DGX: Uncovering general behavior of deep graph models with model-level explanation

Jinlong Hu, Jiacheng Liu, Shoubin Dong, Bin Liao, Junjie Liang, Vasant Honavar

***Abstract*—Deep graph learning models have recently been developed to learn from various graphs that are prevalent in describing and modeling complex systems, including those in bioinformatics. However, a versatile explanation method for uncovering the general graph patterns that guide deep graph models in making predictions remains elusive. In this paper, we propose DGX, a novel deep graph model explainer that generates explanatory graphs to explain trained, black-box deep graph models. Its effectiveness is demonstrated by producing multiple graphs that collectively encode the structural knowledge captured by the graph neural network on both synthetic and real graph data. Importantly, DGX can produce diverse explanations by generating a set of distinguishable graphs and can provide customized explanations based on prior knowledge or constraints specified by users. We apply DGX to explain a mutagenicity prediction model by exploring the underlying groups of mutagenic compounds, and we explain the model on brain functional networks by revealing the structural patterns that enable the model to differentiate autism spectrum disorder from healthy controls. These findings offer an effective, diverse, and customized approach to explaining the underlying mechanisms and enhancing the understanding of models learned from real graph data, particularly in fields such as biomedicine and bioinformatics.**

***Index Terms*—Graph neural networks, model-level explanation, diversity, evolutionary algorithm**

I. Introduction

DEEP graph learning [1], [2] has been developed to extract insights from various graph-structured data in the fields of biomedicine and bioinformatics [3], [4], with applications in areas such as drug discovery [5], predicting metabolic pathways [6], and identifying molecular properties [7]. While these Deep Graph Models (DGMs) demonstrate exceptional learning capabilities, they often encounter challenges in providing transparency in explaining the mechanisms underlying their predictions due to their inherent complexity and the opaque internal processes involved in manipulating graph structures [8]–[10].

Recently, techniques from eXplainable Artificial Intelligence (XAI) [11] have been developed to provide understandable and transparent explanations for the predictions made by these models, particularly in the fields of drug discovery [12]–[14] and biological network analysis [15]–[17]. These explanation methods have been employed to explain DGMs at both the instance level [18]–[21] and the model level [22]–[26]. Instance-level explanation methods focus on interpreting the model's behavior based on specific input samples, while model-level methods aim to uncover the general behavior of the model by identifying input patterns that could lead to particular predictions [27]. The key issue of effective model-level explanation involves exploring the entire solution space, which depends on the structure of the graph (e.g., the adjacency matrix) and the features of the nodes. For example, given n nodes, there is a huge search space of $2^{n(n-1)/2}$ when only the structures are considered. It is obviously difficult to find a polynomial-time algorithm to accurately solve a non-convex optimization problem with exponential growth of scale. Meanwhile, generating effective model-level explanation also requires the algorithm to find a sufficient number of solutions, or uncover more possible graph patterns than those provided by existing solutions, which typically generate only one explanatory graph [22] to explain the DGMs. Furthermore, the existing approaches usually ignore priori domain knowledge and customized needs from users [28]. For example, similar to using high-level concepts for a prediction class (e.g., stripes for zebras) in general neural networks [29], it is essential to determine which structures of specified nodes or elements (e.g., one nitrogen atom and two oxygen atoms) can be predicted with a high probability as a class of nitro compounds in the model.

To address these challenges, we propose a novel deep graph model explainer, namely DGX, which effectively generates a series of diverse and customizable explanatory graphs that explain the DGMs at model level. Specifically, we formulate the task of generating explanatory graphs as a population optimization problem and employ an evolutionary algorithm to optimize this process, resulting in a set of diverse optimal solutions. Furthermore, DGX is a model-agnostic explanation method; the model being explained is treated as a predictive machine, providing fitness assessments for the optimization process within the explanation framework.

Our analysis demonstrates that DGX effectively generates explanatory graphs by producing multiple graphs that yield high predicted probabilities for the graph model. Additionally, DGX can create diverse explanations of the graph model by generating a set of distinguishable graphs. It also allows for customized explanations based on prior knowledge and constraints provided by users, such as given nodes and connectivity patterns within the graphs. Furthermore, in addition to offering effective, diverse, and customized explanations of

Jinlong Hu, Jiacheng Liu, and Shoubin Dong are with the Guangdong Provincial Key Laboratory of Multimodal Big Data Intelligent Analysis, School of Computer Science and Engineering, South China University of Technology, Guangzhou, China (E-mail: jlhu@scut.edu.cn, ljch1201@qq.com, sbdong@scut.edu.cn).

