals with only one view. Next, different from the existing graph translation task [15]–[17], the goal of cross-view brain network generation is to generate the missing view that is a graph with a fixed node order and weighted edges conditioned on the given view. Moreover, as shown in Fig. 1, we further analyze the distribution of edge weights and find that the mean and variance of the normalized edge weights are dramatically small over both views in the dataset, which poses a unique challenge for graph generation task. In addition, the majority of weights are near the mean, which corresponds to the typically known *light-tailed distribution* [18], rendering higher requirements in the accuracy of the weighted graph generation model. Thanks to the surge of deep learning, many successful models, e.g., Variational graph auto-encoders (GVAE) [19], CondGen [20], CycleGAN [22], have been studied to model the complex dependencies and relationships in graph data. However, these graph generative models, which predict the existence of links, fail to generate the ideal weighted graphs with the unique brain network properties of *small-mean-small-variance* and *light-tailed distribution*.

Driven by such unique characteristics, we propose the novel model CroGen[1] to address the essential challenges of cross-view brain network generation. Specifically, CroGen is essentially a neural architecture of CycleGAN [22]. It fully leverages the well-developed graph convolutional network (GCN) [23] to perform node encoding and employs a deep translator for distribution transformation. When decoding, different from predicting the existence of the links directly, we treat cross-view graph generation as a set of regression tasks, and design an effective decoder with node concatenation operation and deep layers for information projection. The ability to learn an injective function guarantees the accuracy of generating small weights to conquer the essential challenges of cross-view brain network generation. The availability of a limited amount of paired views allows direct supervision of the cross-view generation process via a pair-wise loss. In addition, inspired by the semi-supervise methods for graph translation [15], we combine it with CroGen to make full use of single views. For example, when we adopt the 6:2:2 data split that includes 60% individuals with paired views as training set (with direct pairwise supervision), the other two 20% individuals with single views can be used for both (unsupervised) pre-training and testing towards the bidirectional cross-view generation process. This allows us to leverage 80% single views to train an autoencoder [25] to get a more robust embedding for the initialization of cross-view generation models. Finally, to fully demonstrate the value of cross-view graph generation and the power of our proposed CroGen model, we conduct a series of experiments on real-world brain networks from Parkinson's Progression Markers Initiative (PPMI) to evaluate the model against strong deep graph generative models properly adapted to the same setting. Additionally, downstream multi-view brain network analysis task for PD classification is evaluated in the missing view scenario to further verify the effectiveness of CroGen. The comparisons over various graph properties and careful visual inspections verify the supreme effectiveness of CroGen on cross-view brain network generation.

II. Related Work

A. Graph Generation

Graph generation task aims to model the distribution of the given graphs and to generate graphs with novel properties [26]. Due to the wide range of graph generation applications, e.g., traffic congestion prediction [27], novel drug discovery [28], [29], and protein structure modeling [30], the development of graph generation models has been well-established resulting in rich literature and popular models [31]. Recently, thanks to the development of deep learning, deep generative models become a mainstream method because of its capacity of modeling complex dependency in graphs [32]. Deep generative models can be divided into two categories, unconditional generative models and conditional ones. Unconditional generative models learn the distribution $P(G)$ based on the observed realistic graphs. Conditional generative models learn a conditional distribution $P(y|x)$ based on observed realistic graphs with corresponding auxiliary information. Graph translation is a branch of conditional generative model, which translates the graph from source domain to target domain [26]. In the past few years, different models are proposed for graph generation

[1] https://github.com/GongxuLuo/CroGen

with disparate types. Similar to image and text translation, an initial work has been done for graph translation [16]. They formulated an interesting problem of deep graph translation and proposed a novel Graph-Translation-Generative Adversarial Networks (GT-GAN) to handle the challenge of preserving the local property of graphs during generation. Based on this work, [33] stressed the importance of forecasting and synthesizing local events, and proposed Unpaired Graph Translation-Generative Adversarial Nets (UGT-GAN) for unpaired local event synthesize. It leveraged graph convolutional layers and deconvolutional layers for graph translation and extended the consistency loss proposed by CycleGAN [22] for unpaired translation. Similarly, to resolve the limitation of lacking high-quality paired dataset, [15] formulated the scenario of semi-supervised graph translation and proposed SegTran to cope with the semantic gap between the source graph and target graph by designing the translator for embedding transformation. Besides these canonical graph translation models, some important properties such as the topology infromation in graph learning [17], the hierarchical community structure of complex graph organizations [34], and complicated structure transformation of molecules in chemical reaction [35] etc., were also be considered for learning the dependency between graphs. Despite the great success in graph translation models, they mainly focus on predicting the existence of edges by classifier failing to generate the correct edge weights for brain network analysis with the intrinsic challenges of small-mean-small-variance and light-tailed distribution.

