Development and Fast Transferring of General Connectivity-Based Diagnosis Model to New Brain Disorders with Adaptive Graph Meta-Learner

Yuxiao Liu[1], Mianxin Liu[2], Yuanwang Zhang[1], and Dinggang Shen[1,3,4](✉)

[1] School of Biomedical Engineering, ShanghaiTech University, Shanghai 201210, China
[2] Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China
[3] Shanghai United Imaging Intelligence Co., Ltd., Shanghai 200232, China
[4] Shanghai Clinical Research and Trial Center, Shanghai 201210, China
dgshen@shanghaitech.edu.cn

Abstract. The accurate and automatic diagnosis of new brain disorders (BDs) is crucial in the clinical stage. However, previous deep learning based methods require training new models with large data from new BDs, which is often not practical. Recent neuroscience studies suggested that BDs could share commonness from the perspective of functional connectivity derived from fMRI. This potentially enables developing a connectivity-based general model that can be transferred to new BDs to address the difficulty of training new models under data limitations. In this work, we demonstrate this possibility by employing the meta-learning algorithm to develop a general adaptive graph meta-learner and transfer it to new BDs. Specifically, we use an adaptive multi-view graph classifier to select the appropriate view for specific disease classification and a reinforcement-learning-based meta-controller to alleviate the over-fitting when adapting to new datasets with small sizes. Experiments on 4,114 fMRI data from multiple datasets covering a broad range of BDs demonstrate the effectiveness of modules in our framework and the advantages over other comparison methods. This work may pave the basis for fMRI-based deep learning models being widely used in clinical applications.

Keywords: Brain disorders · Graph convolutional network · Brain functional network · Few-shot adaptation · Meta-learning

1 Introduction

Brain disorders (BDs) pose severe challenges to public mental health in the global world. To understand the pathology, and for accurate diagnosis as well, functional MRI (fMRI), one of the MRI modalities, is widely studied for BDs. The fMRI provides assessments of the disease-induced changes in the brain functional connectivity networks (FCNs) among different brain regions of interest (ROIs). And a huge body of studies has successfully built effective classifiers for different BDs based on FCN and deep learning methods [6,14,15]. However, when facing new BDs, it is often needed to train a new classification model, which requires collections of large clinical data. During this process, the high costs in time, money, and labor prevent collecting sufficient

H. Greenspan et al. (Eds.): MICCAI 2023, LNCS 14227, pp. 99–108, 2023.
https://doi.org/10.1007/978-3-031-43993-3_10

data and thus the applications of deep learning models to new BDs with a small number of samples, especially for some rare BDs. Recently, there has been study [24] illustrating that BDs share significant commonness under the perspective of FCN alternations. Based on this knowledge, developing and transferring a general model to new BDs could be possible, which is promising to address the issues of building new classifiers for new BDs under data limitation.

Meta-learning based algorithm is one of the advanced methods to develop a general model based on heterogeneous information from data in different domains or for different tasks. It aims to learn optimal initial parameters for the model (**meta-learner**) which can be quickly generalized to new tasks, directly or with a few new training data for fine-tuning. There have been extensive discussions in the literature on developing more general models utilizing meta-learning [12,18,22,23] in medical fields.

Upon this evidence, it would be promising to develop a general BD diagnosis model based on FCN and further use meta-learning to solve the above-mentioned issues. However, there are still at least two challenges to achieve this goal. **First**, how to optimally extract the generalizable common knowledge (features) from the procedure of diagnosing various BDs? Previous methods focused on conventional FCNs (computed using simple linear correlations) and only treated FCN as a vector [12]. This manner of analysis *neither* properly explores topological features in the FCN, *nor* fully characterizes the complex, high-order functional interactions among brain regions, which are demonstrated to be associated with BDs [2]. **Second**, after developing a general model, how to optimally adapt the model to new datasets with different conditions? The best tuning of the model may exist only within a critical range of parameters, but previous studies blindly search for optimal parameters using *ad hoc* manual configurations, such as the adaption step size, which is easy to cause over-fitting and degrading the performance on small datasets. Theoretically, it would be beneficial to let the model adaptively configure the adaption step size and other parameters, according to the given dataset.

