Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for a degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter now, including display on the World Wide Web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.

Signature:

Yi (Owen) Yang April 3, 2023

Pre-Training Graph Neural Networks for Data-Efficient Brain Network Analysis

By

Yi (Owen) Yang

Carl Yang, Ph.D.
Advisor

Computer Science

Carl Yang, Ph.D.
Advisor

Lars Ruthotto, Ph.D.
Committee Member

Li Xiong, Ph.D.
Committee Member

2023

Pre-Training Graph Neural Networks for Data-Efficient Brain Network Analysis

By

Yi (Owen) Yang

Carl Yang, Ph.D.
Advisor

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences of
Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Computer Science

2023

Abstract

Pre-Training Graph Neural Networks for Data-Efficient Brain Network Analysis
By Yi (Owen) Yang

The human brain is the central hub of the neurobiological system, controlling behavior and cognition in complex ways. Recent advances in neuroscience and neuroimaging analysis have shown a growing interest in the interactions between brain regions of interest (ROIs) and their impact on neural development and disorder diagnosis. As a powerful deep model for analyzing structural data, Graph Neural Networks (GNNs) have been applied for brain network analysis. However, effective training of deep models requires large amounts of labeled data, which is often scarce in brain network datasets due to the complexities of data acquisition and sharing restrictions. To make the most out of available training data, this work examines data- and label-efficient training of GNN model. In particular, the goal is to pre-train GNN to capture intrinsic brain network structures, regardless of clinical outcomes, and is easily adaptable to various downstream tasks. To this end, the proposed framework comprises three key components: (1) a meta-learning based multi-task pre-training platform with dynamic task adaptive reweighing consideration that learns a generalizable model initialization with efficient optimization schedule (2) an unsupervised pre-training objective designed specifically for brain networks, which enables learning from large-scale datasets without task-specific labels; (3) a data-driven atlas mapping pipeline with variance-based ROI alignment mechanism that facilitates knowledge transfer across datasets with different ROI systems. Extensive empirical evaluations using various GNN backbones have demonstrated the robust and superior performance of the proposed framework compared to baseline methods.

Pre-Training Graph Neural Networks for Data-Efficient Brain Network Analysis

By

Yi (Owen) Yang

Carl Yang, Ph.D.
Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Computer Science

2023

Acknowledgments

I would like to acknowledge and thank my advisor, Dr. Carl Yang, for all his help and support throughout this project and beyond. The project would not be completed without his valuable advice, insights, and thoughts. I cherish every discussions we had over the past year, and Dr. Yang is constantly helping me grow and flourish both academically and as a person. I also want to leverage this opportunity to thank my committee members, Dr. Li Xiong and Dr. Lars Ruthotto, for their guidance, kindness, and support in my academic journey.

I also want to thank my collaborators, Mr. Yanqiao Zhu from UCLA and Ms. Hejie Cui from Emory, for their contribution in this project. I still remember our fruitful discussions during the brainstorm phase and your immense support at urgent times.

Lastly, I want to thank my friends and family for your eternal love and trust. Thank you for always being there for me during happy times and during tough times.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Background and Motivation

It has long been an enticing pursuit for neuroscience researchers and mental disorder clinicians to understand the functions and structures of human brains, which are known to be related to many complicated diseases, including bipolar disorder (BP), immunodeficiency virus infection (HIV), and Parkinson's disease (PPMI) [96] which this study will mainly focus on. In the last decade, the development of neuroimaging techniques, such as magnetic resonance imaging (MRI), functional MRI, diffusion tensor imaging (DTI), etc., provides an important source of information that facilitates the diagnosis of various brain diseases. Based on neuroimaging data, one can build brain networks that encode brain anatomical regions as nodes and their connections as edges. This kind of data representation characterizes the complex connections among different regions of interest (ROI). Effective brain network analysis plays a pivotal role in understanding the biological structures and functions of complex neural systems, which potentially helps the early diagnosis of neurological disorders and facilitates neuroscience research [61, 88, 53].

Graph Neural Networks (GNNs) have emerged as a powerful tool for analyzing graph-structured data, delivering impressive results on a wide range of network datasets, including social networks, recommender systems, knowledge graphs, protein and gene networks, and molecules, among others [47, 28, 73, 79, 87, 92, 95, 54, 86]. These models have proven their ability to learn powerful representations and efficiently compute complex graph structures, making them ill-suited for various downstream tasks. In the field of neuroscience, GNN has been applied to brain network analysis, specifically for graph-level classification/regression [92, 87, 21] and important vertex/edge identification [91, 57, 83], towards tasks such as connectome-based disease prediction and multi-level neural pattern discovery. However, deep learning models, including GNNs, require large amounts of labeled data to achieve optimal performance [38, 93, 99]. While neuroimaging datasets are available from national neuroimaging studies such as the ABCD [10], ADNI [32], and PPMI [2], these datasets are still relatively small compared to graph datasets from other domains, such as datasets with 41K to 452K graphs on OGB [39] and datasets with thousands to millions of graphs on NetRepo [71]). The limited amount of data can result in GNNs having difficulty in learning informative knowledge and easily overfitting the data distribution.

Recently, to improve data efficiency, the framework of transfer learning has attracted a lot of attention in many application domains, which allows a model pre-trained on large-scale source datasets to be adapted to smaller target datasets while maintaining robust performance. However, the success of transfer learning depends on the availability of similar supervision labels on the source and target dataset. This is not always feasible in large-scale public studies, particularly in the field of brain network analysis. One other major challenge is the inconsistent ROI parcellation systems in constructing different brain network datasets, which hinders the transferability of pre-trained models across datasets. The process of parcellating raw imaging data into

brain networks is highly complex and usually done ad hoc by domain experts for each study, making it unrealistic to expect every institution to follow the same parcellation system. Although some institutions may release preconstructed brain network datasets [20], the requirement for universal adherence to a single parcellation system is infeasible.

1.2 Overview of Proposed Solutions

To tackle these challenges, this work aims to explore meta-learning techniques and self-supervised pre-training for GNNs. The framework of meta-learning, also known as learning to learn, aims at learning over multiple, seemingly diverse tasks during the pre-training phase, with the goal of deriving a generalized initialization of the model such that it can be adapted to any arbitrary unseen tasks with efficient convergence. One of the advantages of this approach is that the model can be trained by multiple tasks simultaneously. Considering the multimodal nature of the brain network datasets where multiple interrelated types of connections exist among ROIs (e.g., structural and functional connections), we propose to leverage every such modality as one training task and meta-train the models using multiple training tasks. With sufficient amount of meta-training, we have a pre-trained model that simultaneously performs well on all these training tasks, which we believe to be generic and easily transferable to new target tasks.

On the other hand, self-supervised pre-training has been shown to be effective in various domains, such as computer vision [29, 12], natural language processing [18, 69], and graph mining [76]. This work also aims to explore a self-supervised pre-training approach for GNNs on brain networks that is not restricted by task-specific supervision labels. Despite the promising potential, unique challenges still need to be

addressed to achieve effective disease prediction. In particular, this work proposes a novel two-level contrastive learning strategy based on the naturally aligned node systems of brain networks across individuals.

Based on the meta-learning framework and the contrastive self-supervised training strategy, this work further improves the model with brain-network-oriented designs. At first, the datasets used for training and testing usually use different ROI atlas mappings for constructing the brain networks, resulting in different numbers and physical regions of nodes, which hinders the transferability of GNNs. To mitigate this discrepancy, this work proposes to leverage a linear autoencoder model that transforms the original features into low-dimensional representations in a uniform embedding space and aligns them using variance-based projection, which incorporates regularizations that preserve spatial relationships, consider neural modules, and promote sparsity. Secondly, in the meta-training phase, different training tasks may contribute differently to the learning of generic and transferable knowledge which may limit the generalization performance. This work motivates the design consideration by visualizing the relative contribution of the source tasks towards the learning on the target task, where the data-driven observation corroborates with existing clinical research. Based on this findings, this work then proposes an adaptive task reweighing scheme to dynamically adjust the learning rate and weight decay parameters according to the contribution of each meta-training task. Extensive experiments and ablation studies conducted on real-world brain network datasets verify the effectiveness of these proposed strategies.

1.3 Summary of Contribution

In summary, the contribution of this work is four-folded:

- This work is the first to highlight the inherent challenge of limited training samples for learning with brain network data. This work formulates this problem into a data-efficient learning objective with the goal of pre-training the model to a generalizable initialization that can effectively adapt to unseen downstream objectives.

- This work proposes to leverage meta-learning strategies to pre-train a given model on available source tasks. In addition, the pre-training process is powered by a novel two-level contrastive sampling strategy that considers special properties of brain network data.

- This work also addresses unique challenges in multi-dataset and cross-dataset learning on brain networks by proposing brain-network-specific design components featured by a linear autoencoder network with customized regularizations, a dynamic task reweighing mechanism for multi-task pre-training, and a variance-based sorting algorithm to promote ROI alignment after dataset-specific atlas tranformation.

- This work also conducts extensive experiments to benchmark the working effectiveness of the proposed framework against a multitude of state-of-the-art methods adapted to our setting. In addition, the work also investigates the contribution of each constituent parts of the framework through series of ablation studies.

Chapter 2

Related Work

2.1 GNNs for Brain Network Analysis.

In recent years, graph neural networks (GNNs) have attracted broad interest due to their established power for analyzing graph-structured data [80, 87, 48]. Several pioneering deep models have been devised to predict brain diseases by learning the graph structures of brain networks. For example, BrainGNN [52] proposes ROI-aware graph convolutional layers and ROI-selection pooling layers for predicting neurological biomarkers. BrainNetCNN [44] designs a CNN that includes edge-to-edge, edge-to-node, and node-to-graph convolutional filters, leveraging the topological locality of brain connectome structures. BrainNetTF [43] introduces a transformer architecture with an orthonormal clustering readout function that considers ROI similarity within functional modules. Additionally, various studies [15, 42, 100, 14] have shown that, when data is sufficient, GNNs can greatly improve performance in tasks such as disease prediction. However, in reality, the lack of training data is a common issue in neuroscience research, particularly for specific domains and clinical tasks. Despite this, there has been little research into the ability of GNNs to effectively train for brain network analysis when data is limited.

2.2 Meta-Learning for Graph Classification

Recently, meta-learning has drawn significant attention in the machine learning community since it is able to address the problem of limited training data. There are also several attempts of meta-learning for GNN-based graph classification. For example, [11] recognize unseen classes with limited labeled graph samples using meta-training. [7] attempt to develop a general framework that can adapt to three-level tasks — graph classification, node classification, and link prediction with meta-learning, but without considering the unique characteristics of brain networks. [59] use the shared sub-structures between training classes and test classes to design a better meta-learning framework. However, none of the shared sub-structures can be utilized since brain networks are complete graphs. Meta-MGNN [27] proposes a self-supervised learning objective that predicts atom types for molecular datasets. However, there is no precise label for each node for prediction in brain networks.

2.3 Unsupervised Graph Representation Learning and GNN Pre-training.

Unsupervised learning is a widely used technique for training complex models when resources are limited. Recent advancements in contrastive learning [13, 29] have led to various techniques for graphs. For instance, GBT [5] designs a Barlow Twins [94] loss function based on the empirical cross-correlation of node representations learned from two different views of the graph [97]. Similarly, GraphCL [93] involves a comparison of graph-level representations obtained from two different augmentations of the same graph. DGI [82] contrasts graph and node representations learned from the original graph and its corruption.

To obtain strong models for particular downstream tasks, unsupervised training techniques can be used to pre-train a model, which is then fined tuned on the downstream tasks to reduce the dependence on labeled training data. The approach has proven highly successful in computer vision [9, 24], natural language processing [19, 69, 68], and multi-modality (e.g. text-image pair) learning [49, 90]. There are various strategies for pre-training GNNs as well. GPT-GNN [40] proposes graph-oriented pretext tasks, such as masked attribute and edge reconstruction. L2P-GNN [56] introduces dual adaptation by simultaneously optimizing the encoder on a node-level link prediction objective and a graph-level self-supervision task similar to DGI. Others, such as GMPT [36] adopt an inter-graph message-passing approach to obtain context-aware node embedding and optimize the model concurrently under supervision and self-supervision. To the best of our knowledge, the effectiveness of both contrastive learning and pre-training has not been investigated in the context of the unique properties of brain networks.