B. Deep Brain Network Analysis

Brain network analysis has became a hot research topic in recent years because of its vital role in disease diagnosis and understanding of the functional mechanism in brains [36]. Many effective neural network models have been studied to capture the complex dependencies of brain network recently. Considering the high risk of cognitive and neuromotor in preterm infants, the BrainNetCNN framework was proposed to predict cognitive and motor developmental outcome scores from structural brain networks of infants born preterm [37]. To explore how certain brain regions relate to a specific neurological disorder or cognitive stimuli, BrainGNN was designed to discover neurological biomarkers by leveraging topological and functional information of the functional magnetic resonance images (fMRI) [38]. IBGNN [39] proposes an interpretable framework to analyze disorder-specific ROIs and prominent connection. Brain Network Transformer [40] adapts the well-known transformer architecture specifically for brain networks. To explore the intrinsic mechanisms of PD, mutli-view GCN was proposed to fuse multiple modalities of brain images in PD classification [13]. Furthermore, Multi-view multi-graph embedding method was proposed to obtain a robust brain network embedding by exploiting complementary information from multiple neuroimaging modalities or views [11]. Similarly, due to the fact that most existing methods fail to accommodate the intrinsic structure, a novel multimodal brain network classification framework called Multiplex Graph Network (MGNet) was designed to extract latent structures of a set of multimodal brain networks by integrating tensor representation into multiplex GCNs [41]. Although multi-view learning methods are helpful for brain network analysis, the severe missing view problem is ignored, which becomes the main obstacle and hinders the progmaticality of these methods.

III. Methodology

A. Problem formulation

We focus on the novel problem of cross-view brain network generation, which relates graphs from multiple views with different link structures. Given a set of graphs $G = \{G_1, G_2.....G_n\}$, where $G_i = \{N_i^1, N_i^2, A_i^1, A_i^2\}$ represents a brain network described by two views with the same set of nodes N_i, and the adjacent matrixes A_i^1 and A_i^2. The superscribe 1 and 2 indicate the two views respectively, and the subscribe denotes the i-th graph. Note that the order of nodes is fixed and the same in both views [4]. A_i is a matrix holding the set of weights for the corresponding edges, where the weight A_{ij} indicates the connection strength between ROIs i and j in the brain. Apart from the limited number of paired views, the phenomenon of missing view results in abundant individuals with a single view in practice. We use V_1 and V_2 to denote the corresponding views of individuals. $V_1 - to - V_1$ and $V_2 - to - V_2$ represent the single view reconstruction tasks respectively. $V_1 - to - V_2$ and $V_2 - to - V_1$ indicate the bidirectional cross-view brain network generation tasks. In this work, we aim to explore and model the possible correlation patterns between different link structures. That is, by training model M on paired views, the corresponding missing view is generated based on the arbitrary given view.

By analyzing the distribution of the weighted brain networks, we summarize the essential challenges of cross-view brain network generation as follows:

1) *Weighted graph generation.* Compared with the traditional edge-based transformation which predicts the existence of edges, the brain networks with weighted edges lead to more complex node embedding space and dependencies between different link structures.
2) *Light-tailed distribution.* The link structures of brain networks are of small-mean-small-variance. Moreover, the weights obey a light-tailed distribution with extremely small values of mean and variation. Campaired to the simple binary classification task, this regression task has higher requirements on the expressive power and sensitivity of the generative models to generate small weights with fine-grained differences.

B. Model formulation

We propose CroGen to handle the above intrinsic challenges, which coherently joins the power of GCN, CycleGAN and the pre-training schema for cross-view brain network generation. Specifically, CroGen has two generators and two discriminators. The generator consists of an encoder that maps the input signal of the given view to the hidden representation

