Computer Methods and Programs in Biomedicine 257 (2024) 108419

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: https://www.sciencedirect.com/journal/computer-methods-and-programs-in-biomedicine

# Optimizing graph neural network architectures for schizophrenia spectrum disorder prediction using evolutionary algorithms

Shurun Wang [a,b,c], Hao Tang [b,h], Ryutaro Himeno [c], Jordi Solé-Casals [d,f], Cesar F. Caiafa [e], Shuning Han [d,g], Shigeki Aoki [c], Zhe Sun [c,*]

[a] School of Information Science and Technology, University of Science and Technology of China, Hefei, 230027, China
[b] School of Electrical Engineering and Automation, Hefei University of Technology, Hefei, 230009, China
[c] Graduate School of Medicine, Juntendo University, Tokyo, 1138421, Japan
[d] Data and Signal Processing Research Group, University of Vic—Central University of Catalonia, Vic, 08500, Spain
[e] Instituto Argentino de Radioastronomía-CONICET CCT La Plata/CIC-PBA/UNLP, V. Elisa, 1894, Argentina
[f] Department of Psychiatry, University of Cambridge, Cambridge, CB2 3EB, United Kingdom
[g] Image Processing Research Group, RIKEN Center for Advanced Photonics, RIKEN, Wako-Shi, Saitama, 351-0198, Japan
[h] Industrial Automation Engineering Technology Research Center of Anhui Province, Hefei, 230009, China

## ARTICLE INFO

*Keywords:*
Graph neural network
Graph neural architecture search
Evolutionary algorithm
Schizophrenia spectrum disorder
Brain functional connectivity

## ABSTRACT

**Background and Objective:** The accurate diagnosis of schizophrenia spectrum disorder plays an important role in improving patient outcomes, enabling timely interventions, and optimizing treatment plans. Functional connectivity analysis, utilizing functional magnetic resonance imaging data, has been demonstrated to offer invaluable biomarkers conducive to clinical diagnosis. However, previous studies mainly focus on traditional machine learning methods or hand-crafted neural networks, which may not fully capture the spatial topological relationship between brain regions.

**Methods:** This paper proposes an evolutionary algorithm (EA) based graph neural architecture search (GNAS) method. EA-GNAS has the ability to search for high-performance graph neural networks for schizophrenia spectrum disorder prediction. Moreover, we adopt GNNExplainer to investigate the explainability of the acquired architectures, ensuring that the model's predictions are both accurate and comprehensible.

**Results:** The results suggest that the graph neural network model, derived using genetic algorithm search, outperforms under five-fold cross-validation, achieving a fitness of 0.1850. Relative to conventional machine learning and other deep learning approaches, the proposed method yields superior accuracy, F1 score, and AUC values of 0.8246, 0.8438, and 0.8258, respectively.

**Conclusion:** Based on a multi-site dataset from schizophrenia spectrum disorder patients, the findings reveal an enhancement over prior methods, advancing our comprehension of brain function and potentially offering a biomarker for diagnosing schizophrenia spectrum disorder.

## 1. Introduction

Schizophrenia spectrum disorder (SSD) is a serious mental illness that ranks among the leading causes of disability and affects more than twenty million individuals globally [1]. Individuals diagnosed with SSD often face challenges in integrating with their communities. They have a low quality of life due to the severe impairment in social and occupational functioning, often exacerbated by persistent delusions, hallucinations, and cognitive deficits [2]. An accurate diagnosis of SSD allows patients to receive timely and appropriate treatment and support, improving their quality of life, alleviating social functional barriers, and reducing the psychological and economic burdens on families. Therefore, developing an accurate SSD diagnostic method has significant clinical importance and also brings positive benefits to societal health and well-being.

In recent years, research on SSD has concentrated on understanding the underlying brain abnormalities and mechanisms through morphological and neurobiological characteristics. Several prevalent neuroimaging techniques, including functional magnetic resonance imaging (fMRI), magnetoencephalography (MEG), positron emission tomography (PET), and magnetic resonance spectroscopy (MRS), are utilized for brain structure and function analysis in individuals with SSD [3–5]. Among them, resting-state fMRI (rs-fMRI) is primarily used to

[*] Corresponding author.
*E-mail addresses:* wangsr@mail.hfut.edu.cn (S. Wang), z.sun.kc@juntendo.ac.jp (Z. Sun).

https://doi.org/10.1016/j.cmpb.2024.108419
Received 26 November 2023; Received in revised form 1 September 2024; Accepted 8 September 2024
Available online 11 September 2024

measure blood oxygen level-dependent signals in the brain at rest. This technique does not require any specific task stimulation but records brain activity in its resting state. Researchers have found that brain functional connectivity (FC) constructed using rs-fMRI can distinguish between the SSD group and the healthy group [6–8]. Hence, how to design robust classification methods based on brain FC is a topic of interest for many researchers currently.

For over a decade, traditional machine learning (ML) techniques have played a pivotal role in brain network classification. The support vector machine (SVM), a supervised technique, is the most prevalently used ML method in this field. Ma et al. employed SVM to classify 44 schizophrenia patients and 40 healthy controls, achieving an accuracy of 0.8137 [6]. Ramkiran et al. employed the anticorrelation after mean of Antilog (AMA) method to simplify the fully connected FC matrices. Subsequently, they utilized SVM for classifying 112 subjects, achieving an average accuracy of 0.74 [9]. Gao et al. assessed the utility of the amplitude of low-frequency fluctuation (ALFF) via SVM as a means of diagnosing SSD [10]. Besides SVM, other ML methods, such as logistic regression (LR) and random forests (RF), are also mentioned in some SSD prediction approaches. Srinivasagopalan et al. utilized independent component analysis for feature selection and subsequently integrated it with the LR and RF techniques. In the classification of 75 controls and 69 patients, they achieved accuracies of 0.8277 and 0.8333, respectively [11]. In [12], features were extracted from the gray matter volume (GMV) and ALFF. Then, the Xgboost classifier, combined with the information fusion method, was employed to identify 38 healthy controls, 16 deficit schizophrenic patients, and 31 non-deficit schizophrenic patients. These examples exhibit the practical applications of various ML methods. However, the quality and relevance of feature selection profoundly impact the performance of ML, and biased feature extraction can lead to imprecise class discrimination [13,14].

Unlike traditional ML techniques, deep learning (DL) obviates the need for complex feature engineering and strives to establish an end-to-end model for data processing and prediction. It has garnered increasing popularity for diverse biomedical applications, including medical image analysis and disease prediction [15,16]. The BrainNetCNN, proposed in [17], is a convolutional neural network (CNN) framework designed for predicting clinical neurodevelopmental outcomes based on brain networks. This framework leverages the topological locality inherent in structural brain networks and has demonstrated effectiveness in recent biomedical engineering approaches [18–20]. Moreover, brain FC is essentially the graph-structured data, where nodes signify brain regions and edges denote either the structural or functional link between these regions. Therefore, several studies have employed the graph neural network (GNN), a powerful DL model capable of generating embeddings for a node by aggregating features from its neighboring nodes, for tasks such as node classification, graph classification, or link prediction. Chen et al. utilized a graph convolutional network (GCN) to classify schizophrenia patients based on brain region and connectivity features extracted from a combined functional MRI and connectomics analysis [21]. Yu et al. proposed an improved graph attention network (GAT) with the bilinear convolution for diagnosis of schizophrenia [22]. GNN-based methods are also been employed in the diagnosis of other diseases, including autism spectrum disorder (ASD) and major depressive disorder (MDD) [23,24]. Han et al. proposed the FCbasedGCN model for the early prediction of dementia using fMRI data [25]. Zhang et al. proposed the LGGNN model to achieve the precise classification of ASD [26]. Although researchers can optimize the network to enhance recognition accuracy, it is necessary to test the effects of a large number of GNN components (such as convolutional layers, aggregators, and activation functions). This requires researchers to have a deep understanding of network architecture design and a substantial time investment. Thus, how to effectively determine the proper GNN architecture from numerous design choices presents a significant challenge.

Neural architecture search (NAS) can automatically seek high-performance deep neural networks (DNN), and it has achieved remarkable advancements in many fields such as computer vision [27] and natural language processing [28] in recent years. Unlike NAS, GNAS is designed to generate the optimal GNN model for extracting features from graph-structured data. Through the establishment of a predefined search space, GNAS enables the efficient exploration of possible architectures, the assessment of their performance, and the systematic refinement of the search process grounded in acquired knowledge, ultimately yielding the ideal model. To date, most of the GNAS methods are based on reinforcement learning (RL) and EA. RL-based GNAS methods (e.g., GraphNAS [29], Auto-GNN [30]) generally require training a controller to generate component strings describing the structure of GNN. However, training both the controller and the GNN model simultaneously requires more computational time. Additionally, the controller typically generates candidate GNN models in a sequential manner, making it difficult to scale to a large search space and perform parallel model evaluations. EA-based GNAS methods employ an iterative process whereby individuals (i.e., GNN models) are selected from an initialized population and evaluated according to a fitness function. Subsequently, a new population is generated using the best-performing individual from the preceding generation. Since all individual models in the population are independent, EA-based GNAS methods can be easily scaled to a large search space and models can be evaluated simultaneously. AGNAS [31] employs a genetic algorithm for the GNAS process and evaluates individuals in parallel, yet it does not treat the optimizer as a hyper-parameter in the search space. Genetic-GNN [32] includes the optimizer in the search space, but the GNN search process remains limited by the genetic algorithm framework. There are certain obstacles when attempting to extend to other advanced evolutionary algorithms.