Chapter 3

Problem Definition

This work considers the problem of disease prediction with multiple brain network datasets. Formally, given a dataset for one specific disease $\mathcal{D} = \{\mathcal{G}_i\}_{i=1}^{N}$ containing N subjects, where $\mathcal{G}_i$ represents the i^{th} brain network instance. Each brain network object can be considered as an edge weighted graph $\mathcal{G}_i = (\mathcal{V}, \mathcal{E}_i, \boldsymbol{A}_i)$, where $\mathcal{V} = \{v_i\}_{i=1}^{M}$ is the node set of size M describing the defined region of interests (ROIs), $\mathcal{E}_i = \mathcal{V} \times \mathcal{V}$ is the weighted edge set, and $\boldsymbol{A}_i \in \mathbb{R}^{M \times M}$ is the weighted adjacency matrix representing the connectivity among ROIs. Since each disease can be recorded in multiple datasets and each dataset can have multiple views of brain networks, we define a training task to be the prediction of one disease on a specific view of brain networks (e.g., different types of functional networks and structural networks). In our cross-dataset multitask learning setting, one aims to train a $\boldsymbol{\Theta}$ parameterized model $f(\cdot)$ on a set of source tasks $\boldsymbol{S} = \{S_k\}_{k=0}^{K}$ to obtain $\boldsymbol{\Theta_0}$ such that the weights capture generalized domain knowledge of brain structures that are useful and transferable to an unseen target task $\boldsymbol{T}$, where $\boldsymbol{S}$ and $\boldsymbol{T}$ do not necessarily concern the same type of disease. One can then aim to fine-tune $f(\cdot)$ on $\boldsymbol{T}$ such that the model can efficiently adapt to the target task optimal $\boldsymbol{\Theta}^*$ given that available training samples in $\boldsymbol{T}$ are much fewer than those in $\boldsymbol{S}$.

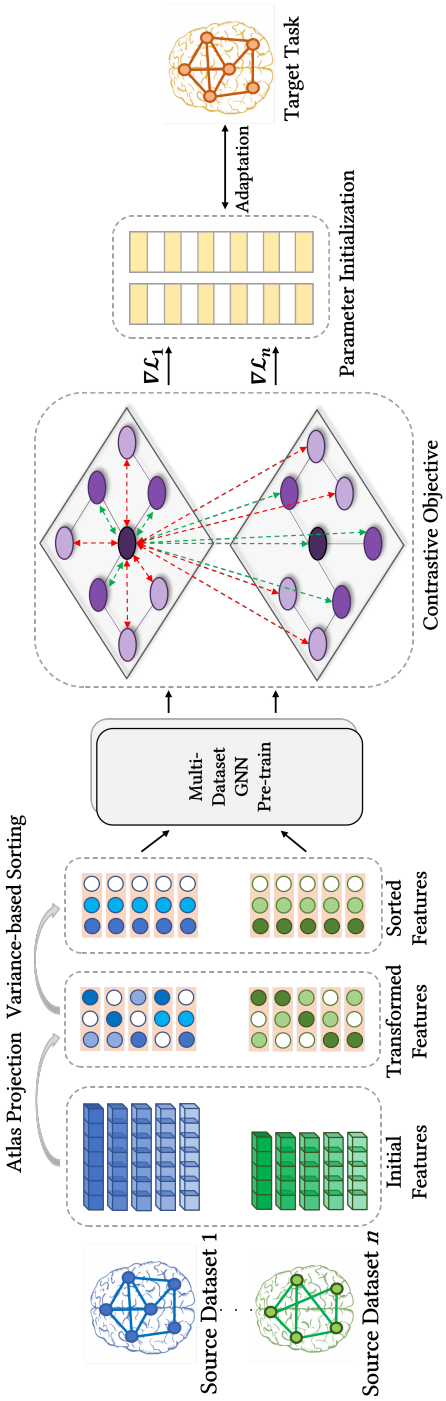


Figure 3.1: Overview of the proposed framework. The initial features of the source datasets are projected to a fixed dimension through atlas transformation followed by variance-based feature alignment, which facilitates self-supervised GNN pre-training on multiple datasets via the novel two-level contrastive learning objective. The learned model can serve as the parameter initialization and be further fine-tuned on target tasks.

Chapter 4

Data-Efficient Training Strategies

Graph neural networks are powerful in learning representations of graph-structured objects such as brain networks. However, under a direct train- and-test setting, GNN needs a relatively large-sized dataset for proper training. With small-sized datasets like brain networks, GNNs may suffer from overfitting and fail to generalize the learned knowledge, which leads to a deterioriated performance in downstream tasks. In this chapter, the paper studies the problem of data-efficient training using multiple sources of datasets. Specifically, given one large-sized dataset and the other smaller-sized datasets, the goal is to study how to pre-train the model on the larger dataset (*i.e.,* the source dataset) and use the learned knowledge to improve the performance on smaller ones (*i.e.,* the target datasets).

In the following, this work proposes to study two data-efficient training strategies for brain network analysis — single-task transfer learning and multi-task meta-learning, both of which are representative techniques in dealing with the absence of sufficient training data. In addition, the work presents two other baseline techniques, namely, learning without pre-training and multi-task transfer learning (without the meta-learning portion).

4.1 Method 1: Learning Without Pre-training (NPT)

As a baseline investigation, one directly applies and trains a randomly initialized model on a given task. The model is optimized under a given objective function. The discussion of the setup of objective functions for pre-training is deferred to the subsequent chapter when introducing the self-supervised training strategies in Chapter 5. Specifically under this setting, the source dataset (*i.e.,* the larger-sized dataset) is not used. Instead, the model is directly evaluated on the smaller-sized target dataset, which indeed have limited training samples. Since the downstream objective is to perform binary classification on a specific disease (*i.e.,* determine whether infected or not), the binary cross-entropy loss used throughout. In particular, the loss is given as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{|\mathcal{D}|} \sum_{(\mathcal{G}_i, y_i) \sim \mathcal{D}} y_i \log \sigma(f_\theta(\mathcal{G}_i)) + (1 - y_i) \log(1 - \sigma(f_\theta(\mathcal{G}_i))), \tag{4.1}$$

where y_i stands for the ground truth label, and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid activation function on the output logits. The testing performance is reported using k-fold cross-validation and this work reports an averaged metric along with standard deviation.

4.2 Method 2: Single-task Transfer Learning (STT)

At first, this work follows the pre-training and fine-tuning scheme in transfer learning [64] to distill knowledge from source task to target task in a sample-efficient way. This framework consists of two consecutive phases: pre-training and fine-tuning. Specifically, one first trains the encoder model on the source task and apply the model weights to train another encoder on the target task.

In the first pre-training phase, one trains the model on the source tasks using the objective described in Eq. (5.2). Then, in the fine-tuning phase, the trained weights

$\mathbf{\Theta}_0$ are used to initialize another encoder model. This model is then fine-tuned on the target task with the same objective function as Eq. (4.1). Since the model has already learned generic knowledge underneath the source task, one uses a smaller learning rate to optimize the model in the fine-tuning phase. This method is summarized in Algorithm 1. Note that although the source dataset may contain multiple structural views, here this study defines only one view as the source task since in the pre-training phase, the model is trained based on one unified objective function and cannot distinguish between multiple tasks, if they are arbitrarily grouped together.

4.3 Method 3: Multi-task Transfer Learning (MTT)

Pre-training on a singular source task is vulnerable to the inherent risk of information loss during transfer learning since the knowledge gaps among source and target domains are not readily quantifiable. This motivates the investigation to train a model that is initialized on some shared knowledge in multiple source tasks when they are available, such that the fine-tuning performance is not conditioned upon any particular knowledge inconsistencies from a source and target pair.

As an immediate solution, this work extends STT into a multi-task setting by expanding the pre-training phase into simultaneously co-learning over multiple source objectives. That is, one can regard each modality from the dataset as an individual task and the model now learns over multiple modalities. To this end, this work formulates all tasks into a distribution, where during pre-training, the model is trained on several objectives sampled from the task distribution, hence the name "multi-task" for this method.

Specifically for each pre-training iteration, one can optimize the model parameters

on a merged objective function which takes the sum over the pre-training objective as given in Eq. (5.2) on all source tasks. For an efficient computation, each iteration processes a mini-batch of data sampled from the source dataset. The learned weights are then used in the fine-tuning phase on the target task following the conventional downstream evaluation procedure, identical to NPT.

4.4 Method 4: Multi-task Meta-Learning (MML)

Meta-learning aims at learning a meta model that is capable of generalizing over a variety of source objectives and can quickly adapt to an arbitrary unseen task. Different from MTT, meta-learning aims at finding an optimal model initialization that enables similarly good performance on multiple pre-training tasks rather than directly combining individual models that are good for each pre-training task through averaging the model weights. This means that meta-learning can achieve better generalization, allowing efficient adaptation to unseen objectives through minimizing the risk of over-fitting the model to outperform on certain tasks while under-perform on others, which is a typical underlying concern of MTT.

Based on such intuition, this work follows the widely adopted model-agnostic meta-learning (MAML) [23] method in brain network learning framework. According to [70], MAML is characterized by two iconic features: (1) rapid learning and (2) feature reuse, which also refers to the outer-loop update and inner-loop adaptation. Specifically, the model is first separately trained on each objective using fast weights during the inner loop function, then the meta parameters of the model are updated by evaluating the loss against the adapted fast weights via the outer-loop module. In other words, the model is optimized by updating on the second-order Hessian of the parameters, which leads to quicker convergence since the optimizer incorporates the

additional curvature information of the loss function that helps estimate the optimal step-size along the optimization trajectory [77]. This effectively reduces the number of training iterations required to achieve a generic model. In addition, the feature reuse inner-loop performs task-specific adaptation, which results in the meta-initialization to be an informative approximation to every task. Due to the fact that the meta-trained model does not pertain to any particular task knowledge, such initialization is therefore non-over-fitting and generically applicable to any unseen target tasks.

To be specific about pipeline design, in the first meta-training phase, one randomly draws n training tasks with a support set (used in inner-loop) and a query set (used in outer-loop) each containing k samples from the pool of training datasets. Then, given the encoder model, the fast weights of the parameters is updated using the objective given in Eq. (5.2) for every pre-training (*i.e.,* source) task. After training the model on all tasks, one updates the meta parameters, *i.e.,* model initialization in our case. Thereafter, in the meta-test phase, one performs the conventional classification procedure on the target data identical to NPT. This method is summarized in Algorithm 2.

Algorithm 1 Single-task supervised transfer learning (STT)

Input: pre-train task S, fine-tune task T, encoder $f(\theta)$
Require: α: learning rate hyperparameter
Randomly initialize θ
▷ Pre-training phase
while not done **do**
    Evaluate the gradient $\nabla_\theta \mathcal{L}_S f(\theta)$
    Update parameters with SGD: $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_S f(\theta)$
end while
▷ Fine-tuning phase
Split T into T_{train} and T_{eval} into K folds
for split **in** K folds **do**
    Get split-specific parameters $\hat{\theta} \leftarrow \theta$
    while not done **do**
        Evaluate the gradient $\nabla_{\hat{\theta}} \mathcal{L}_{T_{\text{train}}} f(\hat{\theta})$
        Update parameters with SGD $\hat{\theta} \leftarrow \hat{\theta} - \alpha \nabla_{\hat{\theta}} \mathcal{L}_{T_{\text{train}}} f(\hat{\theta})$
    end while
    Evaluate ACC, AUC from $f_{\hat{\theta}}(T_{\text{eval}})$
end for

Algorithm 2 Multi-task meta-learning (MML)

Input: meta-train task pool S_τ, meta-test task T, encoder $f(\theta)$
Require: α, β: learning rate hyperparameters
Randomly initialize θ
▷ Meta-training phase
while not done **do**
    for each task τ_i **in** S_τ **do**
        Sample k datapoints $\mathcal{D}_i$ from τ_i
        Evaluate the gradient $\nabla_\theta \mathcal{L}_{\mathcal{D}_i} f(\theta)$
        Compute the adapted parameters $\theta_i' \leftarrow \theta - \beta \nabla_\theta \mathcal{L}_{\mathcal{D}_i} f(\theta)$
        Sample another set of datapoints $\mathcal{D}_i'$ from τ_i
    end for
    Update parameters $\theta \leftarrow \theta - \alpha \nabla_\theta \sum_{\mathcal{D}_i', \theta_i' \sim S_\tau} \mathcal{L}_{\mathcal{D}_i'} f(\theta_i')$
end while
▷ Meta-test phase
Perform k-fold evaluation on target tasks

Chapter 5

Unsupervised Brain Network Pre-training

Given the high cost of acquiring labeled training data for brain network analysis, the pre-training pipeline of this work adopts to the effective label-free learning strategy of contrastive learning (CL). CL aims to maximize the mutual information (MI) between an anchor point of investigation X from a data distribution $\mathcal{H}$ and its positive samples X^+, while minimizing MI with its negative samples X^-. The contrastive objective function is formulated as follows:

$$\mathcal{J}_{\text{con}} = \arg\min \left[\left(-I(X; X^+) + I(X; X^-) \right) \right]. \tag{5.1}$$

In the context of graph CL, given an anchor node representation z_α, a set of positive samples $\mathbf{S}^+$, and a set of negative samples $\mathbf{S}^-$, the training objective is based on the Jensen-Shannon divergence [34],

$$\mathcal{J}_{\text{JSD}}(z_\alpha) = \arg\min \left[\left(-I(z_\alpha; \mathbf{S}^+) + I(z_\alpha; \mathbf{S}^-) \right) \right], \tag{5.2}$$

where

$$I(z_\alpha; \mathbf{S}^+) = \frac{1}{|\mathbf{S}^+|} \sum_{z_{s^+} \in \mathbf{S}^+} \text{sp}\left(\frac{z_\alpha^\top z_{s^+}}{\|z_\alpha\| \|z_{s^+}\|} \right), \tag{5.3}$$

$$I(z_\alpha; \mathbf{S}^-) = \frac{1}{|\mathbf{S}^-|} \sum_{z_{s^-} \in \mathbf{S}^-} \text{sp}\left(\frac{z_\alpha^\top z_{s^-}}{\|z_\alpha\| \|z_{s^-}\|} \right), \tag{5.4}$$

and $\text{sp}(\cdot) = \log(1 + e^{\cdot})$ is softplus nonlinearity.