In this paper, we develop a novel framework (illustrated in Fig. 1) to explore the aforementioned issues. First, we assemble a large amount of data from both public datasets and in-house datasets (i.e., a total of 6 datasets, 4,114 subjects) to develop a general BD diagnosis model with meta-learning. Second, during the meta-learning procedure, we propose an adaptive multi-view graph classifier to mine topological information of low- and high-order FCNs, as different views of brain dynamics for classification. The attention mechanism is implemented to dynamically weigh and fuse different views of the information under given tasks, which collaborate with meta-learning and helps the model to learn to adapt to diagnoses for different BDs. Third, we apply a meta-controller driven by reinforcement learning [25] to choose the optimal adaption step size for the general model, for properly adapting to new BDs with small datasets and alleviating the over-fitting issue.

2 Methods

2.1 Notation and Problem Formulation

We define the entire set of BD diagnosis tasks using different datasets with $\mathcal{T}$. For a specific task $\mathcal{T}_j \in \mathcal{T}$ $(j = 1, 2, ..., 6)$, we have n pairs of FCN and labeled data $\mathcal{D} = \{(F_i, Y_i)\}_{i=1}^{n}$, where F is the fMRI data and Y is the label set of all subjects.

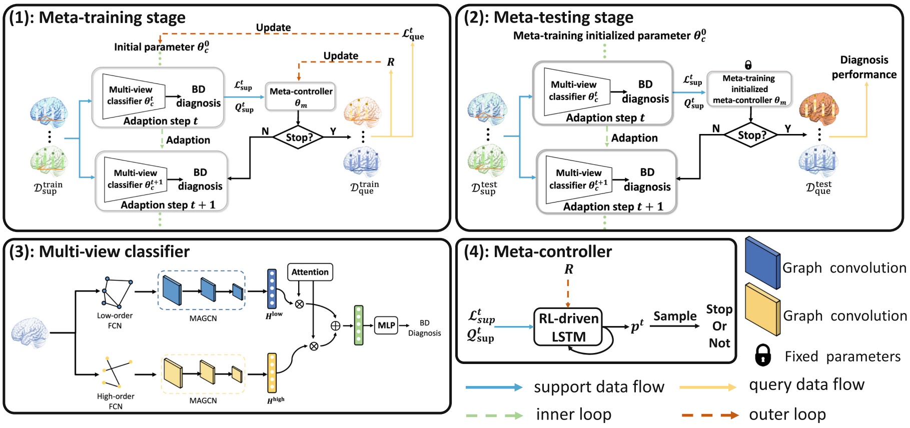
(1): Meta-training stage
Update
$\mathcal{L}_{\text{que}}^t$
Initial parameter θ_c^0
Update
R
Multi-view classifier θ_c^t
BD diagnosis
Adaption step t
$\mathcal{L}_{\text{sup}}^t$
Q_{sup}^t
Meta-controller θ_m
Adaption
N
Stop?
Y
Multi-view classifier θ_c^{t+1}
BD diagnosis
Adaption step $t+1$
$\mathcal{D}_{\text{sup}}^{\text{train}}$
$\mathcal{D}_{\text{que}}^{\text{train}}$
(2): Meta-testing stage
Meta-training initialized parameter θ_c^0
Diagnosis performance
Multi-view classifier θ_c^t
BD diagnosis
Adaption step t
$\mathcal{L}_{\text{sup}}^t$
Q_{sup}^t
Meta-training initialized meta-controller θ_m
Adaption
N
Stop?
Y
Multi-view classifier θ_c^{t+1}
BD diagnosis
Adaption step $t+1$
$\mathcal{D}_{\text{sup}}^{\text{test}}$
$\mathcal{D}_{\text{que}}^{\text{test}}$
(3): Multi-view classifier
Low-order FCN
MAGCN
H^{low}
Attention
MLP
BD Diagnosis
High-order FCN
MAGCN
H^{high}
(4): Meta-controller
R
$\mathcal{L}_{sup}^t$
Q_{sup}^t
RL-driven LSTM
p^t
Sample
Stop Or Not
Graph convolution
Graph convolution
Fixed parameters
support data flow
query data flow
inner loop
outer loop

Fig. 1. The framework of our proposed method. During the meta-training stage (1), we first tune the initial parameters θ_c^0 for T steps on the support data $\mathcal{D}_{\text{sup}}^{\text{train}}$ as the inner loop when adapting to new meta-training tasks. At the step t, meta-controller (4) decides whether to stop based on graph embedding quality Q_{sup}^t and loss $\mathcal{L}_{\text{sup}}^t$ on the support data. If the controller decides to stop at step t, then we tune θ_m and θ_c^0 as the outer loop based on loss reward $R \in \mathbb{R}^{t \times 1}$ and $\mathcal{L}_{\text{que}}^t$ separately on the query data $\mathcal{D}_{\text{que}}^{\text{train}}$ for the next inner loop. Finally, we fix θ_m, and fine-tune the meta-training stage initialized θ_c^0 on $\mathcal{D}_{\text{sup}}^{\text{test}}$ in the meta-testing stage (2), predicting the labels on $\mathcal{D}_{\text{que}}^{\text{test}}$.