Bin Liao is with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China (E-mail: liaobin_lb@163.com).

Junjie Liang and Vasant Honavar are with the College of Information Sciences and Technology, Pennsylvania State University, PA, USA (e-mail: jliang282@outlook.com, vuh14@psu.edu).

Manuscript received January 24, 2024 (Corresponding author: Jinlong Hu).

how the DGM operates, DGX is particularly useful for users seeking to understand the general mechanisms of the model learned from real graph data or for debugging purposes. It also provides a valuable means to validate previous findings or generate potential hypotheses from the model's perspective [30].

II. Related Works

A. Deep Graph Models

In recent years, deep learning methods have demonstrated their expressive power to capture complex patterns from graph-structured data [1], [2], [31]. Graph Neural Networks (GNNs) are a powerful deep learning paradigm for graphs, excelling at learning informative representations in an end-to-end manner, such as Graph Convolutional Networks (GCNs) [32], GraphSAGE [33], Graph Attention Networks (GATs) [34], Graph Isomorphism Networks (GINs) [35], PTDNet [36], and NeuralSparse [37]. Deep graph learning has achieved remarkable success across various tasks, including node classification, graph classification, and link prediction [38]. In this paper, we explain DGMs with an emphasis on graph-level tasks. These models are typically designed based on message-passing or attention mechanisms, which learn dense node representations and extract graph representations for downstream prediction tasks through a permutation-invariant pooling approach.

B. XAI Methods for Deep Graph Models

Combining feature information with combinatorial graph structures enables DGMs to exhibit powerful nonlinear learning capabilities; however, this complexity also makes it more challenging to understand the intricacies of DGMs and the underlying reasons for their predictions. Existing explanation methods can be categorized into two groups based on the type of explanation provided: (local) instance-level explanations and (global) model-level explanations.

1) Instance-level explanation: Instance-level explanation methods typically focus on identifying the key features or subgraphs that contribute to model predictions, offering significant attribution specific to each input example [9]. These methods have garnered significant interest, resulting in the emergence of some highly effective techniques, such as GNNExplainer [18], SubgraphX [39], GraphLIME [40], CF-GNNExplainer [41], DyExplainer [40], and GCN-LRP [21]. For instance, GNNExplainer treats the explanation process as an optimization problem by generating a graph mask (or feature mask) that highlights the significant subgraph (or node features) relevant to the predictions made by GNN models.

2) Model-level explanation: Model-level methods aim to provide generic insights into understanding the decision-making processes of DGMs, by uncovering potential graph patterns that may lead to specific model behaviors, such as maximizing target predictions. XGNN [22] trains a deep reinforcement learning agent to generate explanatory graphs. The reward function guides the generator to incrementally add edges in order to maximize the prediction accuracy of the target graph. GNNInterpreter [23] employs a numerical optimization approach to produce global explanatory graphs, aiming to maximize the prediction probability of the target class. PGExplainer [19] utilizes a deep neural network to parameterize the explanation generation process, providing a method for generating multi-instance explanations. GLGExplainer [25] synthesizes multiple local explanations to create a global summary. These model-level explanation methods offer promising solutions for interpreting black-box models in deep graph learning. However, the exploration of the general behaviors of deep graph models, especially regarding diversity and customized explanations, remains insufficiently explored.

III. Methods

A. Problem Definition

Generally, we denote a graph as $G = (\mathcal{V}, \mathcal{E}, X)$, where $\mathcal{V} = \{n_i\}_1^N$ represents the set of nodes, $\mathcal{E} = \{(n_i, n_j) \,|n_i, n_j \in \mathcal{V} \wedge i \neq j\}$ represents the set of the edges in the graph. N and M denote the number of nodes and edges, respectively. $X \in R^{N \times F}$ is the F-dimensional feature matrix of these N nodes. Mathematically, the adjacency matrix is used to formally represent the structural information of a graph. For an unweighted graph, it is a symmetric binary matrix $A \in B^{N \times N}, B \in \{0, 1\}$. A_{ij} takes the value of 1 when there exists an edge connecting nodes i and j, otherwise 0. For weighted graph, the adjacency matrix is a symmetric real-valued matrix $A \in R^{N \times N}$, where A_{ij} denotes the weights of the edge (n_i, n_j).