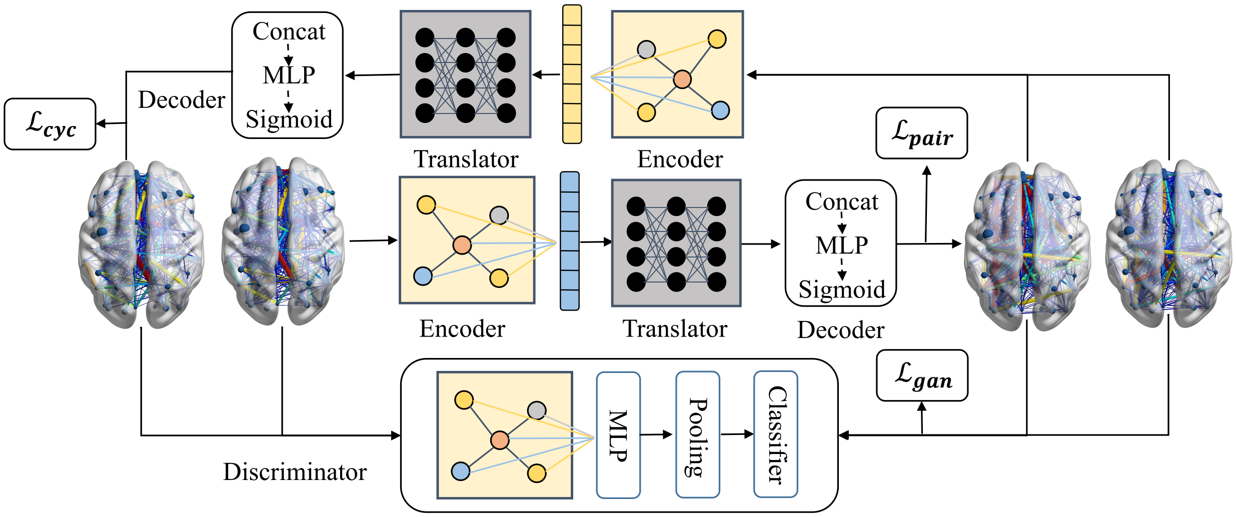


Fig. 2. The overview of CroGen.

space, a translator that transforms the distribution of node embedding from the given view to the corresponding missing view, and a decoder that constructs the link structures and generates the corresponding weights of edges in the missing view. The discriminator maps the observed signal to the specific latent representation space and obtains the graph-level embedding through the pooling operation. Different from the standard architecture of CycleGAN, the availability of paired views allows us to supervise the generation process directly, which mitigates the side efforts of quantifying the similarity of two probability distributions from samples known as two-sample test problem [24]. Fig. 2 illustrates the overall framework of CroGen. In the following, we will introduce the details of each module of CroGen.

a) Generator:

- Encoder: Following CondGen [20], nodes are firstly initialized by an one-hot embedding. Next, the spectral embedding method[2] is utilized for feature extraction based on the given adjacent matrix A, which results in the initial node features [21]. This allows us to get a more compact node embedding and reduce the memory consumption. After that, the two-layer GCN with nonlinear active functions and layer norm updates the input signals to get the node embedding as:

$$H^{(l)} = \sigma(g^{(l)}(f(A), A)), \tag{1}$$

where σ denotes the Leaky-ReLU activation function, $f(A)$ denotes the spectral embedding function for dimension reduction, and $g^{(l)}$ means the l-th layer of GCN.
- Translator: The goal of translator is to build the mapping function to explore the correlation patterns between the two views. This translation process bridges the distribution discrepency through transforming the node embedding of the given view to that of the corresponding missing view. The translator with stacked four-layer Multilayer Perceptions (MLPs) meets the requirements of the complex distribution transformation across two views of networks.
- Decoder: Different from the traditional graph generation task that predicts the existence of the edges, we treat the cross-view brain network generation task as a set of regression tasks. Inspired by the work on exploring the expressive power of Graph Neural Networks (GNNs), the concatenate operation is utilized to aggregate node information and enhance the expressive power of CroGen [42]. This is because compared with mean, sum and max operation for node embedding aggregation, this function preserves complete node information and can be learned to be injective. Next, considering the intrinsic characters of brain network, we append a four-layer fully connected feed-forward network with layernorm [43] and ReLU function to the concatenated embedding for information fusion. Followed by the sigmoid function, the computational process of edge weights is as follows:

$$w_{ij} = sigmoid(MLP({h_i}' \oplus {h_j}')), \tag{2}$$

where ${h_i}'$ and ${h_j}'$ represent the node embedding with distribution transformation, w_{ij} denotes the learned edge weight between node i and node j. Considering the light-tailed distribution and the data preprocessing, we choose the corresponding top K edge weights as the final results.

b) Discriminator: The discriminator tries to determine whether the input graph is real or generated. Given a graph, we get the node embedding through GCN. To perform graph-level classification, the mean pooling operation is utilized to get the graph-level embedding as follows:

$$X = Pool(MLP(g(f(A), A))), \tag{3}$$

where X indicates the graph-level embedding, $Pool$ is the mean pooling operation to get the graph-level embedding by

[2] https://scikit-learn.org/stable/modules/generated/sklearn.manifold.SpectralEmbedding.html

averaging all the node embeddings, and MLP is utilized to transform the representation from node-level to graph-level.

```
Algorithm 1: CroGen
Input: Trainset $G_1, G_2, \cdots, G_n$, testset with $V_1$,
       testset with $V_2$; Hyperparameter $\lambda_{pair}$, $\lambda_{cyc}$, $\beta$;
Output: The corresponding missing view $V_1$ in
        $V_2 - to - V_1$ task and $V_2$ in $V_1 - to - V_2$ task.
1 Extracting the single views from trainset and testset to
    construct the dataset for single view reconstruction.
2 Pre-tain the autoencoder for $V_1 - to - V_1$ and
    $V_2 - to - V_2$ tasks by using $\pounds_{pretrain}$ in Eq. (7);
3 Storing the parameters of the pre-train models and
    initialize the corresponding encoder, decoder and
    discriminator of CroGen separately;
4 Fine-tuning CroGen on trainset;
    while Not Converge do
        Receive a batch of data from Trainset
        Update the discriminator by $\pounds_{gan}$
        if epoch % 2 = 0 then
            Update the generator by $\pounds_{CroGen}$ through Eq. (5)
        end if
    end while
```