ROIs are defined in individual fMRI spaces based on the brain atlas [20] to extract ROI-based fMRI signals.

We extract three features from individual fMRI data for the graph deep learning analysis. First, we compute the low-order FCN A^{low}, where $A^{\text{low}} \in \mathbb{R}^{N_{\text{ROI}} \times N_{\text{ROI}}}$ is a graph adjacency matrix containing the pair-wise correlations among different ROI signals. Second, we compute the high-order FCN A^{high} based on the topological information in A^{low} (described in detail in Sect. 2.3). These two features will be used as *edge features*. Third, we use the corresponding order adjacency matrix, together with the mean and standard deviation value of ROI signals, as the *node features* for both low- and high-order graphs [14].

We have meta-training and meta-testing stages for our meta-learner. The corresponding tasks are named as meta-training task $\mathcal{T}^{\text{train}}$ and meta-testing task $\mathcal{T}^{\text{test}}$. The meta-training stage mimics cross-task adaptations, aiming to make the model *learning to adapt to* new task $\mathcal{T}^{\text{test}}$ with a small number of samples under initial parameters. To simulate the cross-task scenario, we utilized the episodic training mechanism [13], which samples small sets across all meta-training datasets. In detail, for each round the in meta-training stage, we randomly sample the non-overlapping support data $\mathcal{D}_{\text{sup}}^{\text{train}} = \left\{\left(F_i^{\text{train}}, Y_i^{\text{train}}\right)\right\}_{i=1}^{s}$ and query data $\mathcal{D}_{\text{que}}^{\text{train}} = \left\{\left(F_i^{\text{train}}, Y_i^{\text{train}}\right)\right\}_{i=1}^{q}$ across all meta-training datasets, based on which, the $\mathcal{T}^{\text{train}}$ is constructed. With constructed $\mathcal{T}^{\text{train}}$, we have two loops to update the initial parameters of the meta-learner, which are inner and outer loops. During inner loops, initial parameters update gradients are first estimated using the back-propagation learned from the $\mathcal{D}_{\text{sup}}^{\text{train}}$. Then, if initial parameters are updated by the first gradient, we further estimate the gradients based on $\mathcal{D}_{\text{que}}^{\text{train}}$ as the outer loop to finally update them. The two loops will be combined to update the initial meta-learner parameters for better fine-tuning on new tasks as detailed in Sect. 2.2. At the meta-testing stage, we adapt the initialized parameters from the meta-training stage to the new $\mathcal{T}^{\text{test}}$ with a few data and test its performance. We first fine-tune the meta-learner on $\mathcal{D}^{\text{test}}_{\text{sup}} = \{(F_i^{\text{test}}, Y_i^{\text{test}})\}_{i=1}^{s}$, and then report classification performance on $\mathcal{D}^{\text{test}}_{\text{que}} = \{(F_i^{\text{test}}, Y_i^{\text{test}})\}_{i=1}^{q}$. Our target is to make the accuracy on $\mathcal{D}^{\text{test}}_{\text{que}}$ as high as possible when the size of the support data s is only a small portion of $\mathcal{T}^{\text{test}}$. If $s = K$, we denote the setting as K**-shot** classification.

2.2 Meta-Learner Training Algorithm

Our meta-learner consists of two modules, which are 1) multi-view classifier as detailed in Sect. 2.3 and 2) meta-controller as detailed in Sect. 2.4. The pseudo-codes for the meta-learner training algorithm are given in Algorithm 1. During the meta-training stage, our meta-learner algorithm has two iterative parameter update loops, which are inner (lines 3–9) and outer loops (lines 10–14) [5]. The inner loop aims to fast adapt the multi-view graph classifier parameterized with θ_c to the new sampled $\mathcal{T}^{\text{train}}$ under given initial parameters (θ_c^0, θ_m). During the inner loop, the meta-controller parameterized with θ_m decides whether to stop at the adaption step t to avoid over-fitting. While the outer loop aims to improve the generality of the meta-learner by exploring the optimal initial parameters (θ_c^0, θ_m) according to the loss on $\mathcal{D}^{\text{train}}_{\text{que}}$ which can easily adapt to other tasks. The initial parameter updated in the outer loop will be set for the next sampled $\mathcal{T}^{\text{train}}$. At the meta-testing stage, the meta-learner will utilize the meta-training stage initialized (θ_c^0, θ_m), fixing θ_m and fine-tuning θ_c^0 on $\mathcal{D}^{\text{test}}_{\text{sup}}$, finally predicting the label $\mathcal{D}^{\text{test}}_{\text{que}}$.