In general, DGMs learn the structure and feature patterns from graph data and make predictions. Model-level explanation aims to find the underlying graph patterns that maximally contribute to the model making certain predictions. We assume a set of explanatory graphs, each representing a specific discriminative pattern corresponding to a model prediction.

Let $\mathcal{F}(\cdot)$ be a trained deep graph classification model, and $\{c_1, \ldots, c_l\}$ be the set of class labels. Given $\mathcal{F}(\cdot)$ and a class c_i, the expected output of the model-level explanation task is a set of graph patterns $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_K\}$ that maximize the predicted probability for this class. Let $\mathcal{G}_j = (A_j, X_j)$, where A_j denotes the adjacency matrix and X_j denotes the node features, and the explanatory graph generation task can be transformed into a search optimization procedure for the adjacency matrix A_j and corresponding node features X_j. Meanwhile, the set of generated graphs should be diverse or distinguishable. To generate a diverse population of explanatory graphs, we consider the combinatorial optimization objectives as shown in formula (1).

$$\mathcal{L} = \arg\min_{\mathcal{G}} \frac{1}{K} \sum_{j=1}^{K} yloss\left(\mathcal{F}\left(\mathcal{G}_j\right), c_i\right) - \lambda \, \mathcal{D}\left(\mathcal{G}\right) \quad (1)$$

where $yloss(\cdot)$ is the measure that minimizes the error between targeted class c_i and model's prediction for $\mathcal{G}_j$; and function $\mathcal{D}(\cdot)$ assesses the population diversity of $\mathcal{G}$, weighted by the hyper-parameter λ.

As illustrated in Fig. 1, we propose an evolutionary algorithm (EA)-based approach to address the population optimization problem outlined above. This method employs a graph generation procedure that utilizes given nodes to create