To address the aforementioned problems, this paper proposes a novel GNAS framework, named EA-GNAS. It identifies the optimal model through a search within a vast GNN architecture space, and the model is applied for SSD prediction. Different from the exiting GNAS methods that utilize the specific evolutionary algorithm, EA-GNAS is designed to be generic and adaptable to various evolutionary algorithms, enabling a more flexible and comprehensive exploration of the GNN architecture space. Unlike existing SSD prediction methods, the SSD prediction model proposed in this paper is obtained through an automated search process, rather than traditional manual design. The advanced GNAS technology allows for the automatic exploration of the model structure most suitable for specific tasks, which not only reduces the professional knowledge and time cost required to design effective models but also discovers innovative solutions that might be overlooked by manual design methods in a broad architectural space. Moreover, although many studies use DL models to predict SSD, they usually only report the performance of the model without detailing which features within brain FC significantly influence the prediction outcomes. Therefore, to achieve more accurate SSD predictions and a clearer understanding of the underlying factors, it is necessary to develop a robust GNN model and undertake explainability analysis for the model. The principle contributions of this paper are listed as follows:

(1) We propose a novel generic evolutionary GNAS framework named EA-GNAS, designed to automatically identify high-performance GNN architectures from a well-defined search space. Within this framework, six popular evolutionary algorithms are compared to evaluate their search performance in the task of SSD prediction.

(2) We utilize a GNN explainability analysis method, GNNExplainer, to conduct in-depth exploration and explanation of the searched GNN model. Through this approach, the most critical brain region features can be identified, allowing for a detailed understanding of the neural mechanisms that significantly contribute to SSD prediction.

(3) In comparison with four ML-based and six DL-based methods tested on multi-site SSD datasets, our proposed method demonstrates superior performance across several metrics. The source code is publicly available on GitHub at https://github.com/Shurun-Wang/EA-GNAS.

## 2. Problem definition

The purpose of this paper is to propose a framework for searching graph neural network architectures based on evolutionary algorithms, and to apply the discovered optimal network to predict SSD. Additionally, we employ a graph-based explainability method to explain the prediction results. Therefore, the problem definition can be categorized into three aspects: GNN-based disease prediction, GNAS, and graph explainability.

**Definition 1** (*GNN-based Disease Prediction*)**.** The objective of GNN-based disease prediction is to classify the specific property of interest for an entire graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_i\}_{i=1,2,\ldots,V}$ is a set of $V$ unique nodes and $\mathcal{E} = \{e_{i,j}\}_{i,j=1,2,\ldots,V;i\neq j}$ is a set of edges. Specifically, edges can be represented by a binary adjacency matrix of size $V \times V$, where $A_{i,j} = 1$ when $e_{i,j} \in \mathcal{E}$, otherwise $A_{i,j} = 0$. In addition, $\mathcal{F} \in \mathbb{R}^{V\times F}$ is a feature matrix containing all $V$ nodes with their associate $F$ features. In this work, the brain function connectivity of each subject forms a graph, with each brain region represented as a node, the connection between two brain regions represented as an edge, and the correlations between a brain region and other brain regions serving as node features. Therefore, for $M$ subjects, we obtain the input space of graphs $\{\mathcal{G}_m\}_{m=1,2,\ldots,M}$ and a set of class labels $\mathcal{Y}$ (i.e., disease types). The objective of the classification task is to learn a mapping function $\boldsymbol{p} : \{\mathcal{G}_m\}_{m=1,2,\ldots,M} \rightarrow \mathcal{Y}$, and $\boldsymbol{p}$ is the GNN model derived by GNAS.

**Definition 2** (*GNAS*)**.** The graph set $\{\mathcal{G}_m\}_{m=1,2,\ldots,M}$ can be partitioned into training set $\mathcal{D}_{train}$, validation set $\mathcal{D}_{val}$, and test set $\mathcal{D}_{test}$. Given a predefined collection containing all graph neural architectures $\mathcal{P}$, GNAS aims to find an optimal neural architecture $\boldsymbol{p}^* \in \mathcal{P}$, so that the model can be trained on $\mathcal{D}_{train}$ to achieve the best performance, thereby minimizing the fitness on $\mathcal{D}_{val}$. Formally, it can be represented as a following bi-level optimization problem:

$$\begin{cases} \boldsymbol{p}^* = \underset{p\in\mathcal{P}}{\arg\min}\, \mathcal{L}_1(\boldsymbol{p}(\omega^*), \mathcal{D}_{val}) \\ \textbf{s.t.} : \omega^* = \underset{\omega}{\arg\min}\, \mathcal{L}_2(\boldsymbol{p}(\omega), \mathcal{D}_{train}) \end{cases}. \tag{1}$$

Here, $\boldsymbol{p}$ denotes a graph architecture generated by a search method, $\omega$ is the weights of the model $\boldsymbol{p}$, $\mathcal{L}_2(\cdot)$ is the training loss, and $\mathcal{L}_1(\cdot)$ (i.e., the fitness of the search method) measures the performance of $\boldsymbol{p}$ on $\mathcal{D}_{val}$. Finally, the generalization performance of the optimal model $\boldsymbol{p}^*$ is evaluated on $\mathcal{D}_{test}$.

**Definition 3** (*Graph Explainability*)**.** For a specific graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and feature matrix $\mathcal{F}$, the goal of the graph explainability is to identify and score the parts of $\mathcal{G}$ and $\mathcal{F}$ that are most influential in the model's prediction for graph-level classification. In this study, the important brain regions and the connectivity relationships are defined as $\mathcal{G}_s$, and the crucially features are given as $\mathcal{F}_s$.

## 3. Preliminaries

In this section, we separately introduce the preliminary knowledge about graph neural networks, evolutionary algorithms, and the GNNExplainer method.

### 3.1. Graph neural networks

GNN is a category of neural network models capable of directly integrating graph topology and node attributes to facilitate efficient learning of graph representation. Fig. 1 illustrates a typical schematic of using GNN for graph-level binary classification on a graph, and the model contains the following three basic GNN structure components:

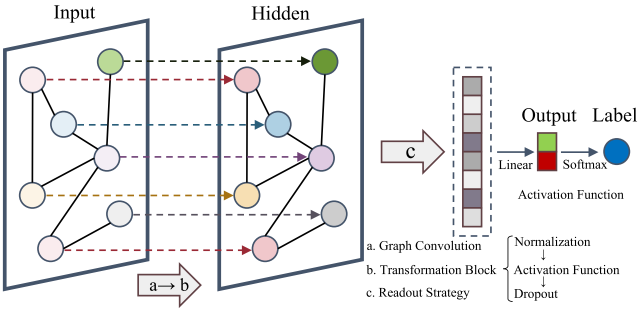

**Fig. 1.** The illustration of the general GNN framework for graph classification.

a. **Graph Convolution**. Most popular graph convolution operations employ a neighbor aggregation (or message-passing) mechanism, where the updating of node features initially involves aggregating messages from neighboring nodes of the current node, followed by combining the aggregated messages with the node's own features. The updating process of the $l$th graph convolution layer for each node $v \in \mathcal{V}$ is formulated as:

$$\begin{cases} \boldsymbol{a}_v^{(l)} = \text{AGGREGATE}^{(l)}\left(\left\{\boldsymbol{f}_u^{(l-1)} : u \in \mathcal{N}(v)\right\}\right) \\ \boldsymbol{f}_v^{(l)} = \text{COMBINE}^{(l)}\left(\boldsymbol{f}_v^{(l-1)}, \boldsymbol{a}_v^{(l)}\right) \end{cases}, \tag{2}$$

where $\mathcal{N}(v)$ represents the set of all neighboring nodes of node $v$. $\boldsymbol{a}_v^{(l)}$ and $\boldsymbol{f}_v^{(l)}$ are the message vector and feature vector of node $v$ at the $l$th layer, respectively. Most graph convolutions such as GCN [33], GraphSAGE [34], GAT [35], and others can be implemented within this framework through various aggregation and combination mechanisms.

b. **Transformation Block**. This block is comprised of normalization, activation function, and dropout, designed in sequence to enhance the model's performance. Normalization involves scaling node features, which improves the training stability and convergence. Typical normalization methods include batch normalization, layer normalization, among others. Activation functions like ReLU and Sigmoid introduce non-linearity into the model, aiding in capturing intricate patterns and relationships in graph-structured data. Dropout is a regularization technique that, during network training, randomly sets a node's hidden representation to zero with a probability $p$, aimed at preventing overfitting and improving the model's generalization capacity.

c. **Readout Strategy**. In graph-level tasks, after updating the node features $\boldsymbol{f}_v$ within the graph, it is typically necessary to employ a readout strategy to aggregate the information of the entire graph structure into a single feature vector $\boldsymbol{h}$. This process can be described as follows:

$$\boldsymbol{h} = \text{READOUT}\left(\left\{\boldsymbol{f}_v, v \in \mathcal{V}\right\}\right). \tag{3}$$

Commonly used readout strategies include mean, sum, max, and others [36]. Different readout strategies have varying effects on the fusion of node features within the entire graph.

### 3.2. Evolutionary algorithms

Evolutionary algorithms are a family of population-based metaheuristic optimization method, which is originally inspired by the process of evolution in nature. In general, most existing evolutionary algorithms (e.g., genetic algorithms [37], particle swarm optimization [38], etc.) share a common framework, as illustrated in Fig. 2, primarily consisting of five components: initialization, evaluation, selection, update, and testing. Specifically, the population represents a collection of candidate individuals, and the selection and update operations continuously iterate the population, selecting the best individual based on evaluation criteria for testing. In the context of evolutionary neural architecture search, each individual represents a specific neural network model, typically measured by its accuracy on a validation dataset, referred to as the fitness value. Throughout the evolution

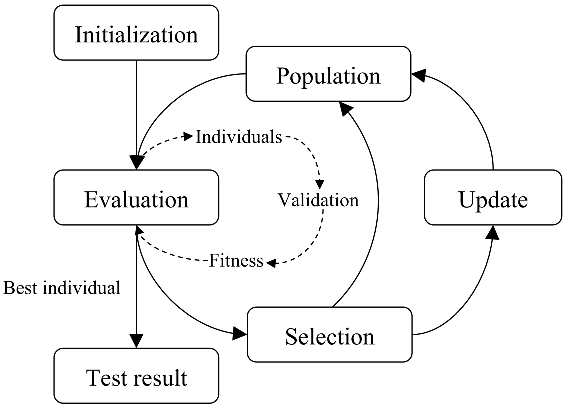

**Fig. 2.** The typical optimization cycle of evolutionary algorithm.

process, environmental selection incorporates evaluated individuals into the population, while underperforming models are eliminated. Furthermore, updating the evaluated individuals ensures the diversity of the population. Through the continuous iteration of the evolutionary population, the model with the best performance in the final population will be used for practical testing tasks.