2.3 Multi-view Graph Classifier $\boldsymbol{\theta_c}$

As mentioned above, exploring the topological information of the FCN is necessary for BD diagnosis. However, a single low-order FCN view can only illustrate the simple pair-wise relation. So, we additionally construct the high-order FCN, which reflects the correlation among ROIs in terms of their own FC patterns, as calculated below:

$$A_{ij}^{\text{high}} = \text{Pearson Correlation}(A_{i*}^{\text{low}}, A_{j*}^{\text{low}}). \tag{1}$$

After constructing high-order adjacency matrix A^{high} as a complementary view, we input them into the **multi-view graph classifier** parameterized with θ_c^t at the adaption step t. To extract the disease-related features of different-view graphs, we use convolution in GCN [11] to aggregate the neighboring node features.

Then, we use the pooling operation to further extract the global topological features of different graph views. Here, we opt for gPOOL operation [7] for its parameter-saving and ability to extract global topological features. For each pooling stage, it acquires the importance of each node by calculating the inner product between the node feature vectors and a learnable vector. The top-k important nodes and their corresponding subgraph will be sampled for the following graph convolutions. By iteratively repeating the node feature aggregation and pooling illustrated in Fig. 1 (3), we can finally acquire

Algorithm 1: Meta-training algorithm.

```
Input: learning rates: $\beta_1$, $\beta_2$, random initialization $(\theta_c^*, \theta_m^*)$
Output: Trained parameters $\left(\theta_c^0, \theta_m\right)$.
$\left(\theta_c^0, \theta_m\right) = (\theta_c^*, \theta_m^*)$
while not done do
    Sample $\mathcal{T}^{\text{train}}$ with $\mathcal{D}_{\text{sup}}^{\text{train}}$ and $\mathcal{D}_{\text{que}}^{\text{train}}$ from meta-training datasets
    while $t \in [1, T_{\text{MAX}}]$ do
        Calculate $\nabla_{\theta_c^{t-1}} \mathcal{L}_{\text{sup}}^t$ on $\mathcal{D}_{\text{sup}}^{\text{train}}$
        $\theta_c^t \leftarrow \theta_c^{t-1} - \beta_1 \nabla_{\theta_c^{t-1}} \mathcal{L}_{\text{sup}}^t$
        Calculate $Q_{\text{sup}}^t$ by Eq. 3 on $\mathcal{D}_{\text{sup}}^{\text{train}}$
        Calculate stop probability by $Q_{\text{sup}}^t$ and $\mathcal{L}_{\text{sup}}^t$ and determine whether to stop at step
        $t$ via Eq. 4, 5
        Calculate reward $R^t$ with loss changes on $\mathcal{D}_{\text{que}}^{\text{train}}$ via Eq. 7
    end
    $\theta_c^0 \leftarrow \theta_c^0 - \beta_1 \nabla_{\theta_c^0} \mathcal{L}_{\text{que}}^T$ on $D_{\text{que}}^{\text{train}}$
    for $t = 0 : T$ do
        $\theta_m \leftarrow \theta_m + \beta_2 R^t \nabla_{\theta_m} \ln p(t)$
    end
end
```

the representation of the graph, denoted as H. Furthermore, to let the model adaptively choose the proper view for classification, we apply an attention-based mechanism to aggregate the learned embeddings by feeding the concatenation of the learned representations into an attention module as below:

$$\alpha = \text{Softmax}\left(\text{ReLU}\left(\left[H^{\text{low}} \| H^{\text{high}}\right] W_1\right) W_2\right). \tag{2}$$

The attention mechanism allows the model to decide which view should rely on for specific tasks more adaptively. Then, we forward attention-weighted features into an MLP and acquire the final prediction label. Here, we use the cross-entropy loss as a constraint to the network.