C. Training Algorithm

Paired views help overcome the two-sample test problem by directly supervising the generation process. However, there are lots of individuals with just a single view in practice. To make full use of the labeled individuals only with single views, we adopt the pre-training fine-tuning pipeline to leverage them through single view reconstruction for the initialization of CroGen, as illustrated in Algorithm 1. The pre-training schema joins the power of autoencoder [44] and Generative Adversarial Nets (GAN) [45]. For instance, in the $V_1 - to - V_1$ task, the node embedding of V_1 is firstly obtained by the encoder. Next, the decoder reconstructs the input view directly based on the obtained embedding. The encoder and decoder form the generator of GAN, which tries to generate a graph that can fool the discriminator. The discriminator is trained to differentiate the real and generated graphs. After training, the encoder is used to initialize that in $V_1 - to - V_2$ task. The decoder and discriminator initialize the corresponding one in the $V_2 - to - V_1$ task. Similarly, the $V_2 - to - V_2$ task is also used to initialize the decoder in $V_1 - to - V_2$ task, and the discriminator and the encoder in $V_2 - to - V_1$ task.

D. Training Details

We jointly train the generator and discriminator for both $V_1 - to - V_2$ and $V_2 - to - V_1$ by optimizing the loss function

$$\pounds_{CroGen} = \pounds_{gan} + \lambda_{pair}\pounds_{pair} + \lambda_{cyc}\pounds_{cyc}, \qquad (4)$$

where $\pounds_{CroGen}$ is the loss function for CroGen, $\pounds_{gan}$ is the adversarial loss to make sure that the graph generated by the generator can fool the discriminator and the discriminator can distinguish the ground truth and the generated graphs. The pair loss $\pounds_{pair} = MSE(\hat{G}, G)$ guarantees the cross-view generation ability. $\pounds_{cyc}$ denotes the cycle consistency loss. λ_{pair} and λ_{cyc} are tunable hyperparameters to control the weights of $\pounds_{pair}$ and $\pounds_{cyc}$. The updating rules in each batch are as follows:

$$\begin{aligned} \theta_G &\overset{+}{\leftarrow} -\nabla_G(\pounds_{gan} + \lambda_{pair}\pounds_{pair} + \lambda_{cyc}\pounds_{cyc}), \\ \theta_D &\overset{+}{\leftarrow} -\nabla_D\pounds_{gan}. \end{aligned} \qquad (5)$$

In the pre-training stage, the loss function of $V_1 - to - V_1$ and $V_2 - to - V_2$ tasks are formulated as:

$$\pounds_{pretrain} = \pounds_{gan} + \beta\pounds_{rec}, \qquad (6)$$

where the $\pounds_{gan}$ is same as that in Eq. (4), $\pounds_{rec}$ is the loss function for single view reconstruction, and β indicates the weight of the reconstruction loss. Similarly, the updating rules are as follows:

$$\begin{aligned} \theta_G &\overset{+}{\leftarrow} -\nabla_G(\pounds_{gan} + \beta\pounds_{rec}), \\ \theta_D &\overset{+}{\leftarrow} -\nabla_D\pounds_{gan}. \end{aligned} \qquad (7)$$

IV. Experiment

In this section, we conduct experiments to evaluate the effectiveness of CroGen for missing view brain network generation. Moreover, the performance is further verified by the downstream PD classification task.

A. Dataset and preprocessing

The dataset we used is Parkinson's Progression Markers Initiative (PPMI), which contains 718 subjects, where 569 subjects suffer from PD, and the rest 149 subjects are Healthy Control (HC) ones. There are two modalities in PPMI: link structures constructed by Probabilistic Index of Connectivity (PICo) [46] and Hough Voting [47] with size of 84 × 84. Nodes indicate brain ROIs, while edges indicate the connection strength among ROIs. The means of edge weights in two views are approximately 0.010 and 0.014, which follow the typical light-tailed distribution. To reduce weak connections between different brain regions, we process the data by setting the weight less than the mean to zero.

B. Baselines

Since no baselines are available for the novel task of cross-view brain network generation, we carefully adapt four up-to-date strong graph generative models as baselines, i.e., Conditional Generative adversarial nets (C-GAN) [48], CondGen [20], Cycle-GAN [22], Variational Auto-encoding Generative Adversarial Networks (VAEGAN) [49]. To ensure a fair comparison, the generator and discriminator are kept in the same setting and design with CroGen.