### 3.3. GNNExplainer method

After training the model obtained through the GNAS process, it is insightful to identify the brain regions that are critical for the final prediction. GNNExplainer, proposed by Ying et al. [39], is employed to obtain the significant subgraph through graph masking. The main idea is to obtain the subgraph $\mathcal{G}_s$ and its associated features $\mathcal{F}_s$ by masking the relevant part of the input entire graph. The mask is determined through an optimization algorithm that iteratively seeks to identify the subgraph that maximizes the mutual information objective:

$$\max_{\mathcal{G}_s} MI(\boldsymbol{Y}, (\mathcal{G}_s, \mathcal{F}_s)) = H(\boldsymbol{Y}) - H(\boldsymbol{Y}|\boldsymbol{G} = \mathcal{G}_s, \boldsymbol{F} = \mathcal{F}_s), \tag{4}$$

where $\boldsymbol{Y}$ represents the predicted label distribution. Random variables are represented by bold letters, while their instantiations are indicated by non-bold letters.

## 4. The proposed method

This section presents EA-GNAS framework to evolve the GNN model. We first define the search space for GNN, then outline the working principle of EA-GNAS. Subsequently, we provide a detailed description of the implementation process, and finally introduce performance evaluation metrics and some comparison methods.

### 4.1. Search space

In general, a complete GNN model is composed of multiple components, and the combination of different components can affect the model's performance. Additionally, different optimizers can also lead to variations in the training results of the model. To ensure that the evolutionary algorithm can search for suitable GNN models, we need to define the search space ($\Omega$) for network components and optimizers, as described in Table 1. In the search space, there are nine components: $c_1$ to $c_6$ represent convolution type, channel, normalization, activation function, dropout, and readout, respectively. These components are essential for constructing a GNN. Different combinations of these components within the search space can generate various GNN models. $o_1$, $o_2$, and $o_3$ represent the optimizer, learning rate, and weight decay, respectively, used for training the GNN model. Specifically, a total of six different graph convolutional types ($c_1$) are selected to learn node features. Among them, GraphConv [33] employs spectral-based convolutional filters to recursively aggregate information from all the direct neighbors of each node and utilizes this information to update the features of each node. ChebConv [40] utilizes the Chebyshev polynomial basis to represent the spectral-based convolutional filters. SAGEConv [34] is a general inductive framework that uses node features to efficiently generate node embeddings for previously unseen data. GATConv [35] incorporates a masked self-attention mechanism, which assigns appropriate weights to various neighboring nodes, thereby directing attention towards nodes of greater significance while attenuating the influence of nodes with comparatively lesser relevance. ResGatedGraphConv [41] introduces residual connections and recurrent neural networks into graph convolution, enabling it to handle variable-length graph tasks. Meanwhile, the number of output channels ($c_2$) in graph convolution determines the output dimensionality of node features, and we offer a total of seven selectable options. Furthermore, the search space encompasses three common normalization methods ($c_3$), five activation functions ($c_4$), and five different dropout rates ($c_5$), where 'None' indicates that the component is not selected. In the readout strategy ($c_6$), five statistical methods are employed to aggregate node features into graph features. Finally, three commonly used optimizers (i.e., Adam, SGD, and RMSprop) ($o_1$), along with different learning rates ($o_2$) and weight decay rates ($o_3$), are provided as candidate components for training the GNN model.

### 4.2. EA-GNAS

**Algorithm 1** A generic graph neural architecture search algorithm based on evolutionary algorithm.

```
Input: The hyper-parameter set $\boldsymbol{\Phi}$ of EA, graph block number $L$,
    learning iteration $N$, individual number $I$, search space $\Omega$, training
    set $\mathcal{D}_{train}$, validation set $\mathcal{D}_{val}$, architecture memory $\mathcal{P} \leftarrow \emptyset$.
Output: The best architecture $\boldsymbol{p}^*$ in $\mathcal{P}$
Obtain the lower bound $\boldsymbol{lb}$ and upper bound $\boldsymbol{ub}$ through $L$ and $\Omega$
Initialize the individuals $\boldsymbol{S} \leftarrow$ Uniform($\boldsymbol{lb}, \boldsymbol{ub}$)
for iteration = 1 to $N$ do
    Obtain all architectures $\boldsymbol{P}$ through decoding $\boldsymbol{S}$
    for $i = 1$ to $I$ do
        Construct the GNN models according to $\boldsymbol{p}_i \in \boldsymbol{P}$
        $\mathcal{P} \leftarrow \mathcal{P} \cup \boldsymbol{p}_i$
        Compute the fitness $\phi_i \leftarrow 1 -$ CrossValidation($\boldsymbol{p}_i, \mathcal{D}_{train}, \mathcal{D}_{val}$)
    end for
    Update $\boldsymbol{S} \leftarrow$ EA($\boldsymbol{\Phi}, \boldsymbol{S}, \boldsymbol{\phi}$)
    Limit boundary $\boldsymbol{S} \leftarrow$ Limit($\boldsymbol{S}, \boldsymbol{lb}, \boldsymbol{ub}$)
end for
```

To describe the architecture of GNN clearly, we treat the components $c_1$ to $c_5$ within the GNN as a single graph convolution block and represent all the components to be searched as a tuple. Assuming we search for the optimal model with $L$ graph convolution blocks, the architecture $\boldsymbol{p}$ can be represented as follows:

$$\boldsymbol{p} = (c_1^{(1)}, c_2^{(1)}, c_3^{(1)}, c_4^{(1)}, c_5^{(1)}, \dots, c_1^{(L)}, c_2^{(L)}, c_3^{(L)}, c_4^{(L)}, c_5^{(L)}, c_6, o_1, o_2, o_3), \tag{5}$$

where the first block and last blocks are indexed by (1) and ($L$), respectively. It is worth noting that the variables in $\boldsymbol{p}$ are discrete, so some evolution algorithms based on continuous space position updates, such as PSO and snake optimization (SO) [42], cannot be directly applied to GNAS. To address this issue, we implement a floor operation following the position updates in the evolutionary algorithm, resulting in an integer value that uniquely corresponds to the candidate positions of the components within the search space. The proposed generic GNAS framework based on EA is shown in Algorithm 1.

The algorithm's inputs include the hyper-parameter set $\boldsymbol{\Phi}$ of the evolutionary algorithm, the hyper-parameters ($L$, $N$, $I$) for training

**Table 1**
The search space for GNN components and optimizer.

| Components | Search space ($\Omega$) |
|---|---|
| Convolution type ($c_1$) | GraphConv, ChebConv, SAGEConv, GATConv, ResGatedGraphConv |
| Channel ($c_2$) | 4, 8, 16, 32, 64, 128, 256 |
| Normalization ($c_3$) | None, Batch Normalization, Layer Normalization, Instance Normalization |
| Activation ($c_4$) | None, Sigmoid, Tanh, ReLU, ELU, Leaky ReLU |
| Dropout ($c_5$) | None, 0.1, 0.2, 0.3, 0.4, 0.5 |
| Readout ($c_6$) | Sum, Mean, Max, Var, Std |
| Optimizer ($o_1$) | Adam, SGD, RMSprop |
| Learning rate ($o_2$) | 1e−2, 1e−3, 1e−4 |
| Weight decay ($o_3$) | 1e−3, 1e−4, 1e−5 |

**Table 2**
Demographic characteristics of subjects included in the SRPBS Dataset.

| Site | | Healthy group | SSD group |
|---|---|---|---|
| KTT | No., M:W | 75, 48:27 | 44, 25:19 |
| | Age | 28.9 (9.1) | 37.3 (9.7) |
| KUT | No., M:W | 159, 93:66 | 44, 20:24 |
| | Age | 36.5 (13.6) | 41.3 (10.9) |
| SWA | No., M:W | 101, 86:15 | 19, 15:4 |
| | Age | 28.4 (7.9) | 42.9 (8.4) |
| UTO | No., M:W | 170, 78:92 | 35, 23:12 |
| | Age | 35.6 (17.5) | 31.7 (10.4) |

M, men; W, women.

the GNN model, and the training and validation sets. Among them, $\boldsymbol{\Phi}$ represents the specific hyper-parameters of different evolutionary algorithms, such as the crossover rate and mutation rate for GA, and the cognitive factor, social factor, and inertia weight for PSO. The algorithm's output is the optimal architecture $\boldsymbol{p}^*$ with the best performance in the architecture memory $\mathcal{P}$. The algorithm initializes a population of individuals using a uniform distribution within the search space, and then iteratively updates the population using the evolutionary algorithm. In each iteration, the algorithm decodes the individuals to obtain a set of GNN architectures, constructs the corresponding models, and evaluates their fitness using cross-validation on the training and validation sets. The fitness values are then used to update the population using the EA algorithm, which searches for the optimal architecture. The algorithm terminates after $N$ iterations, and the best architecture is selected from the final population.