2.4 Meta-Controller θ_m

In machine learning, over-fitting is one of the critical problems that restrict the performance of models, especially in small datasets. Previous works utilize the early stopping to alleviate this problem, but the hand-crafted early stopping parameter is hard to choose. Here, we utilize a neural network to learn the stop policy adaptively, which we call **meta-controller** parameterized with θ_m depicted in Fig. 1 (4). The meta-controller uses a reinforcement learning based algorithm [17] to decide optimal adaption step sizes for cross-task adaptions. Considering the characteristics of the graph data, we let the model determine when to stop *not only* according to the classification loss, *but also* the graph embedding quality on the support data. For the embedding quality, we use the Average Node Information (ANI), denoted as Q, to measure it. It represents how a node can be measured by the neighboring nodes. A high ANI value indicates that the embedding module has learned the most information about the graph. If we keep aggregating the nodal features by graph convolution when the ANI is high, the over-smoothing will happen, making all nodes have similar features and thus degrading the performance. We define the ANI Q_{sup} of the support data within $\mathcal{T}^{\text{train}}$ with the L1 norm [9] as follows:

$$Q_i = \frac{1}{N_{\text{ROI}}} \sum_{j=1} \left\| \left[\left(I - {D_i}^{-1} A_i \right) X_i^L \right]_j \right\|_1, \; Q_{\text{sup}} = 1/s \times \sum_{i=1}^{s} \left(Q_i^{\text{high}} + Q_i^{\text{low}} \right), \tag{3}$$

where D_i X_i^L represent the degree matrix of A_i and the feature matrix in the last layer L, respectively; j denotes j-th node which is also the j-th row in those matrices, and $\| \cdot \|_1$ denotes the L1-norm of row vector.

For classification losses $\mathcal{L}_{\text{sup}}$ and ANI values Q_{sup} on $\mathcal{D}_{\text{sup}}^{\text{train}}$ across t adaption steps, we use them to compute the stop probability p^t at step t with an LSTM [8] model by considering the temporal information as follows:

$$o^t = \text{LSTM} \left(\left[Q_{\text{sup}}^t, \mathcal{L}_{\text{sup}}^t \right], o^{t-1} \right), p^t = \sigma \left(W o^t + b \right), \tag{4}$$

where o^t is the output of the LSTM model at step t and σ is the SoftMax function. Finally, we sample the choices c^t by Bernoulli distribution to decide whether we should stop at step t.

$$c^t \sim Bernouli(p^t) = \begin{cases} 1, & \text{stop the adaption} \\ 0, & \text{keep adaption} \end{cases}. \tag{5}$$

Since the relation between θ_m and θ_c is undifferentiable, it is impossible to take direct gradient descent on θ_m. We use stochastic policy gradient to optimize θ_m. Once we sample the stop choice at step T, we train the meta-controller according to loss changes on $\mathcal{D}_{\text{que}}^{\text{train}}$ across T steps. Given the parameter update trajectory of classifier $\{\theta_c^0, \theta_c^1, ...\theta_c^T\}$ during T steps, we calculate the corresponding loss change trajectory of $\mathcal{D}_{\text{que}}^{\text{train}}$. Based on that, we further define the controller immediate rewards r at step t as the loss change on $\mathcal{D}_{\text{que}}^{\text{train}}$ (caused by parameter update of step t):

$$r^{(t)} = \mathcal{L} \left(\mathcal{D}_{\text{que}}^{\text{train}}; \theta_c^{t-1} \right) - \mathcal{L} \left(\mathcal{D}_{\text{que}}^{\text{train}}; \theta_c^t \right) \tag{6}$$

Then, the accumulative reward R at step t is

$$R^t = \sum_{i=t}^{T} r^i = \mathcal{L} \left(\mathcal{D}_{\text{que}}^{\text{train}}; \theta_c^{t-1} \right) - \mathcal{L} \left(\mathcal{D}_{\text{que}}^{\text{train}}; \theta_c^T \right), \tag{7}$$

where T is the total number of steps and R^t is the change of classification loss on $\mathcal{D}_{\text{que}}^{\text{train}}$ from step t to the end of adaption. Then we update our meta-controller by policy gradients, which is a typical method in Reinforcement learning [25]:

$$\theta_m \leftarrow \theta_m + \beta_2 R^t \nabla_{\theta_m} \ln \left(p^t \right), \tag{8}$$

where∇_{θ_m} is the gradients over θ_m and β_2 is the learning rate.

3 Experiments

3.1 Dataset

We use fMRI meta-training data from five datasets including Alzheimer's Disease Neuroimaging Initiative (ADNI) [1,10], Open Access Series of Imaging Studies (OASIS) [19], and in-house dataset from Huashan Hospital (elder BDs datasets); ADHD-200 [3] and Autism Brain Imaging Data Exchange (ABIDE) [4] (youth BD datasets). For the meta-testing dataset, we use the in-house dataset from Zhongshan Hospital which is about vascular cognitive impairment (VCI). All datasets are shown in Table 1 The details of image acquisition parameters and processing procedures can be found in [16].