The ultimate goal of our framework is to localize effective GNN CL learning [102] for brain networks. Given a dataset $\mathcal{D}$ and an anchor node i from graph $\mathcal{G}_p \in \mathcal{D}$ with the learned representation $z_{i,p}$, this work proposes to categorize the possible sample selections into three fundamental types (a visualization is shown in Figure 5.1):

- $\underline{\mathbf{S_1}}$: $\{z_{j,p} : j \in \mathcal{N}_k(i,p)\}$ refers to the node representation set within the the k-hop neighborhood of the anchor in graph $\mathcal{G}_p$.
- $\underline{\mathbf{S_2}}$: $\{z_{j,p} : j \notin \mathcal{N}_k(i,p)\}$ refers to the remaining node representation set in graph $\mathcal{G}_p$ that are not in the the k-hop neighborhood of the anchor.
- $\underline{\mathbf{S_3}}$: $\{z_{j,q} : \mathcal{G}_q \in \mathcal{D}, j \in \mathcal{G}_q, q \neq p\}$ refers to the node representation set of nodes in all the other graphs of dataset $\mathcal{D}$.

Notice that this framework leverages the k-hop substructure around the anchor node to further differentiate $\mathbf{S_1}$ and $\mathbf{S_2}$ for contrastive optimization. This design is driven by two considerations: **(1) Regarding GNN learning.** Given that node representations are learned from the information aggregation of its k-hop neighborhood, maximizing the MI of an anchor to its k-hop neighbors naturally enhances lossless message passing of GNN convolutions. **(2) Regarding the uniqueness of brain networks.** Brain networks can be anatomically segmented into smaller neural

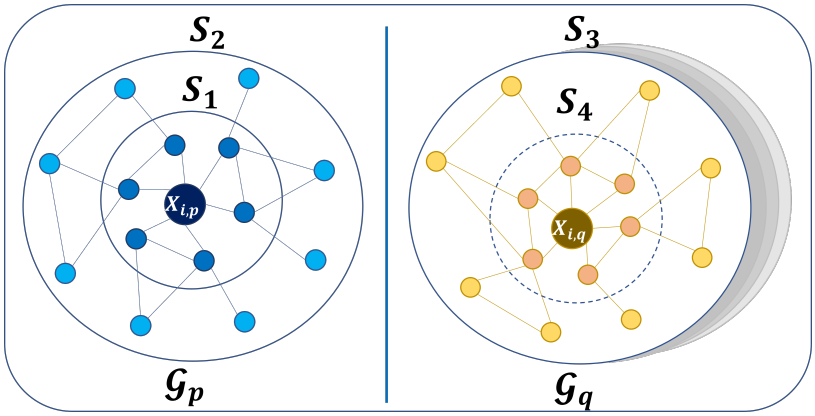


Figure 5.1: Visual demonstration of the sample types where $X_{i,p}$ is the anchor and $\mathbf{S_1}/\mathbf{S_4}$ are sampled as 1-hop neighbors.

system modules [16], thus capturing subgraph-level knowledge can provide valuable signals for brain-related analysis.

Building on these three fundamental types of samples, one can take advantage of the property of brain networks that ROI identities and orders are fixed across samples to introduce an additional sample type. This encourages the GNN to extract shared substructure knowledge by evaluating the MI of an anchor against its presence in other graphs. Given an anchor representation $z_{i,p}$ of node i from graph $\mathcal{G}_p \in \mathcal{D}$, the novel inter-graph sample type is defined as:

- $\underline{\mathbf{S_4}}$:$\{z_{j,q} : j \in \mathcal{N}_k(i,q) \cap \mathcal{N}_k(i,p), \mathcal{G}_q \in \mathcal{D}, q \neq p\}$, refers to the node representation set within the k-hop neighborhood of node i in all other graphs in $\mathcal{D}$. Conceptually, $\mathbf{S_4}$ is a special subset of $\mathbf{S_3}$.

It is important to note that for an anchor node i, its k-hop neighborhood structures might not be identical among different graphs. As a result, one can only consider shared neighborhoods when evaluating the mutual information across multiple graphs. To encourage the learning of unique neighborhood knowledge within a single brain network instance and shared substructure knowledge across the entire dataset, the proposed pipeline configures $\mathbf{S_1}$ and $\mathbf{S_4}$ as positive samples while $\mathbf{S_2}$ and the set $\mathbf{S_3}-\mathbf{S_4}$

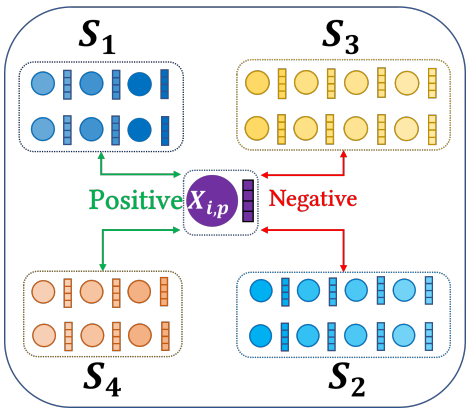


Figure 5.2: The sampling configuration of the proposed framework. $\mathbf{S_1}$ and $\mathbf{S_4}$ are positive samples, $\mathbf{S_2}$ and the set $\mathbf{S_3} - \mathbf{S_4}$ are negative samples.

as negative samples, as illustrated in Figure 5.2. Furthermore, the proposed sampling

Table 5.1: The sampling configuration of existing graph contrastive learning methods.

	$\mathbf{S_1}$	$\mathbf{S_2}$	$\mathbf{S_3}$	$\mathbf{S_4}$
DGI	+	+	/	/
InfoG	+	+	–	/
GCC	+	–	–	/
EGI	+	–	–	/
Proposed	+	–	–	+

categorization can also help understand the objective formulations in various state-of-the-art graph CL frameworks [82, 67, 85, 75, 99]. The findings are summarized in Table 5.1. Specifically, "+" denotes positive sampling; "-" denotes negative sampling; and "/" means that the sample type is not considered. It can be observed that DGI and InfoGraph (InfoG) use graph representation pooled from node representations as a special sample, which is essentially equivalent to jointly considering $\mathbf{S_1}$ and $\mathbf{S_2}$ without explicit differentiation. On the other hand, GCC and EGI, which are more closely related to the proposed framework, leverage neighborhood mutual information maximization on a single graph, but fail to extend this to a multi-graph setting.

Chapter 6

Brain Network Oriented Design Considerations

Unlike conventional graph-structured datasets, brain networks have some unique properties. In this section, this paper first identifies two challenges concerning learning with brain networked data. Accordingly, two design considerations are presented to address these two challenges.

6.1 Data-driven Brain Atlas Mapping

6.1.1 Challenges

For brain network data, ROI templates describe the mapping relationship between nodes and brain atlas. Once the template is chosen, all graphs in a dataset share the same amount of nodes and their physical meanings. In our cross-dataset setting, considering that the source and target datasets are based on different templates, it is difficult to directly transfer the learned knowledge from source to target datasets due to the misalignment of nodes and dimensions in the graphs. Although GNNs are capable of handling input graphs of varied sizes, the model is essentially learning

predictive signals regarding the structures of local subgraphs [50], and thus simply transferring the model parameters without manipulating data-level correspondence may lead to a significant loss of information. Note that we may directly convert different atlas to a unified one through manual mapping. However, finding all such mappings exhaustively is costly and demands tremendous expert efforts because the mapping varies across different pairs of atlas and there is often a lack of ground truth.

To address this issue, this study aims to provide a data-driven atlas mapping solution that is easily accessible and eliminates the strong dependency on network construction. The data-driven atlas mapping solution, which transforms the original node features into lower-dimensional representations that preserve the original connectivity information and align features across datasets, is learned independently on each dataset prior to GNN pre-training.

6.1.2 Autoencoder with Customized Regularizers

The proposed framework adopts a one-layer linear autoencoder (AE) as the base structure that transforms source data into a target dimension with fixed representation in an unsupervised fashion. The AE consists of a linear projection encoder $\mathbf{W}$ and a transposed decoder $\mathbf{W}^\top$, with the goal of learning a low-dimensional projection that can easily reconstruct the original presentation. The loss function is defined as minimizing the reconstruction error

$$\mathcal{L}_{\text{rec}} = (1/M)\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^\top\|_2^2, \tag{6.1}$$

where $\mathbf{X} \in \mathbb{R}^{M\times M}$ is the input and $\mathbf{W} \in \mathbb{R}^{M\times D}$ is the learnable projection [33]. To further enhance the feature compression and to guide the overall AE optimization, this work proposes to incorporate several regularizers that take into account the unique

characteristics of brain networks.:

Locality-Preserving Regularizer (LR)

We aim to ensure that the compressed features preserve the spatial relationships of the original brain surface. To achieve this, we incorporate a locality preserving regularizer [30] to the AE objective. The regularizer is formulated as $\mathcal{L}_{\text{loc}} = (1/M)\|\mathbf{Y} - \mathbf{TY}\|^2$, where $\mathbf{Y} \in \mathbb{R}^{M\times D}$ represents the projected features from the AE and $\mathbf{T} \in \mathbb{R}^{M\times M}$ is a transition matrix constructed from the k-NN graph of the 3D coordinates of ROIs.

Modularity-Aware Regularizer (CR)

Brain networks can be segmented into various neural system modules that characterize functional subsets of ROIs. In graph terminology, they are community structures. The projected feature should also capture information about neural system membership. However, obtaining ground-truth segmentations is a difficult task that requires expert knowledge. To overcome this challenge, we resort to community detection methods on graphs, specifically based on modularity maximization. The regularizer [72] is defined as minimizing

$$\mathcal{L}_{\text{com}} = -\frac{1}{2D}\sum_{i,j=1}^{M}\left[\mathbf{A}_{ij} - \frac{k_i k_j}{2D}\right]\exp(-\|y_i - y_j\|_2^2), \tag{6.2}$$

where $\mathbf{A} \in \mathbb{R}^{M\times M}$ is the graph adjacency matrix, k_i denotes degree of node i, and y_i is the AE projected features. Essentially, this optimization minimizes the L_2 distance between representations of nodes within the same communities, as measured by the modularity score, and maximizes the distance between representations of nodes in different communities.

Sparsity-Oriented Regularizer (SC)

Sparse networks have proven to be effective in learning robust representations from noisy data [41, 74, 60]. In brain connectome analysis, sparsity has also been shown to improve the interpretation of task-specific ROI connections in generation and classification tasks [42]. To this end, we implement the popular KL-divergence smoothing to enforce sparsity in the parameters of the linear projection encoder, $\mathbf{W}$). This is formulated as:

$$\mathcal{L}_{\mathrm{KL}} = \sum_{i=1}^{M} \sum_{j=1}^{D} \left[\rho \log \left(\frac{\rho}{\hat{\rho}_{ij}} \right) + (1-\rho) \log \left(\frac{1-\rho}{1-\hat{\rho}_{ij}} \right) \right], \tag{6.3}$$

where ρ is a small positive float set as the target sparsity value, and $\hat{\rho}_{ij}$ represents the element-wise activation of the encoder projection matrix $\mathbf{W} \in \mathbb{R}^{M \times D}$.