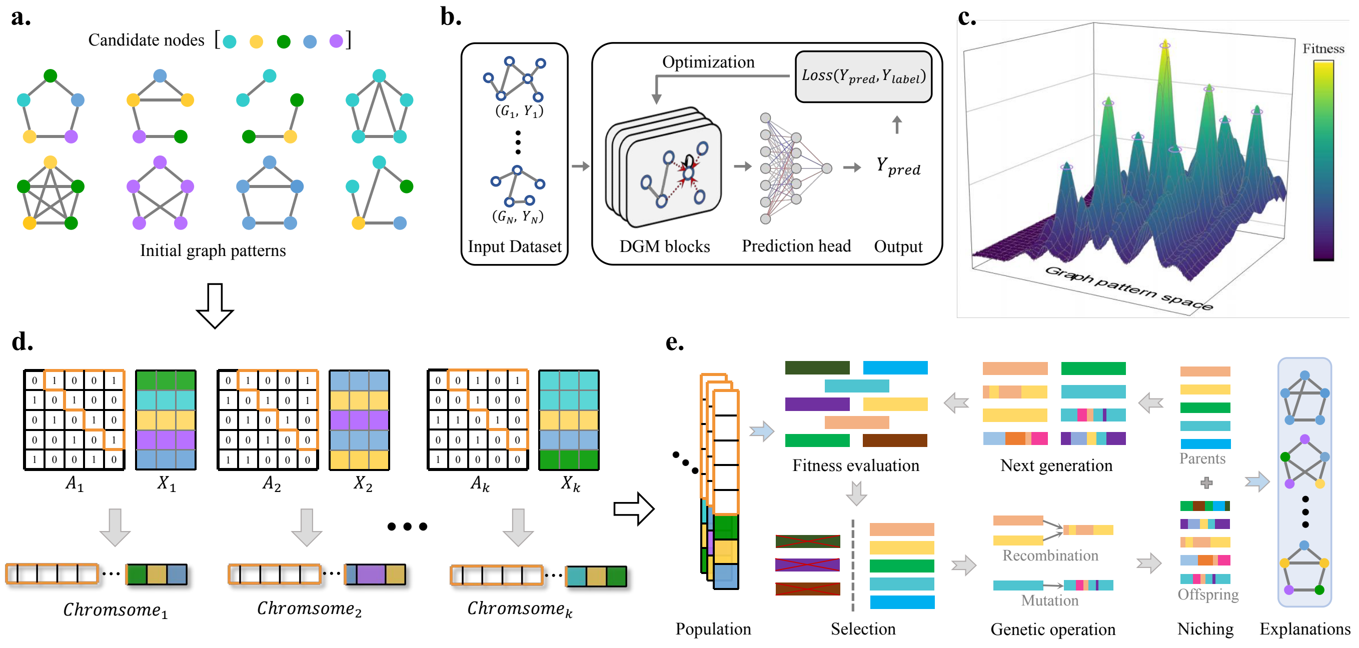


Fig. 1. **Overview of the model-level explanation approach. a**, In initialization phase, a population of k graphs is randomly initialized based on given nodes. **b**, The DGM to be explained was well trained on a source dataset, which later contributes to population's fitness assessment as well as the explaining objective. **c**, There are usually multiple peaks in the potential graph space of DGM, with locally optimal fitness for the explaining of model's certain behavior. **d**, The linear chromosomes are generated from the upper triangle of the adjacency matrix and the node features by directly transforming the two-dimensional structures into a one-dimensional format and concatenating them. In this example, the length of the resulting chromosome is 25, consisting of 10 elements from the adjacency matrix and 15 elements from the node features. **e**, After the fitness evaluation and parental selection, the new offspring population is produced through genetic operations like recombination and mutation, and niching helps retain distinct individuals according to dissimilarity assessment. Finally, an optimal population or a diverse set of explanatory graphs are obtained through evolution.

a population of explanatory graphs. In the first phase, a small set of candidate nodes is established, which may consist of specific nodes of interest identified by the user or any nodes selected from the model's input space. Additionally, the given nodes may include $null$ nodes, whose features can be learned during the optimization process. A population of input graphs is then randomly initialized based on the given candidate nodes (Fig. 1a). In the second phase, the adjacency matrix and node features of each input graph will be extracted to construct their genetic representations, which serve as chromosomes for subsequent algorithms (Fig. 1d). The targeted explained deep graph model acts as a predictive machine, estimating whether the generated graphs will yield high predicted probabilities within the model (Fig. 1e). In the third phase, a variant of the canonical evolutionary algorithm, known as the elitist evolutionary algorithm (EEA) [42], was developed to enhance convergence to the global optimal solution, regardless of the initial population [43], [44]. Additionally, niching methods [45] are employed alongside the EEA to preserve population diversity during the search for solutions. We utilize optimization objectives that promote diversity, as outlined in formula (1), to evaluate fitness in the evolutionary algorithm while searching for feasible solutions within the graph space. Furthermore, to produce customized explanations, we can adjust the fitness function and/or the constraint conditions based on prior knowledge or user interests, thereby guiding the evolutionary algorithm to identify satisfactory solutions and generate a population of explanatory graphs.

B. Evolutionary Algorithm for Generating Explanatory Graphs

DGX aims to uncover the general graph patterns that lead the deep graph model in making specific predictions by effectively searching for multiple explanatory graphs. Our approach iteratively optimizes these explanatory graphs using an evolutionary algorithm framework that incorporates an elitism strategy and a niching method [46]. This elitism strategy mitigates genetic drift by allowing the fittest individuals, known as elites, among the selection candidates to pass on their traits to subsequent generations. This strategy increases selective pressure, improves convergence speed [47], and enhances the algorithm's ability to converge on global optimal solutions, regardless of the initial population [43], [44]. The niching method [45], [48], [49] helps maintain population diversity by penalizing or eliminating similar individuals with lower fitness.

Given a trained GNN model $\mathcal{F}(\cdot)$ and a class of interest c, we aim to obtain a set of explanatory graphs that lead to prediction of c with high probability while ensuring that the graphs differ from one another. Our objective is to generate a set of explanatory graphs $\{\mathcal{G}_i\}^K$ for the deep graph model $\mathcal{F}(\cdot) = c$, where each graph contains a maximum of N nodes, and each node has F dimensional features. The procedure of our explanation framework includes the following components: initial population, fitness evaluation, selection method, recombination and mutation, niching method, convergence, and termination.