Table 1. Dataset for experiments

Dataset	ADNI	OASIS	ABIDE	ADHD-200	Huashan	Zhongshan
Disease	MCI, AD	AD	Autism	ADHD	MCI, AD	VCI
BD	785	83	499	280	100	151
HC	566	634	512	488	167	246
Total	1351	717	1011	768	267	397

3.2 Settings

To ensure a fair comparison, we use three graph convolutional layers, followed by corresponding pooling layers, for all GNN based methods. We use the ADAM optimizer with 1e-4 for learning rate and 5e-4 for weight decay, 100 for epochs, 0.001 for both β_1 and β_2, respectively. For the inner loop fast adaption, we set the minimum and maximum steps by 4 and 16. In the meta-training stage, we sample $\mathcal{D}_{\text{sup}}^{\text{train}}$ and $\mathcal{D}_{\text{que}}^{\text{train}}$ from all training datasets; and, in the meta-testing stage, we only randomly sample $\mathcal{D}_{\text{sup}}^{\text{test}}$ and $\mathcal{D}_{\text{que}}^{\text{test}}$ from the meta-testing dataset. The size of support data for both meta-training and meta-testing stages is depicted in the first line of Table 2, and we set the size of the query data as 256 for two stages. For different datasets, we only diagnose whether they are BD or not. We randomly select the support and the query data five times to report the mean and standard deviation value of the performance.

3.3 Results and Discussions

In Table 2, we compare our method with two SOTA meta-learning based methods [12,17] (in lines 3 and 4) under different K-shot configurations. It can be observed that, when the size of the support data increases, the performances of all methods increase and our proposed method outperforms the SOTA methods in terms of all performance metrics. We also validate the effectiveness of the use of multiple-BD datasets for training our model. When compared with the model without the meta-training stage (line

Table 2. Comparison of different methods, with bold denoting the highest performance. The proposed method shows statistically significant improvements (in level of p-value $<$ 0.05) over all compared methods in terms of all metrics.

1		10-shot				30-shot				50-shot			
2	methods	Acc	Sen	Spe	AUC	Acc	Sen	Spe	AUC	Acc	Sen	Spe	AUC
3	Jaein	$52.6_{\pm 5.7}$	$52.7_{\pm 6.0}$	$55.2_{\pm 5.3}$	$55.8_{\pm 3.7}$	$58.8_{\pm 4.1}$	$59.3_{\pm 3.9}$	$56.8_{\pm 4.9}$	$59.7_{\pm 4.7}$	$63.3_{\pm 4.9}$	$65.2_{\pm 5.2}$	$61.3_{\pm 4.1}$	$63.9_{\pm 5.0}$
4	Ning	$54.3_{\pm 5.1}$	$53.1_{\pm 5.5}$	$56.7_{\pm 6.0}$	$55.4_{\pm 4.3}$	$60.7_{\pm 4.3}$	$61.7_{\pm 5.7}$	$60.3_{\pm 6.1}$	$61.4_{\pm 4.3}$	$65.1_{\pm 5.1}$	$64.7_{\pm 4.7}$	$66.3_{\pm 6.1}$	$65.7_{\pm 5.5}$
5	Ours (w/o pre-training)	$50.4_{\pm 6.3}$	$47.0_{\pm 5.7}$	$53.9_{\pm 6.2}$	$51.3_{\pm 4.9}$	$54.1_{\pm 4.3}$	$52.8_{\pm 4.4}$	$55.1_{\pm 5.3}$	$52.1_{\pm 4.9}$	$58.4_{\pm 4.7}$	$59.9_{\pm 4.9}$	$56.8_{\pm 5.2}$	$58.1_{\pm 6.3}$
6	Ours (w/o multi-view classifier)	$54.7_{\pm 5.8}$	$53.1_{\pm 5.5}$	$55.2_{\pm 6.1}$	$54.4_{\pm 3.0}$	$59.9_{\pm 5.7}$	$59.1_{\pm 4.7}$	$57.3_{\pm 5.4}$	$57.6_{\pm 3.7}$	$65.3_{\pm 4.9}$	$64.6_{\pm 6.0}$	$66.1_{\pm 7.1}$	$64.9_{\pm 3.8}$
7	Ours (w/o meta-controller)	$53.8_{\pm 6.7}$	$51.0_{\pm 7.7}$	$54.9_{\pm 5.2}$	$54.4_{\pm 3.9}$	$60.1_{\pm 7.1}$	$52.8_{\pm 6.2}$	$55.1_{\pm 8.3}$	$57.1_{\pm 4.4}$	$66.4_{\pm 7.7}$	$66.9_{\pm 4.3}$	$65.8_{\pm 7.6}$	$64.1_{\pm 6.6}$
8	Ours (w/o elder BD)	$53.3_{\pm 5.2}$	$52.7_{\pm 7.7}$	$54.9_{\pm 5.9}$	$55.7_{\pm 5.3}$	$57.8_{\pm 4.5}$	$57.6_{\pm 3.7}$	$58.1_{\pm 4.7}$	$58.8_{\pm 6.1}$	$62.4_{\pm 5.0}$	$59.8_{\pm 4.9}$	$66.9_{\pm 5.8}$	$64.4_{\pm 5.4}$
9	Ours (w/o youth BD)	$55.8_{\pm 4.9}$	$55.0_{\pm 5.7}$	$56.4_{\pm 6.3}$	$57.4_{\pm 4.8}$	$59.9_{\pm 5.7}$	$59.1_{\pm 5.7}$	$57.3_{\pm 6.0}$	$56.6_{\pm 5.5}$	$65.3_{\pm 5.3}$	$64.6_{\pm 4.9}$	$66.1_{\pm 6.5}$	$64.9_{\pm 6.3}$
10	Ours	$\mathbf{59.8}_{\pm 4.8}$	$\mathbf{58.0}_{\pm 4.3}$	$\mathbf{60.1}_{\pm 4.1}$	$\mathbf{59.4}_{\pm 4.0}$	$\mathbf{67.9}_{\pm 3.7}$	$\mathbf{66.1}_{\pm 4.2}$	$\mathbf{68.3}_{\pm 3.3}$	$\mathbf{67.6}_{\pm 3.7}$	$\mathbf{71.3}_{\pm 3.3}$	$\mathbf{70.7}_{\pm 3.2}$	$\mathbf{72.1}_{\pm 3.5}$	$\mathbf{70.0}_{\pm 3.6}$