6.1.3 Variance-based Dimension Sorting

In addition to transforming dataset-specific features, cross-dataset alignment of feature signals is also crucial for improving model adaptation. The one-layer AE transforms the original feature vectors into weighted combinations of multiple dimensions, creating new feature dimensions which this work names as *virtual ROIs*. In the context of brain networks, this process helps to group ROIs and their signals. This idea is inspired by the well-studied functional brain modules [66, 3, 31, 6, 98], which provide a higher-level and generic organization of the brain surface, as opposed to fine-grained ROI systems. Since the variations in ROI parcellations are due to differences in clinical conventions, it is reasonable to assume that there exists a shared virtual ROI system underlying different parcellation systems, similar to the discretization of functional brain modules. The community learning and neighborhood preserving regularizers, introduced in Section 6.1.2, allow one to capture these shared virtual ROIs in a data-driven manner. Our ultimate goal is to align the discovered virtual ROIs

across datasets, so that each virtual ROI characterizes the same functional module in the human brain, regardless of its origin. This cross-dataset alignment of virtual ROIs ensures that the model can effectively adapt to new datasets and provide meaningful insights into the different downstream analyses.

The objective of the one-layer linear AE is similar to PCA, as discussed in more detail in Appendix A.1, with the added benefit of incorporating additional regularizers. PCA orders dimensions based on decreasing levels of sample variance [35]. The proposed framework leverages this approach by utilizing the learned parameters of the AE projection to estimate the variance of each virtual ROI (*i.e.*, projected feature dimension). The sample variance of each virtual ROI indicates its representativeness of the original data variations. Given the shared patterns across different parcellation systems, one can expect that similar virtual ROIs in datasets with different atlas templates will have similar variance scores, especially in terms of their order. By sorting the same number of virtual ROIs based on their sample variance in each dataset, the proposed framework aims to align virtual ROI cross datasets, so that each virtual ROI represents the same functional unit in the human brain. The procedure is explained in detail in Algorithm 4 in Appendix A.2.

6.2 Source Task Reweighing

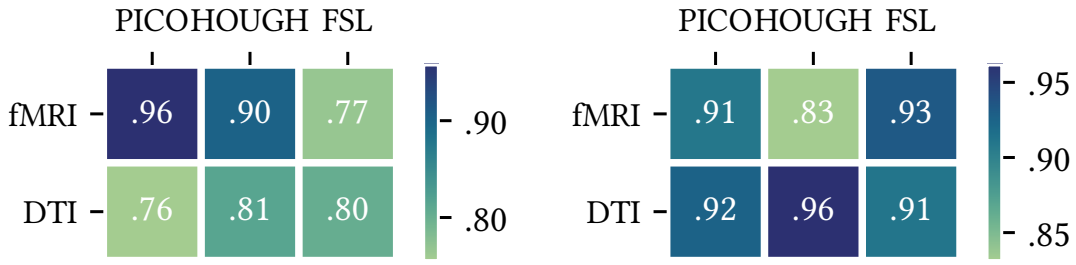
6.2.1 Challenges

Another challenge of cross-dataset brain network analysis is that the previous base meta-learning pipeline fails to consider relative difficulty of different individual tasks. It is possible that varying the choice of source task does not lead to uniform improvements on the target performance. This means that one can suspect that this is because some tasks are easier to learn than others, which will converge faster during

the meta training phase. In other words, the base meta-learning pipeline fails to equally capture the latent knowledge of all source datasets, which potentially hinders the ability of generalization.

6.2.2 Dynamic Task Reweighing

The first step is to investigate the data-level task correlations. In particular, this work analyzes the task similarity between the HIV and BP modalities (*i.e.,* target datasets) with respect to the PPMI modalities (*i.e.,* source dataset). A detailed introduction of the datasets including the variety of data modalities and their pre-processing information are deferred to Chapter 7. Inspired by task2vec [1], for each task, one calculates a respective task embedding that stores information regarding its learning difficulty and latent knowledge. In particular, the embedding is derived from the Fisher information estimation of the positive semidefinite upper bound of the Hessian matrix, on which the model is trained on an encoder model using the same objective in Eq. (5.2). This work visualizes the task correlation in cosine similarity among the embeddings on HIV and BP in 6.2.2 respectively. It can be seen that there is an inherent correlation among source (*i.e.,* PPMI) and target (*i.e.,* HIV, BP) datasets which indicates that there exists shared properties and latent information among the three categories of brain network data. This observation can be corroborated with existing clinical research presented in earlier studies [62, 17, 63, 22], where detailed analyses on the coexistence and co-influence among BP, HIV, and PPMI disease are discussed. This validates the working effectiveness of the cross-dataset learning setting since useful and transferable inter-domain knowledge and shared features can be discovered by learning on a source data. In addition, the visualization also shows a non-uniform task correlation, which suggests that the source tasks are prescribed to varying level of learning and adaptation difficulty relative to the given target task. This demonstrates that the optimizer tends to distribute unequal attention within



(a) PPMI and BP Task Correlation

(b) PPMI and HIV Task Correlation

Figure 6.1: Task correlations among different data modalities from source and target datasets. The Fisher information estimation is first computed to derived from the Hessian matrix by training each task using the same architecture as in Section 4.4. The task embedding is then composed of layer-wise concatenation of the flattened Fisher information matrix.

the source set during meta-training and that the learned initialization will eventually skew towards the optimal of "easier" tasks and fails to generalize over "harder" tasks. This motivates us to develop dynamic inner-loop optimization rules during meta-training towards an unbiased generalization ability.

Following the mechanism proposed by ALFA [4], during the task-specific inner-loop update, the proposed framework implements a trainable hyperparameter generator that guides the rate of convergence for the gradient-descent update. The generator processes the learning state as input, which is consisted of a stacked layer-wise value of model parameter and gradient estimate. The generator then outputs a layer-wise learning rate and weight decay coefficient conditioned on the current learning state. Then, its parameters are updated by the query loss objective as in Eq. (5.2). Different from the original ALFA, where the encoder parameters are frozen from updating at the outer-loop phase, we allow the encoder to be trainable on the query set for quicker adaptation. This variant is summarized (dubbed AR) in Algorithm 3.

Algorithm 3 Multi-task meta-learning with adaptive task reweighing (AR)

Input: meta-train tasks S_τ, meta-test task T, encoder $f(\theta)$, hyperparameter generator $g(\phi)$
Require: η: outer-loop learning rate
Randomly initialize θ, ϕ
▷ Meta-training phase
while not done **do**
    for each task τ_i in S_τ **do**
        Sample n datapoints $\mathcal{D}_i$ from τ_i
        Evaluate the gradient $\nabla_\theta \mathcal{L}_{\mathcal{D}_i} f(\theta)$
        Obtain the task-specific learning state $\rho_i = [\nabla_\theta \mathcal{L}_{\mathcal{D}_i} f(\theta), \theta]$
        Generate hyperparameters α, $\beta = g_\phi(\rho_i)$
        Compute the adapted parameters $\theta_i' \leftarrow \beta \odot \theta - \alpha \odot \nabla_\theta \mathcal{L}_{\mathcal{D}_i} f(\theta)$
        Sample another set of datapoints $\mathcal{D}_i'$ from τ_i
    end for
    Update parameters $\theta \leftarrow \theta - \eta \nabla_\theta \sum_{\mathcal{D}_i', \theta_i' \sim S_\tau} \mathcal{L}_{\mathcal{D}_i'} f(\theta_i')$
    Update parameters $\phi \leftarrow \phi - \eta \nabla_\phi \sum_{\mathcal{D}_i', \theta_i' \sim S_\tau} \mathcal{L}_{\mathcal{D}_i'} f(\theta_i')$
end while
Perform k-fold evaluation on target tasks

Chapter 7

Dataset and Experimental Configuration

7.1 Dataset Details

The empirical study in this work uses three real-world brain network datasets: 1) the Bipolar Disorder (BP) dataset, 2) the Human Immunodeficiency Virus Infection (HIV) dataset, and 3) the Parkinson's Progression Markers Initiative (PPMI) dataset. The BP and HIV are private datasets, while the large-scale PPMI dataset[1] is publicly available for authorized users. The study has been approved by an Institutional Review Board (IRB) to ensure the ethical and responsible use of human subjects in research. The IRB reviewed and approved the study protocols and consent forms, ensuring that the rights and welfare of the participants are protected. The study strictly adheres to the Good Clinical Practice guidelines and U.S. 21 CFR Part 50 (Protection of Human Subjects) to ensure the safety and privacy of the participants. All the data used in this work is processed anonymously to protect the privacy of participants, and no personally identifiable information is used or disclosed.

[1] `https://www.ppmi-info.org/`

7.1.1 Parkinson's Progression Markers Initiative (PPMI)

This is a restrictively public available dataset[2] to speed breakthroughs and support validation on Parkinson's Progression research. This dataset contains 718 subjects, where 569 subjects are Parkinson's Disease (PD) patients and the rest 149 are Healthy Control (HC). The raw imaging signals are pre-processed by Eddy-current and head motion correction using FSL[3] and the brain networks are extracted using the same tool. The EPI-induced susceptibility artifacts correction is handled using Advanced Normalization Tools (ANT)[4]. In the meantime, 84 ROIs are parcellated from T1-weighted structural MRI using Freesurfer[5]. The brain networks are constructed using three whole brain tractography algorithms namely the Probabilistic Index of Connectivity (PICo), Hough voting (Hough), and FSL. Each resulted network for each subject is 84×84. Each brain network is normalized by the maximum value to avoid computation bias for the later feature extraction and evaluation, since matrices derived from different tractography algorithms differ in scales and ranges. The final brain networks were parcellated according to the Desikan-Killiany 84 template.

7.1.2 Bipolar Disorders (BP)

This local dataset is composed of the resting-state fMRI and DTI image data of 52 Bipolar I subjects who are in euthymia and 45 Healthy Controls (HCs) with matched age and gender [8, 58]. The fMRI data was acquired on a 3T Siemens Trio scanner using a $T2^*$ echo planar imaging (EPI) gradient-echo pulse sequence with integrated parallel acquisition technique (IPAT) and DTI data were acquired on a Siemens 3T Trio scanner. The brain networks are constructed using the CONN[6] toolbox and are parcellated using the Brodmann 82 template. A normalization and smoothing after

[2] https://www.ppmi-info.org/
[3] https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/
[4] http://stnava.github.io/ANTs/
[5] https://surfer.nmr.mgh.harvard.edu/
[6] http://www.nitrc.org/projects/conn/

first realigning and co-registering were performed on the raw EPI pictures. After that, the signal was regressed to remove the confounding effects of the motion artifact, white matter, and CSF. The 82 defined ROIs, also identified as cortical and subcortical gray matter regions were produced by Freesurfer, and pairwise signal correlations were used to build the brain networks.

7.1.3 Human Immunodeficiency Virus Infection (HIV)

This local dataset involves fMRI and DTI brain networks for 70 subjects, with 35 of them early HIV patients and the other 35 Healthy Controls (HCs). These two groups of subjects do not differ in demographic distributions such as age and biological sex. The preprocessings for fMRI including brain extraction, slice timing correction and realignment are managed with the DPARSF[7] toolbox, while the preprocessings for DTI such as distortion correction are finished with the help of FSL[3] toolbox. Finally, brain networks with 90 regions of interest are parcellated based on the automated anatomical labeling (AAL 90) atlas template [78].

7.2 Experimental Setup

7.2.1 Backbone Selection and Evaluation Metric

The proposed framework employs GCN as the backbone for the GNN [47] encoder. The experiment also benchmarks using GAT [81] and GIN [87], and the results are provided in Appendix B.1. The hyperparameter tuning follows the standard designs in related studies such as in [89, 84, 37]. The downstream evaluation is binary graph classification for disease prediction. To assess the performance, this experiment uses the two widely used metrics in the medical field [51, 14]: accuracy score (ACC) and the area under the receiver operating characteristic curve (AUC).

[7] `http://rfmri.org/DPARSF/`

7.2.2 GNN Setup

The GCN encoder is composed of 4 graph convolution layers with hidden dimensions of 32, 16, 16, and 8. Similarly, the GAT encoder is built from 4 graph attention layers with hidden dimensions of 32, 16, 16, and 8. Regarding GIN, which is slightly different, the encoder consists of 4 MLP layers with each MLP containing 2 linear layers with a unifying hidden dimension of 8.