C. Protocols

For the 6:2:2 setting, in the testing stage, V_1 is seen as the single view in $V_1 - to - V_2$ task, V_2 is the corresponding

$V_1 - to - V_2$			
Model	Degree	Clustering	MSE
C-GAN	0.1397 ± 0.0183	0.7450 ± 0.1438	0.0571 ± 0.0005
Cycle-GAN	0.1631 ± 0.0124	1.2933 ± 0.1286	0.0683 ± 0.0054
CondGen	0.2009 ± 0.0084	0.7332 ± 0.3635	0.0850 ± 0.0049
VAEGAN	0.1959 ± 0.0058	0.7849 ± 0.3451	0.0791 ± 0.0239
Ablation	0.1000 ± 0.0165	0.3944 ± 0.1167	0.0580 ± 0.0005
CroGen	0.0779 ± 0.0444	0.0344 ± 0.0304	0.0081 ± 0.0003
Pre-train+CroGen	**0.0608 ± 0.0291**	**0.0208 ± 0.0158**	**0.0081 ± 0.0002**
$V_2 - to - V_1$			
C-GAN	0.2127 ± 0.0057	1.0191 ± 0.1819	0.0690 ± 0.0004
Cycle-GAN	0.2465 ± 0.0154	1.5124 ± 0.0603	0.0830 ± 0.0040
CondGen	0.2622 ± 0.0150	0.9567 ± 0.2951	0.0903 ± 0.0046
VAEGAN	0.2669 ± 0.0165	0.9561 ± 0.2216	0.0821 ± 0.0226
Ablation	0.2138 ± 0.0112	0.7681 ± 0.1485	0.0703 ± 0.0005
CroGen	0.1699 ± 0.0942	0.1342 ± 0.1604	0.0088 ± 0.0003
Pre-train+CroGen	**0.1188 ± 0.0595**	**0.1015 ± 0.0589**	**0.0088 ± 0.0001**

TABLE I
PERFORMANCE ON BIDIRECTIONAL CROSS-VIEW BRAIN NETWORK GENERATION.

Model	V_1	V_2	V_1 + GenV_2	GenV_1 + V_2	$V_1 + V_2$
GCN	0.723 ± 0.069	0.716 ± 0.064	0.746 ± 0.049	0.755 ± 0.039	**0.756 ± 0.046**
GAT	0.695 ± 0.041	0.676 ± 0.050	0.752 ± 0.022	**0.754 ± 0.020**	0.751 ± 0.019
GIN	0.662 ± 0.057	0.672 ± 0.045	**0.675 ± 0.061**	0.670 ± 0.060	0.670 ± 0.034
BrainNN	0.792 ± 0.030	0.743 ± 0.066	0.791 ± 0.015	**0.793 ± 0.016**	0.793 ± 0.027

TABLE II
PERFORMANCE ON DOWNSTREAM GRAPH CLASSIFICATION TASK, *Gen* DENOTES THE CORRESPONDING MISSING VIEW GENERATED BY CROGEN.

missing view. Similarly, V_2 is the single view and V_1 is the missing view in the $V_2 - to - V_1$ task. In the pre-training stage, the V_1 in the trainset and the testset for $V_1 - to - V_2$ task is extracted for the $V_1 - to - V_1$ task. In the same way, the V_2 in the trainset and the testset for $V_2 - to - V_1$ task is obtained for the $V_2 - to - V_2$ task. For evaluation, following the Maximum Mean Discrepancy (MMD) [50], [51] measures to evaluate the performance of generative models, we use the degree distribution and clustering coefficient distribution to evaluate the quality of the generated sample by measuring the distance of the corresponding distribution between the generated graph and ground truth [28]. In addition, the Mean Square Error (MSE) loss is utilized to measure the weight difference between the generated graph and the ground truth.

D. Settings

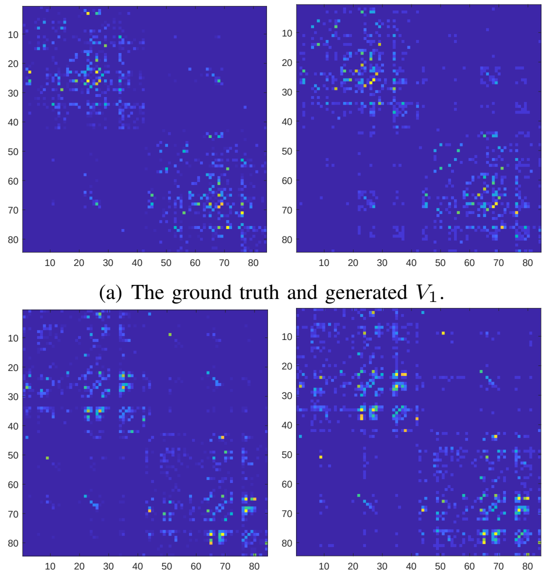
In the cross-view brain network generation task, the learning rate of the generator of CroGen is 0.00006 and 0.001 for the discriminator. The exponential decay rates for the first and the second moment estimates are 0.9 and 0.999 respectively. The dimension of node embedding after reduction is 32, the dimension of hidden node embedding is 64, and the dimension of graph embedding is 16. λ_{pair} and λ_{cyc} determine the weights of pair and cycle loss. We set λ_{pair} to 15 and λ_{cyc} to 5. In the pre-training stage, β, which determines the wight of reconstruction loss, is set to 15. To guarantee the significance of the experimental results, we conduct every experiment five times for all tasks and get the final results by averaging them.