Moreover, several important considerations are worth mentioning. First, when we have a graph block number denoted as $L$, it is essential to note that the dimensions of the lower and upper bounds for optimization variables are identical, each being $5L + 4$. Specifically, $\boldsymbol{lb} = (0, 0, \dots, 0)$, and $\boldsymbol{ub} = (ub_1, ub_2, \dots, ub_{5L+4})$. Here, the $i$th upper bound element in $\boldsymbol{ub}$ is represented as $ub_i$, which is equal to the candidate number of the $i$th component in (5). Secondly, we randomly select a set of $I$ individuals from the interval between $\boldsymbol{lb}$ and $\boldsymbol{ub}$, and we represent this population as $\boldsymbol{S} = \{\boldsymbol{s}_i | i \in [1, I], i \in \mathbb{N}^+\}$. During the construction of the GNN model, the decoding process includes rounding down each individual $\boldsymbol{s}_i \in \mathbb{R}^{5*L+4}$, and this resulting integer value serves as an index for the search space $\Omega$. Finally, the evolutionary algorithm will iteratively update the population $\boldsymbol{S}$ while limiting upper and lower bounds throughout $N$ iterations.

### 4.3. Dataset construction

The rs-fMRI datasets in this study were obtained from the Japanese Strategic Research Program for the Promotion of Brain Science (SRPBS) dataset 1 [43]. This dataset consists of 142 SSD patients (83 men versus 59 women) and 505 health subjects (305 men versus 200 women) from 4 sites (KTT, KUT, SWA, and UTO). Subject demographics information are listed in Table 2, and data are shown as means (standard deviation). The brain features pertaining to individual subjects, specifically functional connectivities (FC), were derived from 10-minute rs-fMRI BOLD signals that had undergone identical transformation procedures. The procedures include slice-time correction, realignment, co-registration, segmentation of T1-weighted structural images, normalization to the Montreal Neurological Institute space, and spatial smoothing using an isotropic Gaussian kernel with a 6 mm full-width at half-maximum. In order to calculate these subject-specific functional connectivities, the BrainVISA Sulci Atlas parcellation scheme was utilized, which subdivided each individual image into 140 regions as detailed by Perrot et al. [44]. For the extraction of time series data and the computation of the connectivity matrix, the standard Pearson correlation coefficient (PCC) method was employed. As a result, 9730 connectivity features were computed for each subject, thereby constituting a weighted brain feature matrix with dimensions $[647, 140, 140]$.

FC data contains a large amount of feature information, but too much information can cause feature redundancy and prolong training time. Therefore, effective preprocessing of FC data is crucial for the GNAS process. After data collection, we performed a series of data preprocessing steps to construct a graph dataset suitable for training GNN models. First, we randomly partitioned all subjects into training and test sets in a 4:1 ratio. Then, we averaged the FC matrices of all training set and used minimum spanning trees (MST) [45] to search for nodes in the averaged graph. The threshold of MST is used to control the scale of the generated minimum spanning tree to filter out important connection relationships. Next, we transformed the results of MST into binary form, creating a binary adjacency matrix, which serves as the node connectivity for the graph dataset. Additionally, the 140-dimensional weighted features of each brain region are used as node features. The combination of node connectivity and node features forms the foundation of the graph dataset. Finally, The training and test graph datasets were constructed according to the binary adjacency matrix and node features. It is important to note that the training set includes a validation set, which is used for both the exploration of GNN models and the prediction of SSD in the optimal GNN model. The test set is exclusively used to evaluate the performance of the optimal GNN model.

### 4.4. Implementation details

Fig. 3 illustrates in detail the implementation framework of the proposed method, including data collection, data preprocessing, EA-GNAS, SSD prediction, and the explainability of prediction results. In the data preprocessing stage, we randomly selected 142 healthy subjects to balance the sample data, avoiding selection bias when training the GNN model. In EA-GNAS, we employed five-fold cross-validation to assess model performance, set the training epochs to 10, batch size to 64, and used cross-entropy as the loss function. Furthermore, we selected six popular evolutionary algorithms, namely, genetic algorithm (GA) [37], particle swarm optimization (PSO) [38], whale optimization algorithm (WOA) [46], grey wolf optimization (GWO) [47], snake optimization (SO) [42], and salp swarm algorithm (SSA) [48]. In GA, the crossover rate and mutation rate were set to 0.8 and 0.01, respectively. In PSO, the cognitive factor, social factor, and inertia weight were set to 2, 2, and 0.9, respectively. In WOA, the constant coefficient was set to 1. In SO, the two threshold values were 0.25 and 0.6, and the

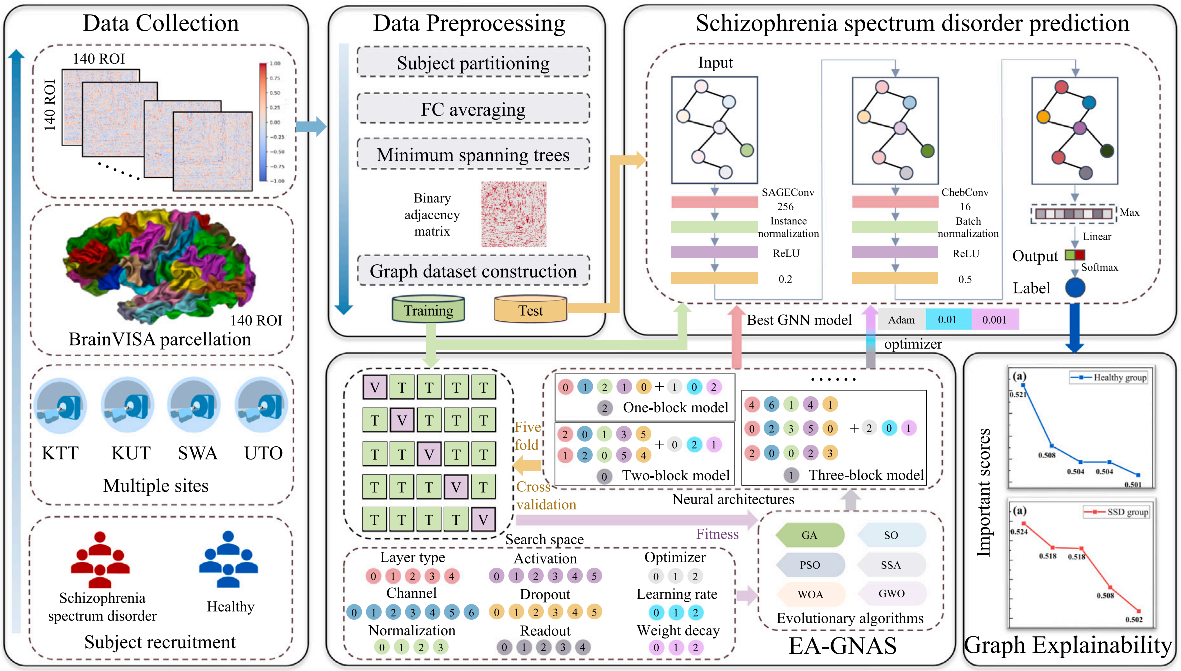

**Fig. 3.** A generic EA-GNAS framework for SSD prediction and graph explainability.

three constant coefficients were 0.5, 0.05, and 0.5. Additionally, the learning iteration number $N$ and the number of individuals $I$ for all evolutionary algorithms were set to 100 and 10, respectively.

### 4.5. Comparison methods and performance metrics

In the comparative study of SSD prediction, we have selected four common ML methods and four DL methods mentioned in Introduction. The ML methods include SVM, KNN, RF, and LR. Among them, SVM aims to find the optimal hyperplane for classifying data points of different categories; KNN relies on distance metrics for data point classification and we set the number of nearest neighbors to 3; RF improves classification performance by combining the prediction results of multiple decision trees and we set the number of trees to 100; LR is a widely used linear model for binary classification problems, estimating the probability that an observation belongs to a certain category using a logistic function. The DL methods include BrainNetCNN [17], LGGNN [26], FCBasedGCN [25], and three other classical GNN models, namely GCN, GAT, and SAGE. Following the structure in [21], the number of layers for GCN [33], GAT [35], and SAGE [34] is set to three, with the number of features in the intermediate layers set to 128.

Five metrics are reported to evaluate the performance of all ML-based and DL-based SSD prediction methods. Accuracy is a measure of the overall performance of a classification model, representing the ratio of correctly predicted samples to the total number of samples, typically used when dealing with evenly distributed classes. Precision refers to the proportion of true positive samples among those predicted as positive by the model, assessing the prediction accuracy of the model. Recall is the ratio of correctly identified positive samples to the total number of true positive samples, evaluating the model's sensitivity. F1 score is the harmonic mean of precision and recall, providing a comprehensive assessment of the model's accuracy and sensitivity. The calculation formulas for these performance metrics are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{6}$$

$$Precision = \frac{TP}{TP + FP}, \tag{7}$$

$$Recall = \frac{TP}{TP + FN}, \tag{8}$$

$$F1\text{-}score = 2 * \frac{Precision * Recall}{Precision + Recall}. \tag{9}$$

Here, $TP$ refers to the healthy samples that are correctly classified. $TN$ represents the SSD samples that are correctly classified. $FP$ indicates the SSD samples that are incorrectly classified, while $FN$ signifies the healthy samples that are incorrectly classified. In addition, the relationship between $TP$ rate and $FP$ rate at different thresholds can be represented by the ROC curve, and AUC represents the area under the ROC curve, typically ranging from 0 to 1, with a higher value indicating better model performance.

## 5. Results and discussions

In this section, we first compare the GNAS results of different evolutionary algorithms and obtain the optimal model. Secondly, we compare the SSD prediction performance of different models. Then, we conduct an explainability analysis of the GNN model. At last, the limitation of this work is discussed.