7.2.3 Pre-training Pipeline Setup

For two-level node contrastive sampling, we set $k = 2$ as the radius regarding k-hop neighborhood sampling for $\mathbf{S_1}$ and $\mathbf{S_4}$. To enable efficient computation on multi-graph MI evaluation, we resort to mini-batching and we set a default batch size of 32. In addition, we leverage the popular Adam [45] optimizer with the learning rate set to 0.002 as well as the cosine annealing scheduler [55] to facilitate GNN training. In general, a complete pre-training cycle takes 400 epochs with an active deployment of early stopping.

7.2.4 Atlas Mapping Regularizer Setup

Following the discussion in section 6.1.2, the total running loss of the AE projection is given as:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \alpha\mathcal{L}_{\text{loc}} + \beta\mathcal{L}_{\text{com}} + \gamma\mathcal{L}_{\text{KL}}, \tag{7.1}$$

in particular, we set α, $\beta = 0.8$ and $\gamma = 0.01$. The one-layer AE encoder transforms the feature signals from all given datasets into a universally projected dimension of 32. For the details of locality-preserving regularizer (*i.e.,* $\mathcal{L}_{\text{loc}}$), the transition matrix $\mathbf{T}$ is built from the 5-nearest-neighbor graph from the 3D coordinates of each atlas templates. For the sparsity-oriented regularizer (*i.e.,* $\mathcal{L}_{\text{KL}}$), the target sparsity value ρ is set to $1e^{-5}$. The overall optimization process, which is similar to model pre-training,

takes a total of 100 epochs with a learning rate of 0.02.

7.2.5 Downstream Evaluation Setup

For each target evaluation, the fine-tuning process features a 5-fold cross-validation, which approximately splits the dataset into 70% training, 10% validation, and 20% testing. To prevent model over-fitting, we implement a L_2 penalty with a coefficient of $1e^{-4}$. Overall, the model fine-tuning process, which is nearly identical to the other two training procedures, takes a total of 200 epochs with a learning rate of 0.001 and a cosine annealing scheduler.

Chapter 8

Experiments and Analysis

The effectiveness of the proposed framework is evaluated through extensive experiments on real brain network datasets, with a focus on the following research questions:

- **RQ1**: How does the proposed framework compare with other unsupervised GNN pre-training frameworks adapted to the scenario of brain networks?
- **RQ2**: What is the contribution of each major component in the proposed framework to the overall performance?
- **RQ3**: How does the choice of sampling method affect model convergence during pre-training and performance in downstream adapting?
- **RQ4**: How effective is the variance-based sorting in aligning virtual ROIs among different parcellation systems?

8.1 Overall Performance Comparison (RQ1)

A comprehensive comparison of the target performance between the proposed framework and popular unsupervised learning strategies is presented in Table 8.1. To fairly compare the methods, the atlas mapping pre-processing, the multi-task meta-learning

Table 8.1: Disease prediction performance comparison. All results are averaged from 5-fold cross-validation along with standard deviations. The best result is highlighted in bold and runner-up (excluding the w/o AR variant) is underlined.

Type	Method	BP-fMRI		BP-DTI		HIV-fMRI		HIV-DTI	
		ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
NPT	GCN	$50.07_{\pm 13.70}$	$50.11_{\pm 15.48}$	$49.51_{\pm 14.68}$	$51.83_{\pm 13.98}$	$56.27_{\pm 15.84}$	$57.16_{\pm 15.14}$	$51.30_{\pm 16.42}$	$53.82_{\pm 14.94}$
TFL	STT	$53.92_{\pm 12.82}$	$54.61_{\pm 11.76}$	$55.51_{\pm 15.74}$	$56.73_{\pm 16.23}$	$61.18_{\pm 14.57}$	$62.88_{\pm 15.58}$	$55.29_{\pm 13.28}$	$57.31_{\pm 14.72}$
	MTT	$60.37_{\pm 12.42}$	$61.64_{\pm 14.83}$	$59.41_{\pm 11.62}$	$59.92_{\pm 13.73}$	$67.65_{\pm 12.26}$	$68.38_{\pm 12.94}$	$60.54_{\pm 13.83}$	$59.46_{\pm 12.33}$
NCL	Node2Vec	$48.51_{\pm 10.39}$	$49.68_{\pm 7.23}$	$50.83_{\pm 8.14}$	$46.70_{\pm 10.33}$	$52.61_{\pm 10.38}$	$50.75_{\pm 10.94}$	$49.65_{\pm 10.30}$	$51.22_{\pm 10.79}$
	DeepWalk	$50.28_{\pm 9.33}$	$51.59_{\pm 9.06}$	$52.17_{\pm 9.74}$	$48.36_{\pm 9.37}$	$54.81_{\pm 11.26}$	$55.55_{\pm 11.93}$	$52.67_{\pm 11.42}$	$50.88_{\pm 10.53}$
	VGAE	$56.71_{\pm 9.68}$	$55.24_{\pm 11.48}$	$54.63_{\pm 12.09}$	$54.21_{\pm 11.94}$	$62.76_{\pm 9.47}$	$61.25_{\pm 11.61}$	$56.90_{\pm 9.72}$	$55.35_{\pm 9.04}$
SCL	GBT	$57.21_{\pm 10.68}$	$57.32_{\pm 10.48}$	$56.29_{\pm 9.35}$	$55.27_{\pm 10.54}$	$65.73_{\pm 10.93}$	$66.08_{\pm 10.43}$	$59.80_{\pm 9.76}$	$57.37_{\pm 9.49}$
	GraphCL	$59.79_{\pm 9.36}$	$59.10_{\pm 10.78}$	$57.57_{\pm 10.63}$	$57.35_{\pm 9.67}$	$67.08_{\pm 9.70}$	$69.17_{\pm 10.68}$	$60.43_{\pm 8.39}$	$60.03_{\pm 10.48}$
	ProGCL	$62.36_{\pm 8.90}$	$62.61_{\pm 9.34}$	$61.26_{\pm 8.37}$	$\underline{62.67_{\pm 8.46}}$	$71.52_{\pm 9.19}$	$72.16_{\pm 9.85}$	$\underline{62.48_{\pm 10.38}}$	$61.94_{\pm 10.57}$
MCL	DGI	$62.44_{\pm 10.12}$	$60.75_{\pm 10.97}$	$58.15_{\pm 9.63}$	$58.95_{\pm 9.60}$	$70.22_{\pm 11.43}$	$70.12_{\pm 12.46}$	$60.83_{\pm 10.84}$	$62.06_{\pm 10.16}$
	InfoG	$62.87_{\pm 9.52}$	$62.37_{\pm 9.67}$	$60.88_{\pm 9.97}$	$60.44_{\pm 9.61}$	$72.46_{\pm 8.71}$	$72.94_{\pm 8.68}$	$61.75_{\pm 9.76}$	$61.37_{\pm 9.85}$
EGS	GCC	$\underline{63.45_{\pm 9.82}}$	$62.39_{\pm 9.08}$	$60.44_{\pm 9.54}$	$60.29_{\pm 10.33}$	$70.97_{\pm 10.31}$	$72.48_{\pm 11.36}$	$61.27_{\pm 9.66}$	$61.38_{\pm 10.72}$
	EGI	$63.38_{\pm 8.93}$	$\underline{63.58_{\pm 8.02}}$	$\underline{61.82_{\pm 8.53}}$	$61.57_{\pm 8.27}$	$\underline{73.46_{\pm 8.49}}$	$\underline{73.28_{\pm 8.68}}$	$60.89_{\pm 9.87}$	$\underline{62.41_{\pm 8.50}}$
Ours	w/o AR	$64.15_{\pm 9.03}$	$64.24_{\pm 8.31}$	$62.41_{\pm 8.52}$	$65.84_{\pm 8.74}$	$74.93_{\pm 9.04}$	$75.74_{\pm 7.80}$	$64.39_{\pm 9.41}$	$64.38_{\pm 9.35}$
	Full	$\mathbf{68.84_{\pm 8.26}}$	$\mathbf{68.45_{\pm 8.96}}$	$\mathbf{66.57_{\pm 7.67}}$	$\mathbf{68.31_{\pm 9.39}}$	$\mathbf{77.80_{\pm 9.76}}$	$\mathbf{77.22_{\pm 8.74}}$	$\mathbf{67.51_{\pm 8.67}}$	$\mathbf{67.74_{\pm 8.59}}$

learning backbone, and the task adaptive reweighing algorithm discussed in section 4.4 are applied to all benchmarked methods, with only few exceptions including STT, MTT, and Ours w/o AR (*i.e.,* without task adaptive reweighing). The purpose of this comparison is to effectively highlight the impact of the proposed two-level contrastive pre-training and there will be further analysis on the effect of atlas mapping in subsequent sections. In addition, for a clearer presentation, this experiment groups the selected baselines according to their optimization strategies:

- No pre-training (NPT): the backbone with randomly initialized parameters for target evaluation.
- Transfer learning (TFL): methods that are formed based on transfer learning paradigm discused in Sections 4.2 4.3 on STT and MTT. Specifically, for STT, the pre-training dataset is defined as the PICo modality of PPMI.
- Non-CL-based (NCL): methods with cost functions regularized by co-occurrence agreement or link reconstruction, including Node2Vec [25], DeepWalk [65], and VGAE [46].

- Single-scale CL (SCL): methods utilizing either node- or graph-level representations in the CL optimization, including GBT [5], ProGCL [85], and GraphCL [93].

- Multi-scale CL (MCL): methods whose CL optimization utilizes both nodes- and graph-level representations, including DGI [82] and InfoG [75].

- Ego-graph sampling (EGS): methods whose contrastive samplings consider k-hop ego-networks as discriminative instances, which are the most similar to the proposed framework, including GCC [67] and EGI [99].

The experiments reveal the following insights:

- The proposed framework of ours consistently outperforms all the baselines, achieving a relative improvement of 7.34%-13.30% over the best-performing baselines and 31.80%-38.26% over the NPT setting. The reported results of the selected baselines are also statistically compared against that of the proposed framework under the paired t-test. With the significance level set to 0.05, the largest two-tailed p value is reported at 0.042, which means that the proposed framework demonstrates statistically significant performance increase over other selected methods.

- The transfer learning based pipeline, including STT and MTT improve over the NPT baseline with a relative gain 8.27% and 16.67% respectively across both metrics, suggesting the relative benefit of model pre-training and knowledge transfer. However, the transfer learning setting still suffers high variance in performance results and inferior overall performance compared to the proposed full framework.

- Compared with the transductive methods of Node2Vec and DeepWalk, the GNN pre-trained by VGAE learns structure-preserving representations and achieves

the best results in the NCL-type methods. This indicates the potential benefit of the locality-preserving regularizer design in the proposed framework.

- Maximizing mutual information between augmented instances may hinder GNNs from learning a shared understanding of the entire dataset. For baselines belonging to the categories of SCL, MCL, and EGS, pre-training with non-augmented CL (InfoG, EGI) generally results in a 4.36% relative improvement across both metrics and a 7.63% relative decrease in performance variance compared to their augmentation-based counterparts (GBT, GraphCL, ProGCL, DGI, GCC). This explains why the proposed framework does not employ data augmentation.

- Multi-scale MI promotes the capture of effective local (*i.e.,* node-level) representations that can summarize the global (*i.e.,* graph-level) information of the entire network. The MCL-type methods typically outperform the SCL-type ones by a relative gain of 2.68% in ACC and 3.27% in AUC.

- The group of baselines considering k-hop neighborhoods (EGS) presents the strongest performance, indicating the importance of local neighborhoods in brain network analysis. The proposed the proposed framework, which captures this aspect through both node- and graph-level CL, is the only one that comprehensively captures the local neighborhoods of nodes.

- The added component of task adaptive reweighing demonstrates a promising working effectiveness by bringing over the non reweighed training with a relative improvement of 4.29% in accuracy score and 5.80% in AUC metric. This shows that the issue of task-biased convergence during multi-task pre-training exists and can be mitigated by additional handling through reweighing mechanisms.

Table 8.2: The four variants of sampling strategies.