Initial Population From the perspective of evolutionary algorithms, we transform each unknown explanatory graph into a genetic representation (chromosome), and then iteratively optimize a set of chromosomes to obtain the optimal population. Chromosome is denoted as a structured vector, where each gene represents a group of variables to be optimized together. We formalize explanatory graph as a tuple (A, X), which encodes the graph structure and node properties respectively. As shown in Fig.1d, we flatten the upper triangle of adjacency matrix A as well as each column of feature matrix X, then concatenate these $F+1$ gene vectors to obtain a chromosome of length $L_C = \frac{N\times(N-1)}{2} + N \times F$. Finally, K chromosomes are created to compose the population and randomly initialized within the gene-wise value domain.

Fitness Evaluation The fitness function is used to measure the quality of each explanatory graph in population, and individual with higher fitness score is more likely to survive to the next generation. We can optimize the generation of explanatory graphs by extending the fitness function to meet users' customized explanation need, for example, sparsity term is used in fitness function to affect the denseness of topology structure of the generated graph, and help uncover the non-redundant structural patterns. For the i_{th} $(1 \leq i \leq K)$ individual $\mathcal{G}_i$, its fitness can be calculated by formula (2) and (3).

$$Fitness\left(\mathcal{G}_i\right) = \phi_c\left(\mathcal{F}\left(A_i, X_i\right)\right) + \mu\left(1 - \psi\left(A_i\right)^{\frac{1}{p}}\right) \tag{2}$$

$$\psi\left(A\right) = \frac{\sum_{i<j} A_{ij}}{N \times \left(N-1\right)/2} \tag{3}$$

Where $\phi_c\left(\cdot\right)$ denotes the validity score associated with predicting the target class c using the trained model $\mathcal{F}\left(\cdot\right)$, either the $\mathcal{L}_1$ or $\mathcal{L}_2$ loss can typically be employed. The second term represents the sparsity metric. Here, $\psi\left(\cdot\right)$ indicates the density of the adjacency matrix, and N is the dimensions (the number of nodes) of the adjacency matrix A. The hyperparameter p (where $p \geq 1$) acts as a scaling factor that enhances the sparsity term. This parameter is utilized to adjust the rate at which the sparsity score of the evaluation increases in relation to the edge density of the input graph. As the value of p increases, the derivative of the sparsity score also rises, resulting in a more rapid escalation of the sparsity score during the optimization of the objective function. In our experiments, p is set to 1.5. The parameter μ (where $\mu > 0$) serves as a weighting factor that balances the two components of the fitness function. This parameter adjusts the relative magnitude of the two terms through weighting; as the sparsity score increases, it constitutes a larger proportion of the fitness score, depending on the desired quality of the generated explanations. Typically, μ is set to 0.5 or 1.

Selective Method To progressively produce better offspring, we select a superior subset of the current population using a fitness-based elite tournament procedure. First, a specified number of ϵ individuals with the highest fitness scores are directly retained to maximize the preservation of optimal genes from the parental generation. Next, we conduct multiple tournaments by randomly selecting τ individuals (a fixed tournament size) from the remaining population and comparing the fitness values of the participants. This process yields θ distinct winners, referred to as elite individuals, who advance to the next stage of the evolutionary process. Elite tournament selection effectively promotes the survival of high-quality individuals, ensuring the preservation and potential enhancement of superior explanations across generations.

Recombination and Mutation DGX employs a series of genetic operators to facilitate population reproduction, enabling the algorithm to further explore and exploit the solution space. These operators consist of two types of transformations based on individual chromosomes: recombination and mutation. Recombination, also known as crossover, involves merging genetic information from two parent individuals to produce new offspring. In contrast, mutation introduces small random alterations to each individual, promoting exploration and preventing premature convergence.

For graph chromosomes consist of genes with multiple encodings and value domains, we can either perform gene-wise transformations independently by combining several simple operators or apply joint transformation operators that consider the dependencies among gene components, such as constraints on nodal attributes and structures in empirical graphs. We have developed a pair of recombination and mutation operators based on the constraints between node types and degrees, which transform chromosomes at the phenotype level rather than the vector level. Specifically, the proposed recombination operator swaps the positions corresponding to a selected node on two parental chromosomes, while the mutation operator jointly modifies the type and connections of a selected node at the corresponding positions on the parental chromosome. These two operators facilitate graph transformations that take node degree constraints into account. Detailed examples of recombination and mutation can be found in the supplementary materials.