5) or reducing either elder or youth BD datasets (lines 8 and 9), we find that the performances all drop significantly. This can validate that *not only* the information from MCI/AD, which is more similar to VCI, is useful, *but also* the general BD commonness suggested by data of youth BDs is beneficial. For the ablation study, we test the performance without a multi-view graph classifier or meta-controller as shown in Table 2 (lines 6 and 7) and Fig. 2 (a). Figure 2 (a) demonstrates that, when the epoch increases, the accuracy of the model assisted with both classifier and controller acquires steady and continuous improvement. In addition, we investigate the associations among ANI, support data accuracy and stop probability. As we can see from Figs. 2(b) and 2(c)), with the increment of ANI (red curve in Fig. 2(b)) and accuracy (green curve in Fig. 2(c)) on the support data, the stop probability also begins to increase (purple curves in Figs. 2(b) and 2(c)) to alleviate over-fitting in the query data, which supports the effectiveness of the meta-controller.

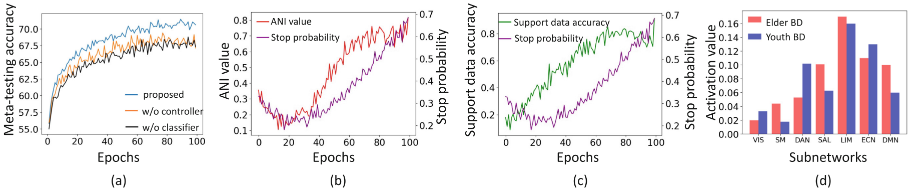

Fig. 2. (a) Accuracy curves; (b) ANI and stop probability curves; (c) Support data accuracy and stop probability curves; (d) Activation values of different subnetworks. (Color figure online)

Finally, we visualize the predictive importance of different resting-state networks, including visual network (VIS), somatomotor network (SM), dorsal attention network (DAN), salience network (SAL), limbic network (LIM), executive control network (ECN) and default mode network (DMN) when adapting to elder BD and younger BD as shown in Fig. 2 (d) by GradCAM [21]. In the results, the LIM consistently shows importance when diagnosing elder and younger BDs, which is in line with previous neuroscience studies [24]. This validates that our proposed method can properly detect meaningful common features among different BDs.