### 5.1. Parameter selection and network architecture search results

Fig. 4 displays the binary adjacency matrix graphs under various MST thresholds and the optimization process of various evolutionary algorithms. In this adjacency matrix, red areas represent the connections between brain regions, while non-red areas indicate no connection. The MST threshold determines the number of brain region connections, with a larger threshold resulting in more brain region connections. When the threshold is 1, it indicates that all 140 brain regions are fully connected pairwise. However, excessive connections can lead to information redundancy and increase the computational burden. Therefore, we selected three thresholds of 0.1, 0.2, and 0.3. To compare the search efficiency of each evolutionary algorithm, Fig. 4 also presents the change curves of the mean fitness of all individuals with each iteration. For a more intuitive comparison, these curves were processed with a moving average, with a sliding iteration number of 10. In each iteration, 10 individuals (i.e., 10 GNN models) are trained and their fitness are calculated. The mean fitness reflects the search effect of the evolutionary algorithm: the lower the value, the better the search

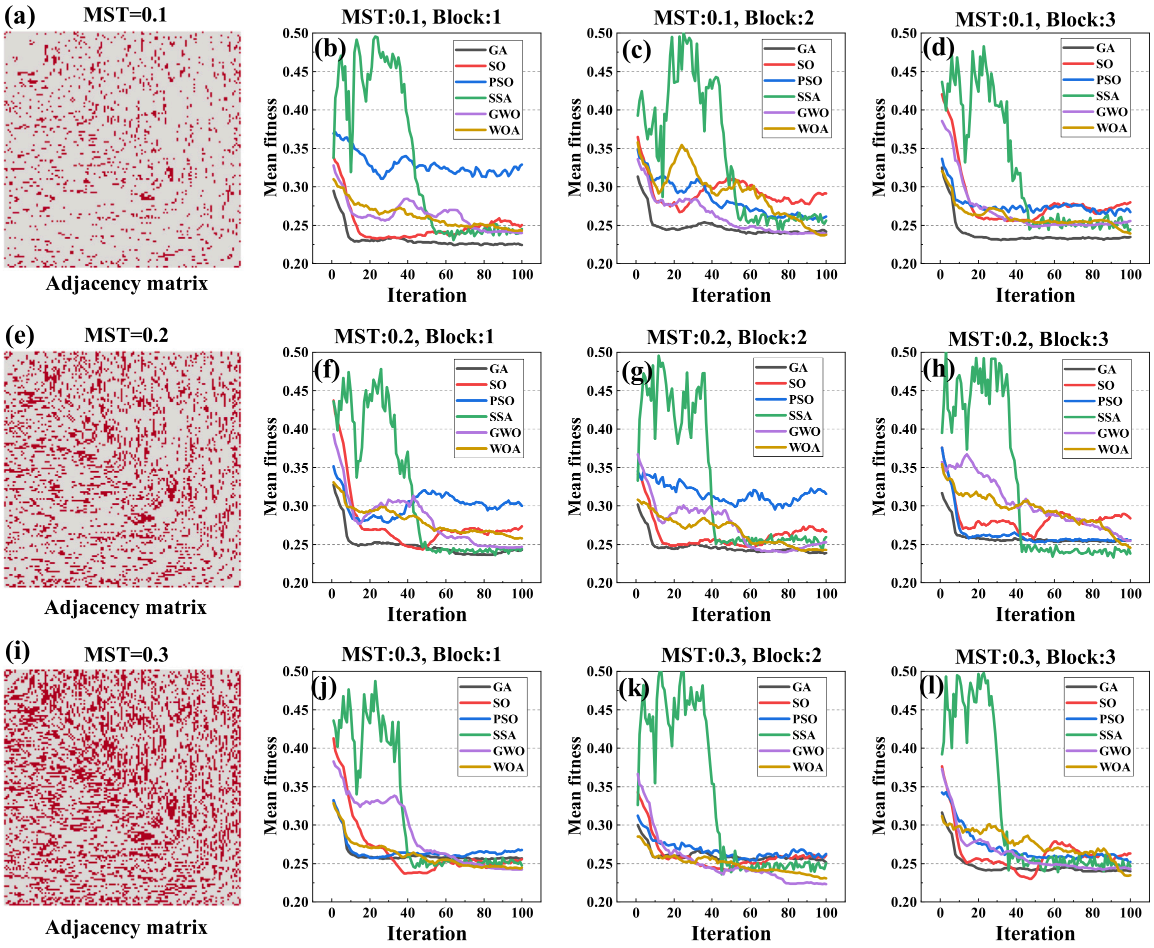

**Fig. 4.** The curves of the mean fitness of all individuals versus iteration under different evolutionary algorithms. (a), (e), and (i) represent the adjacency matrix graphs when the MST threshold is 0.1, 0.2, and 0.3, respectively; (b) to (d) are the mean fitness values under MST 0.1 with block numbers ranging from 1 to 3; (f) to (h) are the mean fitness values under MST 0.2 with block numbers ranging from 1 to 3; (j) to (i) are the mean fitness values under MST 0.3 with block numbers ranging from 1 to 3.

**Table 3**
Best fitness comparison of different evolutionary algorithms under various MST thresholds and block numbers.

| EA | One-block model | | | Two-block model | | | Three-block model | | |
|---|---|---|---|---|---|---|---|---|---|
| | MST = 0.1 | MST = 0.2 | MST = 0.3 | MST = 0.1 | MST = 0.2 | MST = 0.3 | MST = 0.1 | MST = 0.2 | MST = 0.3 |
| SSA | 0.2028 | 0.2202 | 0.2161 | 0.2071 | 0.2204 | 0.2028 | 0.2071 | 0.2114 | 0.1895 |
| WOA | 0.2024 | 0.2072 | 0.2069 | 0.2025 | 0.2116 | **0.1895** | 0.2072 | 0.2027 | **0.1853** |
| GWO | 0.2163 | 0.2207 | 0.2290 | 0.2026 | 0.2070 | 0.1938 | 0.2023 | 0.2026 | 0.2073 |
| PSO | 0.1983 | 0.2073 | 0.2073 | 0.2026 | 0.1899 | 0.2116 | 0.1894 | 0.2070 | 0.1984 |
| SO | 0.2027 | 0.2159 | 0.2069 | **0.1853** | 0.2026 | 0.2071 | 0.1897 | 0.1942 | 0.1984 |
| GA | **0.1895** | **0.2027** | **0.1985** | 0.1939 | **0.1850** | 0.1939 | **0.1851** | **0.1939** | 0.2026 |
| Mean | 0.2020 | 0.2123 | 0.2108 | 0.1990 | 0.2028 | 0.1998 | 0.1968 | 0.2020 | 0.1969 |

effect. The results show that, in most cases, the mean fitness curve of SSA starts to fluctuate dramatically and then gradually stabilizes, while the curve of GA converges faster, indicating that GA can find efficient GNN models and optimizers more quickly. Furthermore, all curves show a downward trend because evolutionary algorithms search based on diversity and selection. In the early stages of exploration, due to higher diversity, the algorithm will broadly explore various combinations, which may result in lower accuracy. However, as iterations progress, the algorithm will focus on high-performance areas, thereby improving accuracy.

Table 3 reports the comparison results of the best fitness obtained by six evolutionary algorithms under different parameter settings. The number of graph blocks typically influences the network's feature extraction performance. As shown in Table 3, an increase in the number of graph blocks leads to a certain decreasing trend in the mean fitness of the algorithms, indicating that during the GNAS process, most GNN models perform better with a larger number of graph blocks. Among all these algorithms, GA consistently demonstrates superior performance across multiple settings, achieving the best fitness values in both the one-block and three-block models at most MST thresholds. This implies that GA has the better robustness and adaptability to varied model complexities and threshold constraints in GNAS. Conversely, while algorithms like WOA and SO exhibit optimal performance in specific scenarios (e.g., SO for the two-block model at MST = 0.1 and WOA for the three-block model at MST = 0.3), their efficacy is not consistent across the board. As a result, we select the GNN model and optimizer found by GA with MST = 0.2 and block=2 for subsequent SSD prediction comparisons, at which point the best fitness reaches its lowest at 0.1850.

Fig. 5 compares the validation accuracies of the optimal models obtained during the search process by different evolutionary algorithms.

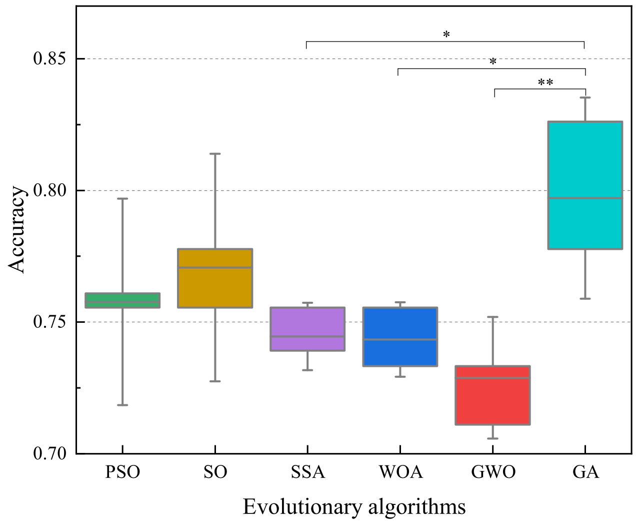

**Fig. 5.** Validation accuracy boxplots of the GNN models searched by six evolutionary algorithms. The t-test was applied to compare GA with other algorithms. The one asterisk $*$ and two asterisks $**$ denote $0.01 < p \leq 0.05$ and $p \leq 0.01$, respectively.