Niching Method Niching methods are utilized in DGX to tackle the problem of premature convergence and to enhance the diversity of solutions within a population. The fundamental principle of niching methods involves modifying both the reproduction and selection processes. During selection, individuals are evaluated not only based on their fitness but also on their dissimilarity to others in the population. Consequently, we propose a crowding-based niching algorithm called Diversity Reinsertion. In this algorithm, the members of both the parent and offspring populations are combined and ranked according to their fitness. Each individual is then examined in descending order of fitness, and the individual that contributes the most to diversity is selected for the next generation. This fitness ranking and global competition effectively ensure the preservation of unique and promising solutions.

Niched evolutionary algorithms aim to identify a diverse array of feasible solutions within the search space by reducing or penalizing similar individuals in the population. The deterministic crowding method [50] employs direct competition between parent and offspring generations, allowing the offspring with higher fitness to replace a similar parent. However, local competition in complex scenarios may lead to insufficient diversity. In this study, we propose a new diversity reinsertion algorithm that integrates both parental and offspring populations while considering fitness and global competition. This approach effectively ensures that the new generation exhibits both the highest global fitness and enhanced diversity. The pseudo-code for the niching algorithm is presented in *Algorithm 1*.

Algorithm 1 Niching method: Diversity reinsertion

```
Input: population size $K$, Parent $\mathcal{G}_P$, Offspring $\mathcal{G}_O$, historical length $L$;
Output: next generation population $\mathcal{G}_N$;
 1: $\mathcal{G}_U \Leftarrow \mathit{Sorted}(\mathcal{G}_P \cup \mathcal{G}_O)$
 2: $\mathbf{D} \Leftarrow \mathit{Dissimilarity}(\mathcal{G}_U)$
 3: $\mathcal{S}, \mathcal{H} \Leftarrow [1], [0]$
 4: for $i = 2, \cdots, |\mathcal{G}_U|$ do
 5:     $\mathcal{Q} \Leftarrow \mathbf{D}[i, \mathcal{S}]$
 6:     $\mathbf{m}, \mathbf{n} \Leftarrow \arg\min(\mathcal{Q}), \arg\min(\mathcal{H})$
 7:     if $\mathcal{Q}[\mathbf{m}] > \mathcal{H}[\mathbf{n}]$ then
 8:         if $|\mathcal{H}| < L$ then
 9:             $\mathcal{H} \Leftarrow \mathcal{Q}[\mathbf{m}] \cup \mathcal{H}$
10:         else
11:             $\mathcal{H}[\mathbf{n}] \Leftarrow \mathcal{Q}[\mathbf{m}]$
12:         if $|\mathcal{S}| < K$ then
13:             $\mathcal{S} \Leftarrow i \cup \mathcal{S}$
14:         else
15:             $\mathcal{S}[\mathbf{m}] \Leftarrow i$
16: $\mathcal{G}_N \Leftarrow \mathcal{G}_U[\mathcal{S}]$
17: return $\mathcal{G}_N$
```

Where $\mathcal{G}_U$ represents the global population formed by combining the parent-offspring populations and sorting them according to the individuals' fitness in descending order, D denotes the distance matrix that captures the pairwise dissimilarity of $\mathcal{G}_U$, S is the array of indices of the selected individuals in $\mathcal{G}_U$, and the historical length L refers to the length of the historical distance buffer H, which is an array containing the top L distances between the selected individuals. Initially, the individual with the highest fitness is chosen. The algorithm then sequentially traverses $\mathcal{G}_U$, adding the current individual i to the set S if the distance between i and the individual indicated by S exceeds a specified threshold. In this context, the threshold is defined as the minimum distance among the selected individuals in the array H. The algorithm continues to iterate until K individuals with the best fitness and dissimilarity have been selected.