	$\mathbf{S_1}$	$\mathbf{S_2}$	$\mathbf{S_3}$	$\mathbf{S_4}$
Var. 1	–	–	/	/
Var. 2	+	–	/	/
Var. 3	+	–	–	/
Var. 4	+	+	–	/

8.2 Ablation Studies (RQ2)

This ablation studies examine two key components of the proposed framework - (1) the two-level contrastive sampling and (2) the atlas mapping regularizers. The best contrastive sampling configuration is fixed when examining the atlas regularizers, and all regularizers are equipped when examining the contrastive samplings. The results, shown in Figure 8.1 (with additional DTI version in Appendix B.2), are analyzed based on the four possible variants of contrastive sampling listed in Table 8.2. The analyses yield the following observations:

- leveraging k-hop neighborhood (*i.e.,* positive $\mathbf{S_2}$) MI maximization brings visible performance gain, confirming its benefit in brain structure learning.
- The extension to multi-graph CL (*i.e.,* consideration of $\mathbf{S_3}$) facilitates the extraction of unique ROI knowledge, leading to improved results in Var. 3/4.
- Var. 4 outperforms Var. 3 as it effectively summarizes of global (*i.e.,* graph-level) information in local node representations.
- The full implementation of the proposed framework brings a relative gain of 4.27% in both metrics on top of Var. 4, highlighting the significance of considering shared substructure knowledge across multiple graphs (*i.e.,* through the inclusion of $\mathbf{S_4}$).

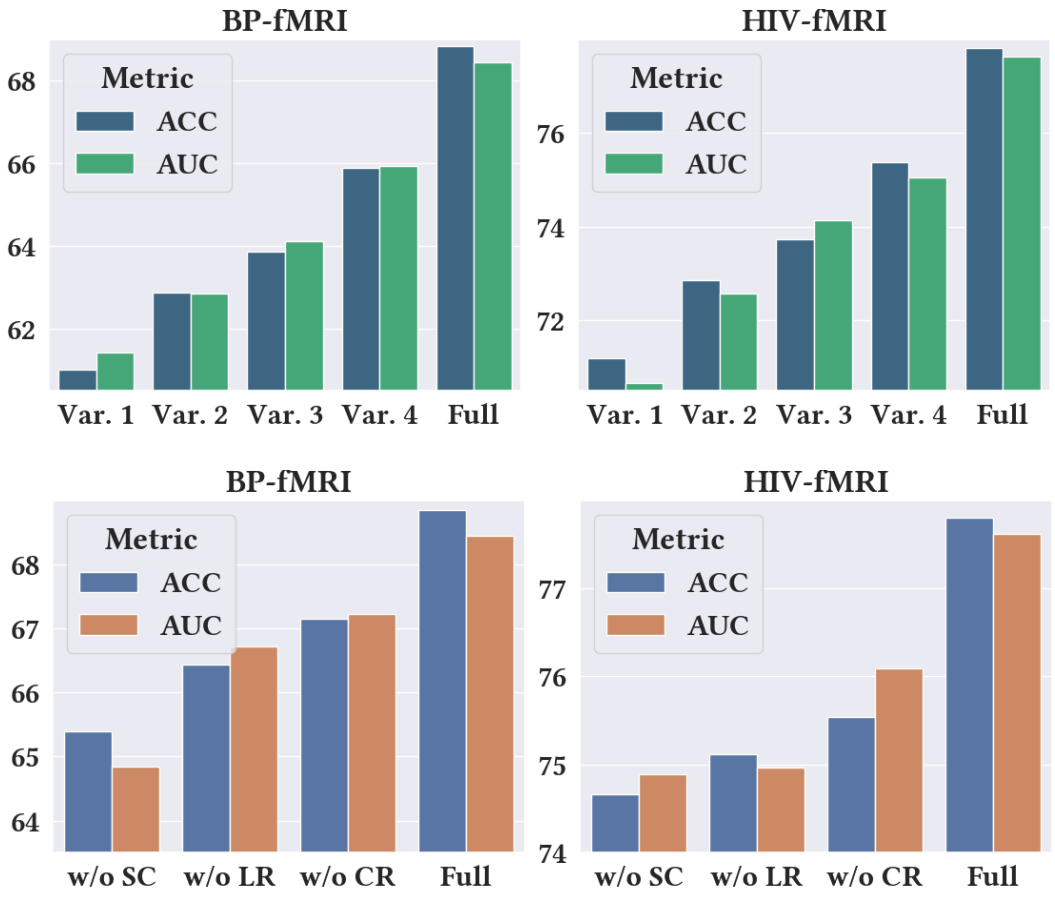


Figure 8.1: Ablation comparisons on contrastive sampling choices (top two) and atlas mapping regularizers (bottom two). The y-axis refers to the numeric values of evaluated metrics (in %).

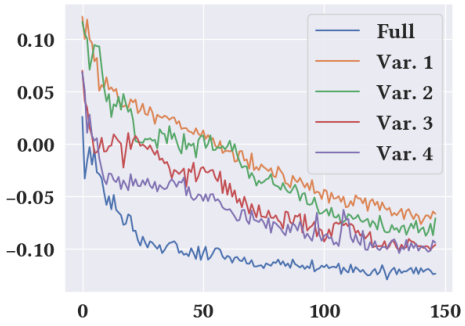
The bottom two sub-figures examine the impact of the atlas mapping regularizers by comparing the results of the full framework to those without the sparsity regularizer (w/o SR), the locality regularizer (w/o LR), and the community regularizer (w/o CR). Two key observations are made:

- The removal of SR leads to the greatest performance drop, emphasizing its crucial role in learning robust projections that can effectively handle noise and prevent over-fitting.
- The inferior results when LR and CR are absent emphasize the importance of spatial sensitivity and blockwise feature information in brain network analysis. This supports our intuition to consider the relative positioning of ROIs in the 3D coordinate as well as knowledge on community belongings based on modularity measures.

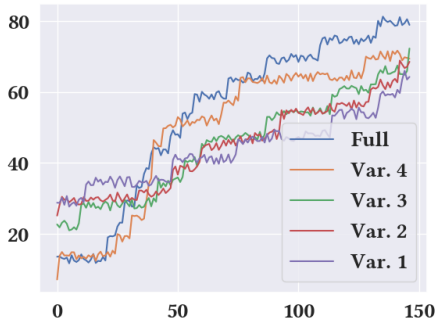
8.3 Analysis of Two-level Contrastive Sampling (RQ3)

Figure 8.2 offers insight into the pre-training convergence, target adaptation progression, and pre-training runtime consumption of the four sampling variants and the full framework. Key observations include:

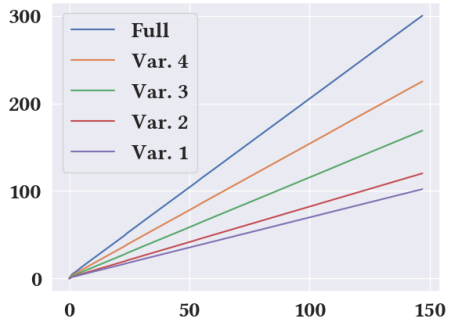
- As seen in Figure 8.2(a), all variants demonstrate efficient pre-training convergence due to the multi-dataset joint optimization inspired by MAML. The full model demonstrates the most optimal convergence, highlighting the advantage of learning shared neighborhood information in brain network data through two-level node contrastive sampling.
- Figure 8.2(b) shows the superiority of our design in terms of downstream adaptation performance compared to other variants.



(a) Pre-train loss on PPMI



(b) Test ACC (%) on HIV-fMRI



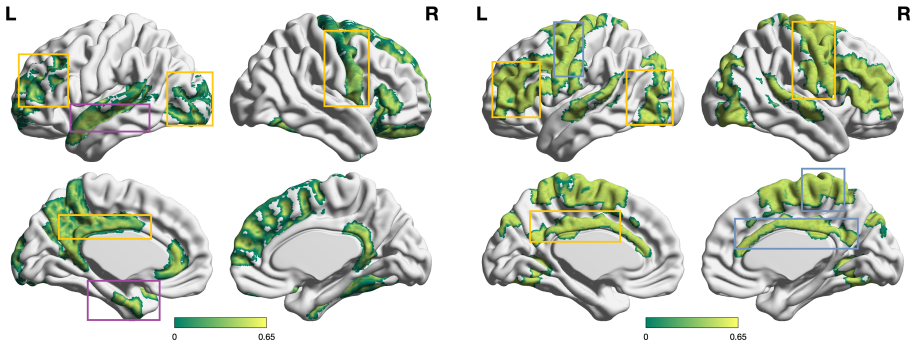
(c) Pre-train time (secs) on PPMI

Figure 8.2: In-depth comparison among the four variants and the full model. The x-axis is epochs.

- Figure 8.2(c) reveals that the more sophisticated the sampling considerations result in greater computational complexity for mutual information evaluation, leading to longer runtime for each pre-training epoch. However, the total time consumptions are all on the same scale.

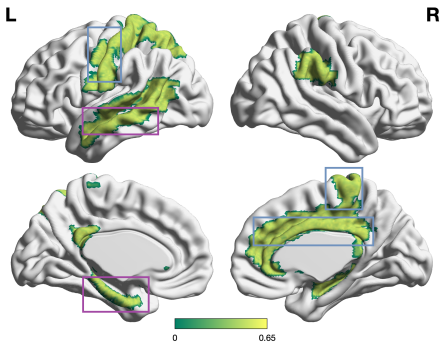
8.4 Analysis of ROI Alignment (RQ4)

To further validate the variance-based virtual ROI sorting, this experiment selects the top 2 virtual ROIs with the highest sample variances for each atlas template (*i.e.*, dataset) and backtrack to locate their corresponding projected ROIs. The results are illustrated in Figure 8.3, which shows a 3D brain surface visualization highlighting the original ROIs. From this, one can draw two main conclusions:



(a) PPMI mapping

(b) BP mapping



(c) HIV mapping

Figure 8.3: The virtual ROI mapping across the three investigated datasets. Overlapping regions are highlighted with colored boxes. In particular, the annotation use gold-colored boxes for the PPMI and BP atlases; blue-colored boxes for the BP and HIV atlases; and purple-colored boxes for the PPMI and HIV atlases.

- There exists multiple regional overlaps between pairs of two atlas templates, reflecting some working effectiveness of our proposed solution as well as confirming the feasibility of converting between atlas templates.

- It is relatively harder to find regions that overlap across all three atlas templates which shows a limitation of the proposed unsupervised ROI alignment scheme, suggesting a need to modify against the current heuristic that considers beyond mere variance measures which may inspire further study and research opportunity.

Chapter 9

Conclusions and Future Directions

This work focuses on data efficient learning on small-sized brain network datasets through leveraging meta learning techniques, self-supervised contrastive pre-training objectives, and brain network oriented design considerations. The experiments have demonstrated the effectiveness of the proposed framework in the application of brain network based disease predictions. This work is also the first to discover the inherent challenges in learning on small-sized brain network datasets and formulate this problem into a data-efficient learning objective, where the goal to find a generalizable model initialization that achieves efficient adaption on target tasks. To this end, the proposed framework leverages transfer learning and, more importantly, meta-learning strategies to serve as backbone frameworks for model pre-training. In addition, the framework also features a novel two-level contrastive sampling strategies to enable unsupervised model pre-training. Considering the special properties of brain networks from traditional graphical data, the framework proposes an automated atlas transformation design and variance-based sorting to help address the incompatibility challenge of cross-dataset brain network ROI template dimensions. Besides, the proposed framework also introduces an adaptive task reweighing algorithm that helps resolve biased learning issues in the conventional meta-training pipeline. Extensive ex-

perimentation demonstrated the effectiveness of proposed methodologies. It is worth noting that the proposed framework is naturally generic and can be easily scaled to other types of neuroimaging datasets. The training pipeline can also be generalized to any parameterized model that is optimized on any customizable objectives and data sampling strategies.

However, Learning on brain network data is still prescribed to various challenges. First, most brain networks are expressed by multiple views and modalities, in which to achieve a comprehensive feature extraction, would require GNN models to capture complex inter-relations within graph modalities. Simply applying multi-facet meta-learning and separately optimizing on individual views fail to consider the intricacies of some shared and complementary knowledge underneath the multi-view datasets. Second, the target performance on supervised disease classification still suffers from relatively high data variance under the k-fold evaluation scheme. This suggests that, assuming given a balanced dataset, the current GNN models are sensitive to batch effects, which would require additional handling of data noise and further development of GNN models that achieve good out-of-distribution performance. For future investigation, my research will primarily focus on addressing the aforementioned challenges by performing theoretical and empirical analyses on GNN architectures for brain network learning. To tackle the data scarcity issue, exploration of data augmentation and synthetic generation techniques [26, 101] are also advised to expand available training samples with artificially constructed, domain- and distribution-aware data instances. Since the raw neuroimaging signals are represented in time-series, it is also worth investigating methods of learning over time-series data and dynamic structures which can further reduce the cost of data collection by removing the need to pre-process these signals into brain networks which are known to be time consuming.