**Table 4**
Comparison results of performance metrics for the GNN models searched by different evolutionary algorithms.

| EA | Accuracy | Precision | Recall | F1 score | AUC |
|---|---|---|---|---|---|
| SSA | 0.7544 | 0.8519 | 0.6970 | 0.7667 | 0.7652 |
| WOA | 0.7544 | **0.8800** | 0.6667 | 0.7586 | 0.7708 |
| GWO | 0.7193 | 0.8400 | 0.7241 | 0.6364 | 0.7348 |
| PSO | 0.7719 | 0.8333 | 0.7937 | 0.7576 | 0.7746 |
| SO | 0.7895 | 0.8621 | 0.7955 | 0.7576 | 0.8065 |
| GA | **0.8246** | 0.8710 | **0.8182** | **0.8438** | **0.8258** |

**Table 5**
Comparison results of performance metrics for different methods.

| Methods | | Accuracy | Precision | Recall | F1 score | AUC |
|---|---|---|---|---|---|---|
| ML-based | SVM [6] | 0.7368 | 0.8462 | 0.6667 | 0.7458 | 0.7500 |
| | KNN [14] | 0.6491 | 0.8421 | 0.4848 | 0.6154 | 0.6799 |
| | RF [11] | 0.6667 | 0.8500 | 0.5152 | 0.6415 | 0.6951 |
| | LR [11] | 0.7368 | 0.8750 | 0.6364 | 0.7368 | 0.7557 |
| DL-based | BrainNetCNN [17] | 0.7719 | 0.7632 | **0.8788** | 0.8169 | 0.7519 |
| | GCN [33] | 0.7368 | 0.7647 | 0.7879 | 0.7761 | 0.7273 |
| | GAT [35] | 0.7544 | 0.7879 | 0.7879 | 0.7879 | 0.7481 |
| | SAGE [34] | 0.7544 | 0.8276 | 0.7273 | 0.7742 | 0.7595 |
| | FCBasedGCN [25] | 0.7368 | 0.7368 | 0.8485 | 0.7887 | 0.7159 |
| | LGGNN [26] | 0.8070 | **0.8846** | 0.7419 | 0.8070 | 0.8133 |
| | GA-GNAS (Ours) | **0.8246** | 0.8710 | 0.8182 | **0.8438** | **0.8258** |

The t-test is employed to compare whether there is a significant difference between the accuracies of the two sample groups. The findings reveals that the average accuracy of the GNN model obtained through the GA search process is close to 0.80, substantially surpassing the average accuracies of the GNN models derived from other algorithmic searches. Meanwhile, the results yielded by the GA exhibits statistically significant deviations from those of the SSA, WOA, and GWO algorithms ($p \leq 0.05$).

Fig. 6 shows the changes in the components of the best architecture during the GNAS process with GA. Most components, such as the optimizer and layers in the second block, are determined in the early stages of exploration, indicating that these specific components provide a significant performance boost to the network. At the same time, some components in the first block only stabilize after the middle of the exploration phase. This suggest that the algorithm might get stuck in a local optimum initially but later jumps out in search of a better solution. Therefore, in the early stages of the search, the evolutionary algorithm determines the main structure of the network, and afterwards, it continuously optimizes the construction details of the architecture.

To further demonstrate the differences among the GNN models searched by different evolutionary algorithms, Table 4 reports their performance on the SSD prediction task. Bold values indicate the best scores, while underlined values denote the second-best scores. The GNN model searched by GA achieves an accuracy of 0.8246, a recall of 0.8182, an F1 score of 0.8438, and an AUC of 0.8258, performing the best overall. The model searched by WOA has the highest precision of 0.8800. According to the accuracy and comprehensive metric AUC, on this SSD prediction task, the GNN model searched by GA outperforms those by other algorithms, followed by SO, while GWO performs the worst. The results are consistent with the cross-validation results in Fig. 5.

### 5.2. Comparison results with baseline methods

Table 5 reports the performance metrics of various ML-based and DL-based methods. Among the ML-based methods, LR has the highest precision of 0.8750, while SVM and LR both achieve the same top accuracy of 0.7368. KNN performs the worst among all ML-based methods. Among the DL-based methods, GA-GNAS outperforms others in terms of accuracy, F1 score, and AUC. LGGNN has the highest precision of 0.8846, and ranks second in accuracy and AUC. However, this method's recall is not outstanding, implying that its predictions may result in more false negatives. Overall, GA-GNAS displays balanced performance across all metrics, indicating that it can provide robust prediction results under various evaluation conditions. Such balance is particularly important for practical applications because it ensures reliable predictions in various scenarios and conditions.

In this study, the GNN-based methods outperforms traditional machine learning methods in most evaluation metrics. This phenomenon is primarily because the connections between brain regions exhibit clear spatial structures and interdependent characteristics. GNN can capture long-distance dependencies through iterative node updates and message-passing mechanisms, thus fully utilizing the relational information in these structured data, enhancing the results of SSD prediction. Additionally, the proposed method shows superior performance compared to hand-crafted GNN models. This superiority is mainly because hand-crafted architectures are often limited by the designer's experience and prior knowledge, whereas evolutionary algorithms are not constrained by these factors. They can explore excellent architectures that human designers might overlook or not consider. Therefore, using evolutionary algorithms can not only obtain a high-performance GNN model for SSD prediction but also save the time and effort required for manually designing and adjusting GNN models.

### 5.3. Explainability analysis of the GNN models

Although the proposed methods and some comparative methods can effectively distinguish between the healthy group and the SSD group, the key factors behind the inference of these models remain unclear. Therefore, we utilized the GNNExplainer method for explainability analysis of GNN models. Since GNNExplainer is a technique specifically designed for the explainability analysis of the GNN model, and BrainNetCNN and ML-based methods are not GNN models, we did not perform explainability analysis on them. BrainVISA has defined 140 brain regions, and we present the top five brain regions that have the greatest impact on the different GNN models for both the healthy group and the SSD group, as shown in Figs. 7 and 8. The x-axis represents the brain region annotations defined by BrainVISA, and the y-axis represents the importance scores of each brain region. It is found that the brain regions of interest differ among models, yet