Convergence and Termination The initial population undergoes a process of evaluation, selection, reproduction, and niching techniques until it reaches convergence. Convergence can be assessed by monitoring whether the target population value stops showing significant increases across successive generations. Although the optimal solution can typically be obtained without limiting the number of iterations, the inherent randomness of the algorithm introduces uncertainty regarding the number of iterations needed to achieve the optimal solution. In our experimental design, we adopted the maximum allowed number of iterations as the termination condition, taking into account both the computational cost and the necessity for fairness in comparing the explanatory results of different methods.

C. Practical Considerations

Interactive and Customized Explanation DGX supports user interaction with model-level explanations by explicitly specifying which parts of the nodes in the explanatory graph should be preserved, including their node attributes and structural connections. Algorithmically, the positions corresponding to the specified nodes on the chromosome vector are initialized to fixed values and remain constant throughout the reproduction process. Additionally, DGX enables users to incorporate prior knowledge to construct a prophetic population, thereby guiding the evolutionary algorithm to efficiently search for customized or expected solutions for users. This process involves generating plausible explanations based on existing knowledge to replace the randomly initialized population. We believe that flexible interactions are crucial in enabling users to explore deep graph models from multiple perspectives, thereby facilitating a comprehensive understanding of model knowledge and behavior.

Fitness Assignment with Constraints The existing model-level explanation methods typically utilize predefined constraints to search for expected solutions. In DGX, we introduce new penalty functions to address the challenge of generating explanatory graphs with complex constraints. Each penalty function assigns a positive value to individuals that violate their constraints, where the punishment is proportional to the severity of the violation. This penalty is then incorporated into the fitness evaluation of EA. As shown in formula (4), ρ_k denotes the penalty function of the k_{th} constraint, which can be defined in any form that satisfies mapping feasible individuals to non-positive values and infeasible individuals to positive values. This approach discourages the selection of infeasible individuals and promotes convergence towards an optimal population that adheres to the specified constraints.

$$Penalty\left(\rho_k, \mathcal{G}_i\right) = \begin{cases} 0 & \rho_k\left(\mathcal{G}_i\right) \leq 0 \\ \rho_k\left(\mathcal{G}_i\right) & \rho_k\left(\mathcal{G}_i\right) > 0 \end{cases} \tag{4}$$

We address potential constraints based on the feasibility criterion and employ linear fitness scaling to determine the final individual fitness [46], [51]. Specifically, DGX integrates the fitness function and penalty function for each constraint to compute an objective value, ensuring that feasible individuals receive higher values than their non-feasible counterparts. Ultimately, the scaled fitness value is assigned to each individual based on the ranking of its objective value within the population.

Diversity and Graph Similarity To guide DGX to generate diverse explanations as much as possible, we apply the mean pairwise dissimilarity (MPD) to evaluate the diversity of the population. Specifically, MPD calculates the average distance between all pairs of explanatory graphs within the population.

$$Diversity\left(\mathcal{G}\right) = \frac{1}{C_K^2} \sum_{i<j}^{K} 1 - sim\left(\mathcal{G}_i, \mathcal{G}_j\right) \tag{5}$$

where we define the dissimilarity between individual i and j as $1 - sim(\mathcal{G}_i, \mathcal{G}_j)$, and C_K^2 denotes the number of distinct pairs in $\mathcal{G}$. $sim(\cdot)$ is the normalized graph similarity metric defined according to the specific dataset, considering that different datasets may provide various insights into the distance between graphs. There are some effective methods for graph comparison and similarity assessment, including graph edit distance, graph kernels, and graph matching [52]–[56]. In this paper, we adopt the subgraph matching kernel [57], a kernel-based method, as an example to calculate the similarity score for each pair of explanatory graphs and normalize it to the range $[0, 1]$. Then, the pairwise dissimilarity is obtained for the niching method and for assessing population diversity.

IV. Experiments

A. Dataset and Evaluation metrics

Datasets To validate our approach, we constructed a synthetic graph dataset with ground-truth motifs, referred to as MOT. The graph dataset comprises two classes, with each graph consisting of a randomly generated scale-free network and one type of ground-truth motif corresponding to its class, as noted in [18], [19]. The scale-free networks are undirected acyclic graphs generated randomly using the Barabási-Albert model, containing between 5 and 20 nodes from candidate types. As illustrated in Fig. 2a, the motifs (i.e., Ground Truth) of the two classes are designed based on the following rules: (i) The motifs of Class One consist of six graph patterns that adhere to the definition of mutual exclusion. Specifically, they include three distinct graph structures constructed from five nodes, which may consist of either a single node type or multiple node types. (ii) The motifs of Class Two are generated by randomly modifying the graph structure or node types from the motifs of Class One, i.e. adding new edges, deleting existing edges, and replacing nodes with either single or multiple node types.