E. Experimental results and analysis

1) Performance on cross-view brain network generation: Experimental results on the PPMI dataset are shown in Table I to evaluate the effectiveness of CroGen. From Table I we can see that compared with strong baselines, CroGen gets the best results on all evaluation methods and the variance is smaller in Clustering coefficient and MSE. What is more, the quality of the missing view generated by CroGen has an order of magnitude improvement in the evaluation metrics. This is because the translator with a deeper network fits the complex dependency better, and the joint loss function provides strong supervised objection to guide the weighted graph generation of the missing view. Moreover, the decoder composed by the concatenation operation and MLP layers is more powerful. The concatenation operation guarantees the expressive power for node information aggregation. The four-layer MLP and sigmoid function compose a strong regressor, which satisfies the requirements of weighted edge generation. Therefore, compared with dot product function, this decoder is more sensitive and powerful to handle the intrinsic challenges of small-mean-small-variance and light-tailed distribution. To test whether CroGen can effectively generate the missing view, we visualize the connection map between ROIs of ground-truth graph (left) and the generated one (right) in both views as shown in Fig 3. From Fig 3, we can see that the distribution of generated views consist with the ground truth. To verify the effectiveness of the decoder, ablation study is conducted by replacing the decoder with dot production (DP). Experimental results in Table I

$V_1 - to - V_2$			
λ_{pair}-λ_{cyc}	Degree	Clustering	MSE
5-5	0.1045 ± 0.0519	0.0621 ± 0.0950	0.0082 ± 0.0003
10-5	0.0715 ± 0.0351	0.0215 ± 0.0179	0.0082 ± 0.0003
15-5	**0.0608** ± **0.0291**	**0.0208** ± **0.0158**	**0.0081** ± **0.0002**
20-5	0.0675 ± 0.0406	0.0287 ± 0.0114	0.0082 ± 0.0002
15-10	0.0984 ± 0.0517	0.0533 ± 0.0474	0.0082 ± 0.0003
15-15	0.0944 ± 0.0625	0.0600 ± 0.0709	0.0081 ± 0.0003
15-20	0.0900 ± 0.0550	0.0376 ± 0.0421	**0.0081** ± **0.0002**
$V_2 - to - V_1$			
5-5	0.1428 ± 0.0743	0.0896 ± 0.0847	0.0090 ± 0.0003
10-5	0.1331 ± 0.0788	0.0913 ± 0.0556	0.0088 ± 0.0003
15-5	**0.1188** ± **0.0595**	0.1015 ± 0.0589	**0.0088** ± **0.0001**
20-5	0.1194 ± 0.0619	**0.0584** ± **0.0287**	0.0089 ± 0.0004
15-10	0.1738 ± 0.0958	0.1289 ± 0.0951	0.0090 ± 0.0004
15-15	0.1210 ± 0.0800	0.0983 ± 0.0788	0.0089 ± 0.0003
15-20	0.1669 ± 0.0708	0.1129 ± 0.0656	0.0088 ± 0.0002

TABLE III
HYPERPARAMETER ANALYSIS.



(a) The ground truth and generated V_1.

(b) The ground truth and generated V_2.

Fig. 3. visualization of the connectivity map between ROIs in multi-view brain networks.

show a dramatic decrease in the degree distribution, clustering coefficient, and MSE loss evaluation metrics. This illustrates the superiority of the decoder of CroGen over the dot product on the cross-view graph generation.

Table I shows that CroGen initialized by the pre-taining models demonstrates better performance compared with random initialization. The reason is that the single views provide more prior knowledge to assist the cross-view brain network generation. To be specific, we can use 80% individuals with single view in the reconstruction tasks. Therefore, the encoder, decoder, and discriminator obtain more information than random initialization, leading to a better performance directly.

2) Performance on PD classification: To further verify the effectiveness of CroGen and the quality of the generated missing views, we conduct experiments on up-to-date models for the downstream PD classification task, including GCN [23], GAT [52], GIN [42] and BrainGNN [38]. Specifically, these models are trained with five data settings, i.e. V_1, V_2, V_1 + corresponding generated V_2 by CroGen, V_2 + corresponding generated V_1 by CroGen, and V_1 + V_2. Experimental results are reported in Table II. It is shown that the up-to-date models trained on completed datasets i.e. V_1 with corresponding generated V_2 and V_2 with corresponding generated V_1 get better performance compared to single view datasets. Besides, the performances are rather close to that on full ground truth V_1 + V_2 dataset and sometimes exceed.