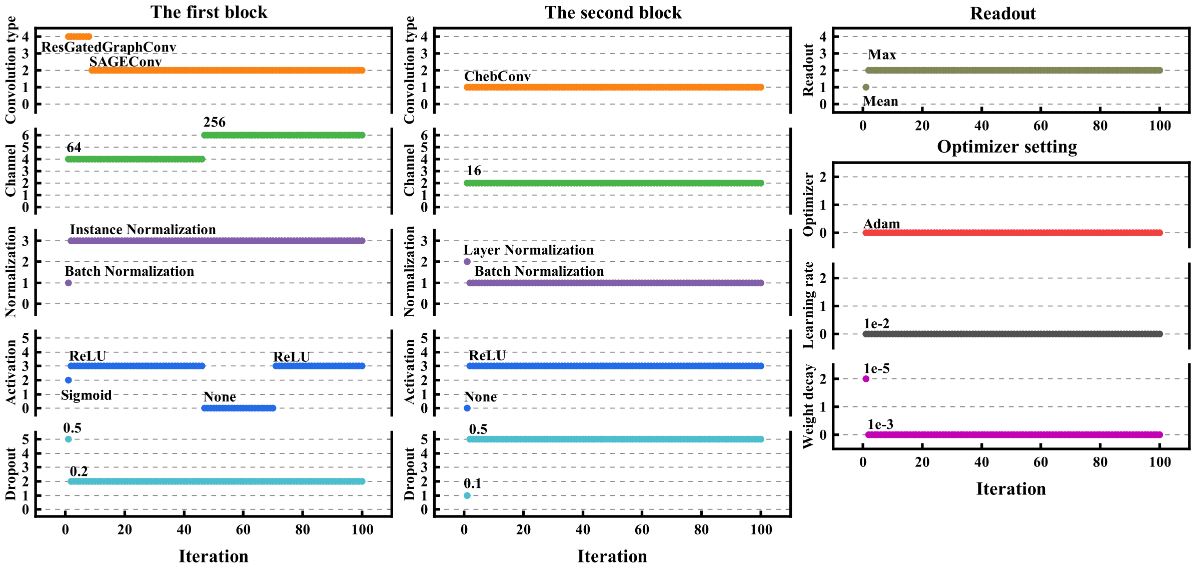

**Fig. 6.** The search process of all components in the optimal architecture obtained through GA-GNAS.

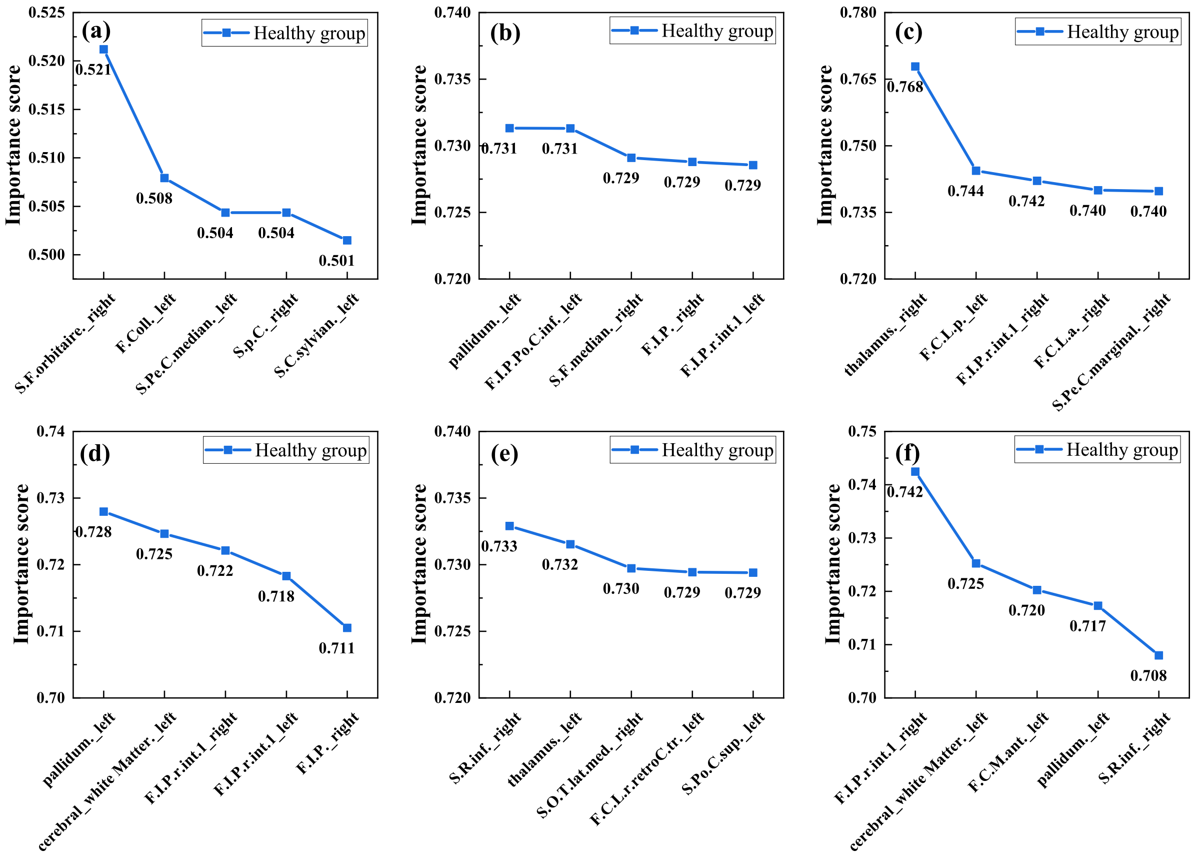

**Fig. 7.** Results of the GNNExplainer methods under different GNN models in the healthy group. From (a) to (f) are the results of GA-GNAS, GCN, GAT, SAGE, FCBasedGCN, and LGGNN, respectively.

there are similarities. For the healthy group, GCN, SAGE, and LGGNN all identify the Intraparietal Sulcus (brain regions abbreviated with F.I.P) and Pallidum as two important areas. GAT and FCBasedGCN both recognize the Thalamus as having a high impact. For the SSD group, the Ventricle is very important for predicting SSD, identified by all GNN models. This is consistent with existing research indicating that ventricle enlargement is a key indicator for diagnosing SSD [49,50]. Additionally, the Temporal Sulcus (brain regions abbreviated with S.T.) reveals another piece of evidence impacting the identification of SSD. This finding is consistent with research reporting differences in the connectivity of the posterior superior temporal sulcus in social cognition between SSD patients and healthy individuals [51]. Other brain regions identified by the GNN models may also be meaningful for the diagnosis of SSD, providing valuable references for neuroscientists.

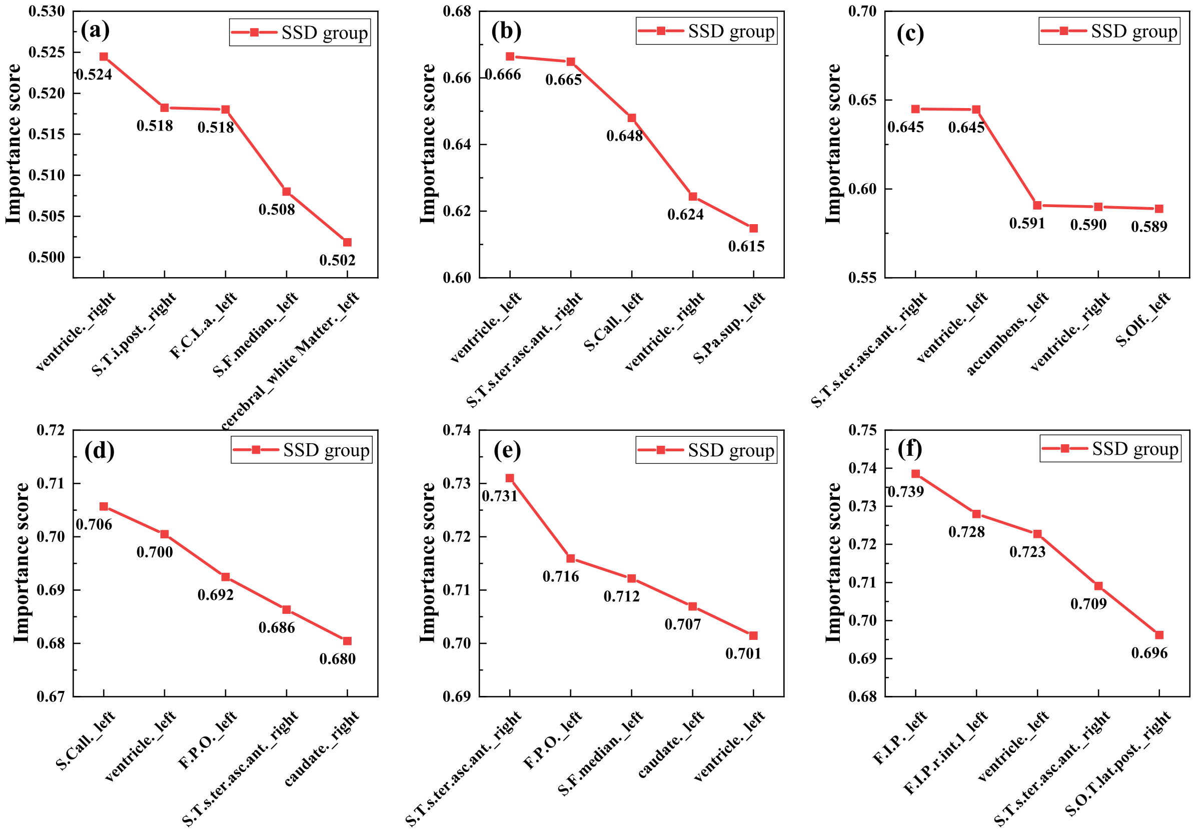

**Fig. 8.** Results of the GNNExplainer methods under different GNN models in the SSD group. From (a) to (f) are the results of GA-GNAS, GCN, GAT, SAGE, FCBasedGCN, and LGGNN, respectively.

### *5.4. Limitation*

Although this study proposes a effective deep learning method for accurately predicting SSD, there are still some limitations. On one hand, the size of the dataset we used may not be sufficient to capture all the complexities of SSD. A smaller dataset might lead to model overfitting, reducing its generalizability on broader and more diverse data. On the other hand, we conducted experimental comparisons of MST and graph block at different values and selected the GNN model searched by GA-GNAS with MST at 0.2 and graph block at 2. This parameter combination is predefined and might not be optimal, potentially limiting enhancements to the model's performance. Therefore, future research will consider using larger datasets to increase data diversity and further optimizing and expanding the search space to overcome these limitation.

## 6. Conclusion

In this study, we propose a novel framework, EA-GNAS, to search for an optimal GNN model and its associated optimizer. This model aims to classify SSD patients and healthy individuals using rs-fMRI data. Specifically, we compare multiple evolutionary algorithms, with GA demonstrating superior performance in the majority of cases. Additionally, when juxtaposed with conventional ML techniques and other DL methodologies, GA-GNAS exhibits enhanced performance in accuracy, F1-score, and AUC metrics. These comparative findings underscore the efficacy and robustness of the proposed method. Furthermore, we employ GNNExplainer to illustrate which brain regions and connections the GNN model pays more attention to when classifying SSD patients and healthy individuals. This offers valuable insights for neuroscientists delving into SSD analysis.

## CRediT authorship contribution statement

**Shurun Wang:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Hao Tang:** Writing – review & editing, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization. **Ryutaro Himeno:** Writing – review & editing, Supervision, Methodology. **Jordi Solé-Casals:** Writing – review & editing, Validation, Methodology. **Cesar F. Caiafa:** Writing – review & editing, Resources, Methodology, Conceptualization. **Shuning Han:** Writing – review & editing, Methodology, Investigation, Data curation. **Shigeki Aoki:** Writing – review & editing, Supervision, Methodology, Investigation. **Zhe Sun:** Writing – review & editing, Visualization, Supervision, Resources, Methodology, Data curation, Conceptualization.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: The authors declare that they have no known competing Financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The authors would like to thank China Scholarship Council for supporting this work. S. H. work was carried out as part of the doctoral programme in Experimental Sciences and Technology at the University of Vic - Central University of Catalonia.

## Funding

This work was supported by the National Key R&D Program of China (No. 2017YFE0129700) and China Scholarship Council (No. 202206690046).

## References

[1] H.Y. Chong, S.L. Teoh, D.B.-C. Wu, S. Kotirum, C.-F. Chiou, N. Chaiyakunapruk, Global economic burden of schizophrenia: a systematic review, Neuropsych. Dis. Treat. (2016) 357–373.

[2] K.R. Patel, J. Cherian, K. Gohil, D. Atkinson, Schizophrenia: overview and treatment options, Pharmacol. Ther. 39 (9) (2014) 638.

[3] M. Dabiri, F. Dehghani Firouzabadi, K. Yang, P.B. Barker, R.R. Lee, D.M. Yousem, Neuroimaging in schizophrenia: A review article, Front. Neurosci. 16 (2022) 1042814.

[4] L. Attademo, F. Bernardini, N. Verdolini, Neural correlates of schizotypal personality disorder: a systematic review of neuroimaging and eeg studies, Curr. Med. Imaging 17 (11) (2021) 1283–1298.

[5] C. Cruz-Martinez, C.A. Reyes-Garcia, N. Vanello, A novel event-related fMRI supervoxels-based representation and its application to schizophrenia diagnosis, Comput. Methods Programs Biomed. 213 (2022) 106509.

[6] X. Ma, W.F.Z. Yang, W. Zheng, Z. Li, J. Tang, L. Yuan, L. Ouyang, Y. Wang, C. Li, K. Jin, et al., Neuronal dysfunction in individuals at early stage of schizophrenia, a resting-state fmri study, Psychiatry Res. 322 (2023) 115123.

[7] I. Gallos, L. Mantonakis, E. Spilioti, E. Kattoulas, E. Savvidou, E. Anyfandi, E. Karavasilis, N. Kelekis, N. Smyrnis, C. Siettos, The relation of integrated psychological therapy to resting state functional brain connectivity networks in patients with schizophrenia, Psychiatry Res. 306 (2021) 114270.