Appendix A

Autoencoder Structure Analysis

A.1 Bridging Reconstruction Minimization and Variance Maximization

In this section, we briefly discuss how the reconstruction minimizing objective in one-layer AE can be cast to a variance-maximizing objective in PCA. Assume given a data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, its covariance matrix $\mathbf{\Sigma} = \mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{n \times n}$, and a single-layer AE projection matrix $\mathbf{W} \in \mathbb{R}^{d \times m}$ with parameters randomly initialized from the continuous uniform distribution $\mathcal{U}(0, 1)$, the reconstruction objective is:

$$\begin{aligned}
\frac{1}{n}\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^\top\|^2 &= \frac{1}{n}\mathrm{tr}((\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^\top) \\
&\quad \cdot (\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^\top)^\top) \\
&= \frac{1}{n}\mathrm{tr}((\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^\top) \\
&\quad \cdot (\mathbf{X}^\top - \mathbf{W}\mathbf{W}^\top\mathbf{X}^\top)) \\
&= \frac{1}{n}[\mathrm{tr}(\mathbf{X}\mathbf{X}^\top) - \mathrm{tr}(\mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{X}^\top) \\
&\quad - \mathrm{tr}(\mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{X}^\top) \\
&\quad + \mathrm{tr}(\mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{W}\mathbf{W}^\top\mathbf{X}^\top)]
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{n}[c_1 - 2 \cdot \mathrm{tr}(\mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{X}^\top) \\
&+ \mathrm{tr}(\hat{\mathbf{X}}\hat{\mathbf{X}}^\top)] \\
&= \frac{1}{n}[c_1 - 2 \cdot \mathrm{tr}(\mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{X}^\top) + c_2] \\
&= c_3 - c_4 \cdot \mathrm{tr}(\mathbf{W}^\top\mathbf{X}^\top\mathbf{X}\mathbf{W}) \\
&= c_3 - c_4 \cdot \mathrm{tr}(\mathbf{W}^\top\mathbf{\Sigma}\mathbf{W})
\end{aligned}
$$

Notice that c_1, c_2, c_3, c_4 are non-negative scalar constants that do not influence the overall optimization trajectory. Hence, alternatively, the optimal AE projection also maximizes the sample variance $\mathrm{tr}(\mathbf{W}^\top\mathbf{\Sigma}\mathbf{W})$, achieving an identical end goal of PCA transform. Specifically, according to PCA, variance maximization is realized by constructing the projection $\mathbf{W}$ to contain the set of orthonormal eigenvectors of $\mathbf{\Sigma}$ that gives the largest eigenvalues [35]. That is, there is an orthogonality constraint on $\mathbf{W}$. Minimizing the MSE reconstruction also results in an orthogonal $\mathbf{W}$:

$$\frac{1}{M}\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^\top\|^2 = 0 \Rightarrow \mathbf{W}\mathbf{W}^\top = \mathbf{I}$$

Therefore, the optimal AE projection $\mathbf{W}$ is also capturing a set of variance-maximizing orthogonal vectors. Note that the AE optimized $\mathbf{W}$ is theoretically equivalent to the eigendecomposition of $\mathbf{\Sigma}$ if and only if the reconstruction loss is 0. Therefore, in practice, the AE is, at best, an approximate solution to variance maximization.

A.2 Variance-based Sorting Procedure

Following the discussion in A.1, assuming a perfect optimization, the linear one-layer AE behaves similarly to PCA, and there is an equivalence relation between their respective objective functions. Notice that in PCA, the eigenvalue of the covariance matrix $\mathbf{\Sigma}$ signifies the intensity of data variation along the direction of its

corresponding eigenvector, which is essentially a column entry of the transformation matrix. Then intuitively, given an optimized AE projection $\mathbf{W}$, we can examine, for each column of $\mathbf{W}$, its representativeness (*i.e.,* data variance) of the data covariance with a scalar estimate (*i.e.,* an eigenvalue-like scoring). Inspired by the properties of eigendecomposition, we can approximate these estimates by measuring the distance of $\mathbf{W}$ w.r.t to the product of linearly transforming $\mathbf{W}$ through $\mathbf{\Sigma}$ by a scaling factor of $\boldsymbol{\lambda}$. More specifically, we want to solve for $\boldsymbol{\lambda}$ such that $\mathbf{\Sigma}\boldsymbol{w} = \boldsymbol{\lambda}\boldsymbol{w}$ for every column vector $\boldsymbol{w} \in \mathbf{W}$. Under the PCA perspective, $\boldsymbol{\lambda}$ contains the variance estimate for each column-wise individual projection of $\mathbf{W}$. To this end, we detail the sorting procedure in Algorithm 4.

Algorithm 4 Overview procedure for variance-based sorting

Input: Original feature matrix $\mathbf{X} \in \mathbb{R}^{M \times M}$; AE optimized projection matrix $\mathbf{W} \in \mathbb{R}^{M \times D}$

Initialize: Scalar vector $\boldsymbol{\lambda} \in \mathbb{R}^{D}$; Small positive float ϵ

Output: Sorted AE projection matrix $\tilde{\mathbf{W}}$

1: Normalize the feature matrix: $\mathbf{X_n} \leftarrow \mathbf{X}/\|\mathbf{X}\|$
2: Compute data covariance matrix: $\mathbf{\Sigma} \leftarrow \mathbf{X_n}^{\top}\mathbf{X_n}$
3: Solve for $\boldsymbol{\lambda}$ such that $|\mathbf{\Sigma W} - \mathbf{W} \odot \mathrm{diag}(\boldsymbol{\lambda})| \leqslant \epsilon$
4: Sort column vectors $\boldsymbol{w} \in \mathbf{W}$ according to (sorted) decreasing order of $\boldsymbol{\lambda}$ to obtain $\tilde{\mathbf{W}}$

Appendix B

Additional Experiments

B.1 Performance with GAT and GIN

Table B.1: Disease prediction performance of our framework using GAT and GIN. The best performer is highlighted in bold.

Method	BP-fMRI		BP-DTI		HIV-fMRI		HIV-DTI	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Ours w/ GCN	**68.84**$_{\pm 8.26}$	68.45$_{\pm 8.96}$	**66.57**$_{\pm 7.67}$	**68.31**$_{\pm 9.39}$	**77.80**$_{\pm 9.76}$	77.22$_{\pm 8.74}$	**67.51**$_{\pm 8.67}$	**67.74**$_{\pm 8.59}$
Ours w/ GAT	66.96$_{\pm 9.71}$	**69.68**$_{\pm 9.61}$	64.23$_{\pm 10.47}$	63.76$_{\pm 10.49}$	74.93$_{\pm 10.35}$	75.78$_{\pm 11.12}$	65.84$_{\pm 9.74}$	66.51$_{\pm 12.07}$
Ours w/ GIN	66.30$_{\pm 8.77}$	68.92$_{\pm 9.37}$	64.48$_{\pm 9.83}$	66.44$_{\pm 8.58}$	75.96$_{\pm 9.56}$	**77.63**$_{\pm 10.10}$	67.36$_{\pm 9.26}$	65.95$_{\pm 11.76}$

Table B.1 reports the downstream performance of the proposed full framework using GAT and GIN as backbone encoders. In general, the two encoders deliver inferior performance compared to GCN, which suggests that complex GNN convolutions (*e.g.,* GAT and GIN) might not be as effective as they seem when learning on bran network datasets.

B.2 Additional Ablation Studies on DTI

Figure B.1 presents the ablation studies on the DTI view following the same setup as discussed in Section 8.2. One can draw similar conclusions from the DTI-based

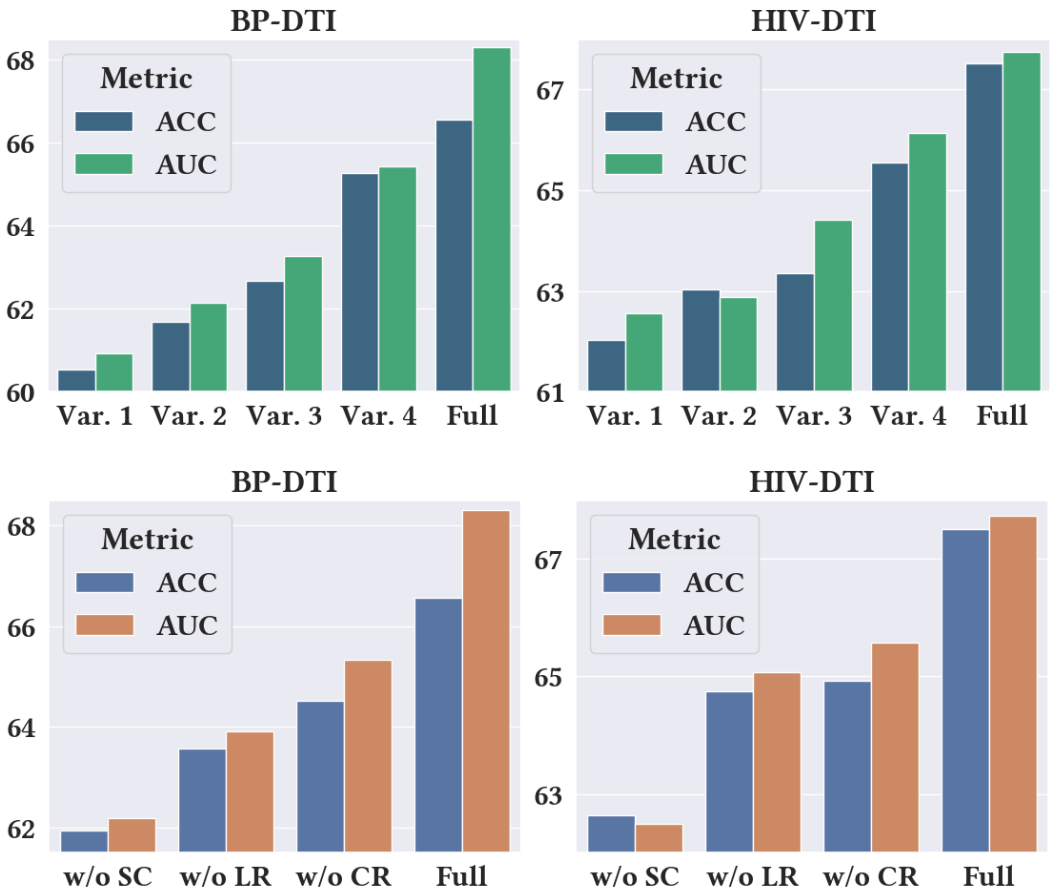


Figure B.1: Additional ablation comparisons on DTI views. The top two subfigures refer to contrastive sampling considerations and the bottom two subfigures refer to atlas mapping regularizers. The y-axis refers to the numeric values of evaluated metrics (in %). This Appendix benchmarks results on the DTI modality of the BP and HIV dataset.

analysis where each constituent component of the two-level sampling consideration as well as the atlas mapping mechanism has proven positive contribution and significance towards the overall performance and robustness.

Bibliography

[1] Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C Fowlkes, Stefano Soatto, and Pietro Perona. Task2vec: Task embedding for meta-learning. In *ICCV*, 2019.

[2] Darko Aleksovski, Dragana Miljkovic, Daniele Bravi, and Angelo Antonini. Disease progression in parkinson subtypes: the ppmi dataset. *Neurol. Sci.*, 39:1971–1976, 2018.

[3] John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *Psychological review*, 2004.

[4] Sungyong Baik, Myungsub Choi, Janghoon Choi, Heewon Kim, and Kyoung Mu Lee. Meta-learning with adaptive hyperparameters. In *NeurIPS*, 2020.

[5] Piotr Bielak, Tomasz Kajdanowicz, and Nitesh V Chawla. Graph barlow twins: A self-supervised representation learning framework for graphs. *Knowl Based Syst*, 2022.

[6] Korbinian Brodmann. *Vergleichende Lokalisationslehre der Grosshirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues*. 1909.

[7] Davide Buffelli and Fabio Vandin. A meta-learning approach for graph representation learning in multi-task settings, 2020.

[8] Bokai Cao, Liang Zhan, Xiangnan Kong, Philip S Yu, Nathalie Vizueta, Lori L Altshuler, and Alex D Leow. Identification of discriminative subgraph patterns in fmri brain networks in bipolar affective disorder. In *International Conference on Brain Informatics and Health*, 2015.

[9] Yue Cao, Zhenda Xie, Bin Liu, Yutong Lin, Zheng Zhang, and Han Hu. Parametric instance classification for unsupervised visual feature learning. *NeurIPS*, 2020.

[10] BJ Casey, Tariq Cannonier, May I Conley, Alexandra O Cohen, Deanna M Barch, Mary M Heitzeg, Mary E Soules, Theresa Teslovich, Danielle V Dellarco, Hugh Garavan, et al. The adolescent brain cognitive development (abcd) study: imaging acquisition across 21 sites. *Dev. Cogn. Neurosci.*, 32:43–54, 2018.