4 Conclusion

In this work, we focus on the issues of developing a classifier on new BD datasets with small samples and propose a novel framework. A broad of datasets covering elder and youth BDs are used to train the model to estimate the common features among BDs. An adaptive multi-view graph classifier is proposed to enable the model efficiently extract features for different BD diagnosis tasks. In addition, to avoid over-fitting during the adaptions to new data, we utilize a novel meta-controller driven by RL. Extensive experiments demonstrate the effectiveness and generalization of our proposed method. It is expected that advanced graph embedding methods can be integrated into our framework to improve performance. Our work is also promising to be extended to neuroscience studies to reveal both common and unique characteristics of different BDs.

Acknowledgment. This work was supported in part by National Natural Science Foundation of China (grant number 62131015), Science and Technology Commission of Shanghai Municipality (STCSM) (grant number 21010502600), and The Key R&D Program of Guangdong Province, China (grant number 2021B0101420006).

References

1. Aisen, P.S., et al.: Alzheimer's disease neuroimaging initiative 2 clinical core: progress and plans. Alzheimer's Dementia **11**(7), 734–739 (2015)
2. Chen, X., et al.: High-order resting-state functional connectivity network for MCI classification. Hum. Brain Mapp. **37**(9), 3282–3296 (2016)
3. ADHD-200 Consortium: The ADHD-200 consortium: a model to advance the translational potential of neuroimaging in clinical neuroscience. Front. Syst. Neurosci. **6**, 62 (2012)
4. Di Martino, A., et al.: The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. Mol. Psychiatry **19**(6), 659–667 (2014)
5. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning, pp. 1126–1135. PMLR (2017)
6. Gadgil, S., Zhao, Q., Pfefferbaum, A., Sullivan, E.V., Adeli, E., Pohl, K.M.: Spatio-temporal graph convolution for resting-state fMRI analysis. In: Martel, A.L., et al. (eds.) MICCAI 2020. LNCS, vol. 12267, pp. 528–538. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59728-3_52
7. Gao, H., Ji, S.: Graph U-nets. In: International Conference on Machine Learning, pp. 2083–2092. PMLR (2019)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Hou, Y., et al.: Measuring and improving the use of graph information in graph neural networks. arXiv preprint arXiv:2206.13170 (2022)
10. Jack, C.R., Jr., et al.: Magnetic resonance imaging in Alzheimer's disease neuroimaging initiative 2. Alzheimer's Dementia **11**(7), 740–756 (2015)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2017)
12. Lee, J., Kang, E., Jeon, E., Suk, H.-I.: Meta-modulation network for domain generalization in multi-site fMRI classification. In: MICCAI 2021. LNCS, vol. 12905, pp. 500–509. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87240-3_48

13. Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.Z., Hospedales, T.M.: Episodic training for domain generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1446–1455 (2019)
14. Li, X., et al.: BrainGNN: interpretable brain graph neural network for fMRI analysis. Med. Image Anal. **74**, 102233 (2021)
15. Liu, M., et al.: Multiscale functional connectome abnormality predicts cognitive outcomes in subcortical ischemic vascular disease. Cereb. Cortex **32**(21), 4641–4656 (2022)
16. Liu, M., et al.: Deep learning reveals the common spectrum underlying multiple brain disorders in youth and elders from brain functional networks. arXiv preprint arXiv:2302.11871 (2023)
17. Ma, N., et al.: Adaptive-step graph meta-learner for few-shot graph classification. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, pp. 1055–1064 (2020)
18. Mahajan, K., Sharma, M., Vig, L.: Meta-DermDiagnosis: few-shot skin disease identification using meta-learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2020)
19. Marcus, D.S., Wang, T.H., Parker, J., Csernansky, J.G., Morris, J.C., Buckner, R.L.: Open Access Series of Imaging Studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. J. Cogn. Neurosci. **19**(9), 1498–1507 (2007)
20. Schaefer, A., et al.: Local-global parcellation of the human cerebral cortex from intrinsic functional connectivity MRI. Cereb. Cortex **28**(9), 3095–3114 (2018)
21. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626 (2017)
22. Singh, R., Bharti, V., Purohit, V., Kumar, A., Singh, A.K., Singh, S.K.: MetaMed: few-shot medical image classification using gradient-based meta-learning. Pattern Recognit. **120**, 108111 (2021)
23. Sun, L., et al.: Few-shot medical image segmentation using a global correlation network with discriminative embedding. Comput. Biol. Med. **140**, 105067 (2022)
24. Taylor, J.J., et al.: A transdiagnostic network for psychiatric illness derived from atrophy and lesions. Nat. Hum. Behav. **7**(3), 420–429 (2023)
25. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. **8**, 229–256 (1992). https://doi.org/10.1007/BF00992696