F. Hyperparameter analysis

There are two important hyperparameters λ_{pair} and λ_{cyc} in CroGen to tune the weights of pair loss $\pounds_{pair}$ and cycle consistency loss $\pounds_{cyc}$ for cross-view brain network generation. $\pounds_{pair}$ supervises the generation process directly. $\pounds_{cyc}$ further considers the cycle consistency. We compare the performance of CroGen under different λ_{pair} and λ_{cyc} values to analyze the importance of the two losses. Experimental results of hyperparameter analysis are shown in Table III. It shows that $\pounds_{pair}$ is often more important than λ_{cyc}. Since the cross-view generation is our final goal, it is reasonable to pay more attention on it. Overall, CroGen present a relatively stable performance unless λ_{pair} and λ_{cyc} are set to extreme values like 5-5 or 15-20, which has a relative small proportion for the pair loss. With proper weights such as 10-5, 15-5 and 20-5, CroGen obtains better performance and smaller variance compare with other baselines.

V. CONCLUSION

To the best of our knowledge, this is the first research effort towards the novel but important problem of cross-view brain network generation considering the missing view phenomenon. To address the intrinsic challenges of cross-view brain network generation, we design CroGen to coherently join the power of GCN, CycleGan and pre-train schema. In practice, we conduct experiments on real-life Parkinson's Progression Markers Initiative (PPMI) data to validate the effectiveness of CroGen in comparison with several up-to-date generative models adapted to our missing view brain network generation settings. Moreover, the quality of the generated view is validated concretely on downstream Parkinson's Disease (PD) classification task.

Acknowledgment

This research was partially supported by the University Research Committee of Emory University and the internal funds of the Computer Science Department of Emory University.

References

[1] World Health Organization. "Parkinson disease: a public health approach: technical brief." 2022.

[2] F. Alex, A. Zalesky, and E. Bullmore. "Fundamentals of brain network analysis." Academic Press, 2016.

[3] M.A. Myszczynska, et al. "Applications of machine learning to diagnosis and treatment of neurodegenerative diseases." Nature Reviews Neurology 16, 2020, 440-456.

[4] H. Cui, W. Dai, Y. Zhu, et al. "BrainGB: A Benchmark for Brain Network Analysis with Graph Neural Networks." IEEE Trans. Med. Imaging, 2022.

[5] Y. Yu, X. Kan, H. Cui, et al. "Learning Task-Aware Effective Brain Connectivity for fMRI Analysis with Graph Neural Networks." arXiv preprint arXiv:2211.00261, 2022.

[6] M. Politis. "Neuroimaging in Parkinson disease: from research setting to clinical practice." Nature Reviews Neurology 10, 2014, 708-722.

[7] X. Kan, H. Cui, J. Lukemire, Y. Guo, C. Yang. "Fbnetgen: Task-aware gnn-based fmri analysis via functional brain network generation." MIDL, 2022.

[8] Lee, Jinhyuk, et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining." Bioinformatics 36, 2020, 1234-1240.

[9] R. Xu, et al. "Counterfactual and Factual Reasoning over Hypergraphs for Interpretable Clinical Predictions on EHR." ML4H, 2022.

[10] Y. Yu, Y. Li, J. Shen, et al. "Steam: Self-supervised taxonomy expansion with mini-paths." SIGKDD, 2020.

[11] Y. L, L. He, B. Cao, et al. "Multi-view multi-graph embedding for brain network clustering analysis." AAAI, 2018.

[12] Y. Zhu, H. Cui, Li. He, Li. Sun, C. Yang. "Joint embedding of structural and functional brain networks with graph neural networks for mental illness diagnosis." EMBC, 2022.

[13] X Zhang, L He, K Chen, Y Luo, J Zhou and F Wang. "Multi-view graph convolutional network and its applications on neuroimage analysis for parkinsons disease." AMIA, 2018.

[14] Y. Yang, Y. Zhu, H. Cui, et al. "Data-Efficient Brain Connectome Analysis via Multi-Task Meta-Learning." SIGKDD, 2022.

[15] T. Zhao, X. Tang, X. Zhang and S. Wang. "Semi-Supervised Graph-to-Graph Translation." CIKM, 2020.

[16] X. Guo, L. Wu, and L. Zhao. "Deep graph translation." IEEE TNNLS, 2022.

[17] M. Sun, and P. Li. "Graph to graph: a topology aware approach for graph structures learning and generation." AISTATS, 2019.

[18] R. R. Wilcox, Applying contemporary statistical techniques. Elsevier, 2003.

[19] T. N Kipf and M. Welling. "Variational graph auto-encoders." NeurIPS Workshop on Bayesian Deep Learning, 2016.

[20] C. Yang, P. Zhuang, W. Shi, A.n Luu and P. Li "Conditional structure generation through graph variational generative adversarial nets." NeurIPS, 2019.

[21] H. Cui, Z. Lu, P. Li, C. Yang. "On positional and structural node features for graph neural networks on non-attributed graphs." CIKM, 2022.

[22] J.Y. Zhu, T. Park, P. Isola and A. A. Efros . "Unpaired image-to-image translation using cycle-consistent adversarial networks." ICCV, 2017.