[11] Jatin Chauhan, Deepak Nathani, and Manohar Kaul. Few-shot learning on graphs via super-classes based on graph spectral measures. In *ICLR*, 2020.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[14] Hejie Cui, Wei Dai, Yanqiao Zhu, Xuan Kan, Antonio Aodong Chen Gu, Joshua Lukemire, Liang Zhan, Lifang He, Ying Guo, and Carl Yang. Braingb: A benchmark for brain network analysis with graph neural networks. *IEEE TMI*, 2022.

[15] Hejie Cui, Wei Dai, Yanqiao Zhu, Xiaoxiao Li, Lifang He, and Carl Yang.

Brainnnexplainer: An interpretable graph neural network framework for brain network based disease analysis. In *MICCAI*, 2022.

[16] Hejie Cui, Wei Dai, Yanqiao Zhu, Xiaoxiao Li, Lifang He, and Carl Yang. Interpretable graph neural networks for connectome-based brain disorder analysis. 2022.

[17] Luis Filipe Dehner, Mariana Spitz, and João Santos Pereira. Parkinsonism in hiv infected patients during antiretroviral therapy–data from a brazilian tertiary hospital. *Braz. J. Infect. Dis.*, 2016.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[20] Adriana Di Martino, Chao-Gan Yan, Qingyang Li, Erin Denio, Francisco X Castellanos, Kaat Alaerts, Jeffrey S Anderson, Michal Assaf, Susan Y Bookheimer, Mirella Dapretto, et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular Psychiatry*, 2014.

[21] Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph neural networks for graph classification. In *ICLR*, 2020.

[22] Patrícia R Faustino, Gonçalo S Duarte, Inês Chendo, Ana Castro Caldas, Sofia Reimão, Ricardo M Fernandes, José Vale, Michele Tinazzi, Kailash Bhatia, and Joaquim J Ferreira. Risk of developing parkinson disease in bipolar disorder: a systematic review and meta-analysis. *JAMA Neurol.*, 2020.

[23] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[24] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 2020.

[25] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, 2016.

[26] Xiaojie Guo and Liang Zhao. A systematic survey on deep generative models for graph generation. *arXiv preprint arXiv:2007.06686*, 2020.

[27] Zhichun Guo, Chuxu Zhang, Wenhao Yu, John Herr, Olaf Wiest, Meng Jiang, and Nitesh V Chawla. Few-shot graph learning for molecular property prediction. In *WWW*, 2021.

[28] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

[29] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[30] Xiaofei He, Deng Cai, Shuicheng Yan, and Hong-Jiang Zhang. Neighborhood preserving embedding. In *ICCV*, 2005.

[31] Kirsten Hilger, Makoto Fukushima, Olaf Sporns, and Christian J Fiebach. Temporal stability of functional brain modules associated with human intelligence. *Human brain mapping*, 2020.

[32] Chris Hinrichs, Vikas Singh, Lopamudra Mukherjee, Guofan Xu, Moo K Chung, Sterling C Johnson, Alzheimer's Disease Neuroimaging Initiative, et al. Spa-

tially augmented lpboosting for ad classification with evaluations on the adni dataset. *NeuroImage*, 48:138–149, 2009.

[33] Geoffrey E Hinton and Richard Zemel. Autoencoders, minimum description length and helmholtz free energy. *NeurIPS*, 1993.

[34] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *ICLR*, 2019.

[35] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *J Educ Psychol*, 1933.

[36] Yupeng Hou, Binbin Hu, Wayne Xin Zhao, Zhiqiang Zhang, Jun Zhou, and Ji-Rong Wen. Neural graph matching for pre-training graph neural networks. In *SDM*, 2022.

[37] Jinlong Hu, Lijie Cao, Tenghui Li, Shoubin Dong, and Ping Li. Gat-li: a graph attention network based learning and interpreting method for functional brain network classification. *BMC bioinformatics*, 2021.

[38] W Hu, B Liu, J Gomes, M Zitnik, P Liang, V Pande, and J Leskovec. Strategies for pre-training graph neural networks. In *ICLR*, 2020.

[39] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. 2020.

[40] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *SIGKDD*, 2020.

[41] Seongah Jeong, Xiang Li, Jiarui Yang, Quanzheng Li, and Vahid Tarokh. Dictionary learning and sparse coding-based denoising for high-resolution task

functional connectivity mri analysis. In *International Workshop on Machine Learning in Medical Imaging*, 2017.

[42] Xuan Kan, Hejie Cui, Joshua Lukemire, Ying Guo, and Carl Yang. Fbnetgen: Task-aware gnn-based fmri analysis via functional brain network generation. *MIDL*, 2022.

[43] Xuan Kan, Wei Dai, Hejie Cui, Zilong Zhang, Ying Guo, and Carl Yang. Brain network transformer. In *NeurIPS*, 2022.

[44] Jeremy Kawahara, Colin J Brown, Steven P Miller, Brian G Booth, Vann Chau, Ruth E Grunau, Jill G Zwicker, and Ghassan Hamarneh. Brainnetcnn: Convolutional neural networks for brain networks; towards predicting neurodevelopment. *Neuroimage*, 2017.

[45] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[46] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *NIPS Workshop on Bayesian Deep Learning*, 2016.

[47] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[48] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[49] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

[50] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, 2018.

[51] Xiaoxiao Li, Yuan Zhou, Nicha Dvornek, Muhan Zhang, Siyuan Gao, Juntang Zhuang, Dustin Scheinost, Lawrence H Staib, Pamela Ventola, and James S Duncan. Braingnn: Interpretable brain graph neural network for fmri analysis. *Med Image Anal*, 2021.

[52] Xiaoxiao Li, Yuan Zhou, Nicha C. Dvornek, Muhan Zhang, Siyuan Gao, Juntang Zhuang, Dustin Scheinost, Lawrence H. Staib, Pamela Ventola, and James S. Duncan. Braingnn: Interpretable brain graph neural network for fmri analysis. *Med. Image Anal.*, 2021.

[53] Martin A Lindquist. The statistical analysis of fmri data. *Stat Sci*, 23:439–464, 2008.

[54] Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. *ICLR*, 2022.

[55] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[56] Yuanfu Lu, Xunqiang Jiang, Yuan Fang, and Chuan Shi. Learning to pre-train graph neural networks. In *AAAI*, 2021.

[57] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. In *NeurIPS*, 2020.

[58] Guixiang Ma, Chun-Ta Lu, Lifang He, S Yu Philip, and Ann B Ragin. Multi-view graph embedding with hub detection for brain network analysis. In *ICDM*, 2017.

[59] Ning Ma, Jiajun Bu, Jieyu Yang, Zhen Zhang, Chengwei Yao, Zhi Yu, Sheng Zhou, and Xifeng Yan. Adaptive-step graph meta-learner for few-shot graph classification. In *CIKM*, 2020.

[60] Alireza Makhzani and Brendan Frey. K-sparse autoencoders. *ICLR*, 2014.

[61] Gustav Martensson, Joana B Pereira, Patrizia Mecocci, Bruno Vellas, Magda Tsolaki, Iwona Kłoszewska, Hilkka Soininen, Simon Lovestone, Andrew Simmons, Giovanni Volpe, et al. Stability of graph theoretical measures in structural brain networks in alzheimer's disease. *Scientific reports*, 8:1–15, 2018.

[62] Christina S Meade, Lisa A Bevilacqua, and Mary D Key. Bipolar disorder is associated with hiv transmission risk behavior among patients in treatment for hiv. *AIDS Behav.*, 2012.

[63] Tânia Novaretti, Nathália Novaretti, and Vitor Tumas. Bipolar disorder, a precursor of parkinson's disease? *Dement Neuropsychol.*, 2016.

[64] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 2010.

[65] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *SIGKDD*, 2014.

[66] Lars Philipson. Functional modules of the brain. *Journal of theoretical biology*, 2002.

[67] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *SIGKDD*, 2020.

[68] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark,

et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[69] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[70] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. 2019.

[71] Ryan A Rossi and Nesreen K Ahmed. An interactive data repository with visual analytics. *SIGKDD*, 2016.

[72] Guillaume Salha-Galvan, Johannes F. Lutzeyer, George Dasoulas, Romain Hennequin, and Michalis Vazirgiannis. Modularity-aware graph autoencoders for joint community detection and link prediction. *Neural Netw*, 2022.

[73] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *ESWC*, 2018.

[74] Rui Shi, Jian Ji, Chunhui Zhang, and Qiguang Miao. Boosting sparsity-induced autoencoder: A novel sparse feature ensemble learning for image classification. *Int. J. Adv. Robot. Syst.*, 2019.

[75] Fan-Yun Sun, Jordan Hoffman, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR*, 2019.

[76] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In *SIGKDD*, 2022.

[77] Hong Hui Tan and King Hann Lim. Review of second-order optimization techniques in artificial neural networks backpropagation. In *IOP Conf. Ser.: Mater. Sci. Eng.*, 2019.

[78] Nathalie Tzourio-Mazoyer, Brigitte Landeau, Dimitri Papathanassiou, Fabrice Crivello, Olivier Etard, Nicolas Delcroix, Bernard Mazoyer, and Marc Joliot. Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain. *NeuroImage*, 2002.

[79] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. In *ICLR*, 2020.

[80] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[81] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *ICLR*, 2018.

[82] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *ICLR*, 2019.

[83] M Vu and MT Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. In *NeurIPS*, 2020.

[84] Simon Wein, Wilhelm M Malloni, Ana Maria Tomé, Sebastian M Frank, G-I Henze, Stefan Wüst, Mark W Greenlee, and Elmar W Lang. A graph neural network framework for causal inference in brain networks. *Scientific reports*, 2021.

[85] Jun Xia, Lirong Wu, Ge Wang, Jintao Chen, and Stan Z Li. Progcl: Rethinking hard negative mining in graph contrastive learning. In *ICML*, 2022.

[86] Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. Pre-trained encyclopedia: Weakly supervised knowledge-pretrained language model. *ICLR*, 2020.

[87] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

[88] Noriaki Yahata, Jun Morimoto, Ryuichiro Hashimoto, Giuseppe Lisi, Kazuhisa Shibata, Yuki Kawakubo, Hitoshi Kuwabara, Miho Kuroda, Takashi Yamada, Fukuda Megumi, et al. A small number of abnormal brain connections predicts adult autism spectrum disorder. *Nat. Commun.*, 7:1–12, 2016.

[89] Chunde Yang, Panyu Wang, Jia Tan, Qingshui Liu, and Xinwei Li. Autism spectrum disorder diagnosis using graph attention network based on spatial-constrained sparse functional brain networks. *Computers in Biology and Medicine*, 2021.

[90] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *ICLR*, 2022.

[91] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In *NeurIPS*, 2019.

[92] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *NeurIPS*, 2018.

[93] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *NeurIPS*, 2020.

[94] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021.

[95] Shijie Zhang, Hongzhi Yin, Tong Chen, Quoc Viet Nguyen Hung, Zi Huang, and Lizhen Cui. Gcn-based user representation learning for unifying robust recommendation and fraudster detection. In *SIGIR*, 2020.

[96] Xi Zhang, Lifang He, Kun Chen, Yuan Luo, Jiayu Zhou, and Fei Wang. Multi-View Graph Convolutional Network and Its Applications on Neuroimage Analysis for Parkinson's Disease. *AMIA*, December 2018.

[97] Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. Data augmentation for graph neural networks. In *AAAI*, 2021.

[98] Zhen Zhou, Xiaobo Chen, Yu Zhang, Dan Hu, Lishan Qiao, Renping Yu, Pew-Thian Yap, Gang Pan, Han Zhang, and Dinggang Shen. A toolbox for brain network construction and classification (brainnetclass). *Hum Brain Mapp*, 2020.

[99] Qi Zhu, Carl Yang, Yidan Xu, Haonan Wang, Chao Zhang, and Jiawei Han. Transfer learning of graph neural networks with ego-graph information maximization. In *NeurIPS*, 2021.

[100] Yanqiao Zhu, Hejie Cui, Lifang He, Lichao Sun, and Carl Yang. Joint embedding of structural and functional brain networks with graph neural networks for mental illness diagnosis. In *EMBC*, 2022.

[101] Yanqiao Zhu, Yuanqi Du, Yinkai Wang, Yichen Xu, Jieyu Zhang, Qiang Liu, and Shu Wu. A survey on deep graph generation: Methods and applications. *arXiv preprint arXiv:2203.06714*, 2022.

[102] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph Contrastive Learning with Adaptive Augmentation. In *WWW*, 2021.