[8] Y. Du, X. He, P. Kochunov, G. Pearlson, L.E. Hong, T.G. van Erp, A. Belger, V.D. Calhoun, A new multimodality fusion classification approach to explore the uniqueness of schizophrenia and autism spectrum disorder, Hum. Brain Mapp. 43 (12) (2022) 3887–3903.

[9] S. Ramkiran, A. Sharma, N.P. Rao, Resting-state anticorrelated networks in schizophrenia, Psychiatry Res. Neuroim. 284 (2019) 1–8.

[10] Y. Gao, X. Tong, J. Hu, H. Huang, T. Guo, G. Wang, Y. Li, G. Wang, Decreased resting-state neural signal in the left angular gyrus as a potential neuroimaging biomarker of schizophrenia: an amplitude of low-frequency fluctuation and support vector machine analysis, Front. Psychiatry 13 (2022) 949512.

[11] S. Srinivasagopalan, J. Barry, V. Gurupur, S. Thankachan, A deep learning approach for diagnosing schizophrenic patients, J. Exp. Theor. Artif. Intell. 31 (6) (2019) 803–816.

[12] W. Zhu, S. Shen, Z. Zhang, Improved multiclassification of schizophrenia based on xgboost and information fusion for small datasets, Comput. Math. Methods Med. 2022 (2022).

[13] M. Sharma, R.K. Patel, A. Garg, R.-S. Tan, U.R. Acharya, Automated detection of schizophrenia using deep learning: a review for the last decade, Physiol. Meas. (2023).

[14] J.R. de Miras, A.J. Ibáñez Molina, M.F. Soriano, S. Iglesias-Parro, Schizophrenia classification using machine learning on resting state EEG signal, Biomed. Signal Proces. 79 (2023) 104233.

[15] D. Sadeghi, A. Shoeibi, N. Ghassemi, P. Moridian, A. Khadem, R. Alizadehsani, M. Teshnehlab, J.M. Gorriz, F. Khozeimeh, Y.-D. Zhang, et al., An overview of artificial intelligence techniques for diagnosis of schizophrenia based on magnetic resonance imaging modalities: Methods, challenges, and future works, Comput. Biol. Med. 146 (2022) 105554.

[16] L.-L. Zeng, Z. Fan, J. Su, M. Gan, L. Peng, H. Shen, D. Hu, Gradient matching federated domain adaptation for brain image classification, IEEE T. Neur. Net. Lear. (2022).

[17] J. Kawahara, C.J. Brown, S.P. Miller, B.G. Booth, V. Chau, R.E. Grunau, J.G. Zwicker, G. Hamarneh, BrainNetCNN: Convolutional neural networks for brain networks; towards predicting neurodevelopment, NeuroImage 146 (2017) 1038–1049.

[18] A. Gupta, R. Daniel, A. Rao, P.P. Roy, S. Chandra, B.-G. Kim, Raw electroencephalogram-based cognitive workload classification using directed and nondirected functional connectivity analysis and deep learning, Big Data (2023).

[19] X. Chen, B. Li, H. Jia, F. Feng, F. Duan, Z. Sun, C.F. Caiafa, J. Solé-Casals, Graph empirical mode decomposition-based data augmentation applied to gifted children mri analysis, Front. Neurosci. 16 (2022) 866735.

[20] K.-H. Oh, I.-S. Oh, U. Tsogt, J. Shen, W.-S. Kim, C. Liu, N.-I. Kang, K.-H. Lee, J. Sui, S.-W. Kim, et al., Diagnosis of schizophrenia with functional connectome data: a graph-based convolutional neural network approach, BMC Neurosci. 23 (1) (2022) 1–11.

[21] X. Chen, J. Zhou, P. Ke, J. Huang, D. Xiong, Y. Huang, G. Ma, Y. Ning, F. Wu, K. Wu, Classification of schizophrenia patients using a graph convolutional network: a combined functional MRI and connectomics analysis, Biomed. Signal Proces. 80 (2023) 104293.

[22] R. Yu, C. Pan, X. Fei, M. Chen, D. Shen, Multi-graph attention networks with bilinear convolution for diagnosis of schizophrenia, IEEE J. Biomed. Health. 27 (3) (2023) 1443–1454.

[23] M. Liu, H. Zhang, F. Shi, D. Shen, Hierarchical graph convolutional network built by multiscale atlases for brain disorder diagnosis using functional connectivity, IEEE T. Neur. Net. Lear. (2023).

[24] E.N. Pitsik, V.A. Maximenko, S.A. Kurkin, A.P. Sergeev, D. Stoyanov, R. Paunova, S. Kandilarova, D. Simeonova, A.E. Hramov, The topology of fMRI-based networks defines the performance of a graph neural network for the classification of patients with major depressive disorder, Chaos Solitons Fractals 167 (2023) 113041.

[25] S. Han, Z. Sun, K. Zhao, F. Duan, C.F. Caiafa, Y. Zhang, J. Solé-Casals, Early prediction of dementia using fMRI data with a graph convolutional network approach, J. Neural Eng. 21 (1) (2024) 016013.

[26] H. Zhang, R. Song, L. Wang, L. Zhang, D. Wang, C. Wang, W. Zhang, Classification of brain disorders in rs-fMRI via local-to-global graph neural networks, IEEE Trans. Med. Imaging 42 (2) (2022) 444–455.

[27] Z. Lu, I. Whalen, Y. Dhebar, K. Deb, E.D. Goodman, W. Banzhaf, V.N. Boddeti, Multiobjective evolutionary design of deep convolutional neural networks for image classification, IEEE Trans. Evol. Comput. 25 (2) (2020) 277–291.

[28] D. Sapra, A.D. Pimentel, Designing convolutional neural networks with constrained evolutionary piecemeal training, Appl. Intell. 52 (15) (2022) 17103–17117.

[29] Y. Gao, H. Yang, P. Zhang, C. Zhou, Y. Hu, Graphnas: Graph neural architecture search with reinforcement learning, 2019, arXiv preprint arXiv:1904.09981.

[30] K. Zhou, X. Huang, Q. Song, R. Chen, X. Hu, Auto-gnn: Neural architecture search of graph neural networks, Front. Big Data 5 (2022) 1029307.

[31] J. Chen, J. Gao, Y. Chen, B.M. Oloulade, T. Lyu, Z. Li, Auto-GNAS: A parallel graph neural architecture search framework, IEEE Trans. Parallel Distrib. Syst. 33 (11) (2022) 3117–3128.

[32] M. Shi, Y. Tang, X. Zhu, Y. Huang, D. Wilson, Y. Zhuang, J. Liu, Genetic-gnn: Evolutionary architecture search for graph neural networks, Knowl.-Based Syst. 247 (2022) 108752.

[33] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of the 5th International Conference on Learning Representations, 2017.

[34] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, Vol. 30, 2017.

[35] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations, 2018.

[36] R. Li, X. Yuan, M. Radfar, P. Marendy, W. Ni, T.J. O'Brien, P.M. Casillas-Espinosa, Graph signal processing, graph neural network and graph learning on biological data: a systematic review, IEEE Rev. Biomed. Eng. (2021).

[37] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–73.

[38] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.

[39] Z. Ying, D. Bourgeois, J. You, M. Zitnik, J. Leskovec, GNNExplainer: Generating explanations for graph neural networks, in: Advances in Neural Information Processing Systems, Vol. 32, 2019.

[40] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in Neural Information Processing Systems, Vol. 29, 2016.

[41] X. Bresson, T. Laurent, Residual gated graph convnets, 2017, arXiv preprint arXiv:1711.07553.

[42] F.A. Hashim, A.G. Hussien, Snake optimizer: A novel meta-heuristic optimization algorithm, Knowl.-Based Syst. 242 (2022) 108320.

[43] S.C. Tanaka, A. Yamashita, N. Yahata, T. Itahashi, G. Lisi, T. Yamada, N. Ichikawa, M. Takamura, Y. Yoshihara, A. Kunimatsu, et al., A multi-site, multi-disorder resting-state magnetic resonance image database, Sci. Data 8 (1) (2021) 227.

[44] M. Perrot, D. Rivière, J.-F. Mangin, Cortical sulci recognition and spatial normalization, Med. Image Anal. 15 (4) (2011) 529–550.

[45] C. Stam, P. Tewarie, E. Van Dellen, E. Van Straaten, A. Hillebrand, P. Van Mieghem, The trees and the forest: characterization of complex brain networks with minimum spanning trees, Int. J. Psychophysiol. 92 (3) (2014) 129–138.

[46] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.

[47] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.

[48] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.

[49] T.-Y. Eom, S.B. Han, J. Kim, J.A. Blundon, Y.-D. Wang, J. Yu, K. Anderson, D.B. Kaminski, S.M. Sakurada, S.M. Pruett-Miller, et al., Schizophrenia-related microdeletion causes defective ciliary motility and brain ventricle enlargement via microRNA-dependent mechanisms in mice, Nature Commun. 11 (1) (2020) 912.

[50] B. Franke, J.L. Stein, S. Ripke, V. Anttila, D.P. Hibar, K.J. Van Hulzen, A. Arias-Vasquez, J.W. Smoller, T.E. Nichols, M.C. Neale, et al., Genetic influences on schizophrenia and subcortical brain volumes: large-scale proof of concept, Nature Neurosci. 19 (3) (2016) 420–431.

[51] D. Mier, S. Eisenacher, F. Rausch, S. Englisch, M.F. Gerchen, V. Zamoscik, A. Meyer-Lindenberg, M. Zink, P. Kirsch, Aberrant activity and connectivity of the posterior superior temporal sulcus during social cognition in schizophrenia, Eur. Arch. Psychiatry Clin. Neurosci. 267 (7) (2017) 597–610.