[23] W. Max, and T. N. Kipf. "Semi-supervised classification with graph convolutional networks." ICLR, 2017.

[24] R. Lars, and E. Haber. "An introduction to deep generative modeling." GAMMMitteilungen 44.2 , 2021),e202100008.

[25] C. Liou, W. Cheng, J. Liou and D. Liou. "Autoencoder for words." Neurocomputing 139, 2014, 84-96.

[26] X. Guo, and L. Zhao. "A systematic survey on deep generative models for graph generation." arXiv preprint arXiv:2007.06686, 2020.

[27] Y. Zhang, S. Wang, B. Chen, J. Cao, and Z. Huang. "Trafficgan: Network-scale deep traffic prediction with generative adversarial nets." IEEE TITS 22, 2019, 219-230.

[28] J. You, R. Xing, X. Ren, et al. "Graphrnn: Generating realistic graphs with deep auto-regressive models." ICML, 2018.

[29] M. Xu, L. Yu, Y. Song, C. Shi, S. Ermon, and J. Tang. "Geodiff: A geometric diffusion model for molecular conformation generation." arXiv preprint arXiv:2203.02923, 2022.

[30] N. Anand, and P. Huang. "Generative modeling for protein structures." NeurIPS, 2018.

[31] B. Katy, S. Sanyal, and A. Vespignani. "Network science." Annu. rev. inf. sci. technol. 41.1, 2007, 537-607.

[32] B. T. Sam, L. Adam, L. Yang, G. W. Chris. "Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models." arXiv preprint arXiv:2103.04922, 2021.

[33] Y. Gao, X. Guo, and L. Zhao. "Local event forecasting and synthesis using unpaired deep graph translations." SIGSPATIAL Workshop on Analytics for Local Events and News. 2018.

[34] D. Zhou, L. Zheng, J. Xu and J. He. "Misc-GAN: A multi-scale generative model for graphs." Frontiers in big Data 2, 2019, 3.

[35] K. Do, T. Tran, and S. Venkatesh. "Graph transformation policy network for chemical reaction prediction." SIGKDD, 2019.

[36] A. Fornito, A. Zalesky, and E. Bullmore. "Fundamentals of brain network analysis." Academic Press, 2016.

[37] Kawahara, Jeremy, et al. "BrainNetCNN: Convolutional neural networks for brain networks; towards predicting neurodevelopment." NeuroImage 146, 2017, 1038-1049.

[38] X. Li, Y. Zhou, N. Dvornek, M. Zhang, S. Gao, J. Zhuang and J.S. Duncan. "Braingnn: Interpretable brain graph neural network for fmri analysis." Med Image Anal 74, 2021, 102233.

[39] H. Cui, W. Dai, Y. Zhu, et al. "Interpretable graph neural networks for connectome-based brain disorder analysis." MICCAI, 2022.

[40] X. Kan, W. Dai, H. Cui, et al. "Brain network transformer." NeurIPS, 2022.

[41] Z. Kong, L. Sun, H. Peng, L. Zhan, Y. Chen and L. He. "Multiplex graph networks for multimodal brain network analysis." arXiv preprint arXiv:2108.00158, 2021.

[42] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. "How powerful are graph neural networks?." ICLR, 2019.

[43] J. L. Ba, J. R. Kiros, and G. E. Hinton. "Layer normalization." arXiv preprint arXiv:1607.06450, 2016.

[44] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks." AIChE journal 37, 1991, 233-243.

[45] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. "Generative adversarial nets." NeurIPS, 2014.

[46] G. J.M. Parker, H. A. Haroon, and C. AM WheelerKingshott. "A framework for a streamlinebased probabilistic index of connectivity (PICo) using a structural interpretation of MRI diffusion measurements." Journal of Magnetic Resonance Imaging: Magn Reson Med 18, 2003, 242-254.

[47] I. Aganj, C. Lenglet, N. Jahanshad, E. Yacoub, N. Harel, P. M. Thompson, and G. Sapiro. "A Hough transform global probabilistic approach to multiple-subject diffusion MRI tractography." Med Image Anal 15, 2011, 414-425.

[48] M. Mirza, and Simon Osindero. "Conditional generative adversarial nets." arXiv preprint arXiv:1411.1784, 2014.

[49] A.B.L. Larsen, S.K. Snderby, H. Larochelle and O. Winther. "Autoencoding beyond pixels using a learned similarity metric." ICML, 2016.

[50] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Scholkopf and A. Smola. "A kernel two-sample test." JMLR 13, 2012, 723-773.

[51] Y. Yu, et al. "Cold-Start Data Selection for Few-shot Language Model Fine-tuning: A Prompt-Based Uncertainty Propagation Approach." arXiv preprint arXiv:2209.06995, 2022.

[52] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Li'o and Y. Bengio. "Graph attention networks." ICLR, 2017.