In addition, we applied our approach to explain two real-world graph classification tasks: molecular property prediction using the Mutagenicity dataset [58], [59] and the identification of biological brain networks utilizing the Autism Brain Imaging Data Exchange I (ABIDE I) dataset [60]. The Mutagenicity dataset consists of chemical compounds categorized into two classes: mutagen and non-mutagen. In this dataset, molecules are represented as graphs, with atoms as nodes and chemical bonds as edges. The ABIDE I dataset is derived from functional magnetic resonance imaging (fMRI) data and focuses on functional brain networks, which highlight similar activation patterns across various brain regions and model their potential functional interactions. In this context, nodes represent distinct and unique regions of interest (ROIs) in the brain, while edges indicate the functional connectivities (FCs) between these ROIs. Specifically, we parcellated the brain into 116 ROIs based on the Anatomical Automatic Labeling (AAL) atlas [61]. To create the functional brain network dataset, we calculated the Pearson correlation coefficients for pairwise ROIs and constructed the adjacency matrix of the functional brain network using a threshold of the top 20% [62], [63].

The characteristics of the dataset and the classification performance are presented in Table I. In this table, #Graphs indicates the total number of samples, while #Node and #Edge represent the average number of nodes and edges, respectively. The F1-Score and AUC columns provide an evaluation of the classification performance of the trained graph models.

TABLE I
Dataset characteristics and classification performance

Datasets	#Graphs	#Node	#Edges	F1-Score	AUC
MOT	12000	22.45	26.63	0.9961	0.9998
Mutagenicity	4337	30.32	30.77	0.8559	0.8690
ABIDE I	871	116	1334	0.6438	0.7083

Evaluation Metrics Model-level explanations are often evaluated qualitatively through user observation alone [22]. In addition to this qualitative assessment, we employ five quantitative metrics to compare the performance of explanations: $Validity$, $Diversity$, $Hitness$, $Feasibility$, and $Fidelity$.

$Validity$ is defined as the average of the predicted probabilities of total generated explanatory graphs for the target class, which indicates the quality of the model-level explanations produced by the method in a specific explanatory task.

$$Validity(\mathcal{G}) = \frac{1}{K}\sum_{i=1}^{K} \phi_c(\mathcal{F}(\mathcal{G}_i)) \tag{6}$$

$Diversity$ of a set of explanatory graphs is measured with MPD, which is defined in Section III-C. MPD provides an approximate measure of diversity among individuals within the population and is designed to flexibly account for differences in attributes and structures by defining a graph similarity function.

$Hitness$ refers to the ratio of unique true positive patterns to the total number of ground-truth patterns, while $Feasibility$ is the ratio of unique true positive patterns to the total number of generated positive patterns. Essentially, these two metrics can be compared to recall and precision in binary classification tasks.

$$Hitness(\mathcal{G}) = \frac{\left|\{unique\ \mathcal{G}_i^+ \mid \mathcal{G}_i^+ \propto \mathcal{G}_{GT}\}\right|}{|\mathcal{G}_{GT}|} \tag{7}$$

$$Feasibility(\mathcal{G}) = \frac{\left|\{unique\ \mathcal{G}_i^+ \mid \mathcal{G}_i^+ \propto \mathcal{G}_{GT}\}\right|}{|\mathcal{G}^+|} \tag{8}$$

Where the generated graphs that are predicted to belong to the target class are referred to as positive patterns, represented as $\mathcal{G}^+$. Graphs that align with the ground-truth definition are categorized as true positive patterns, denoted as $\mathcal{G}_{GT}$. To apply these metrics, it is essential to have access to the ground-truth explanatory graph of the model, such as the MOT dataset. A higher $Hitness$ score suggests that the set of explanatory graphs produced by the explainer can cover a larger segment of all the model's explanatory graphs. Meanwhile, a higher $Feasibility$ score indicates that the explainer generates a greater number of ground-truth explanatory graphs.

$Fidelity$ refers to the robust fidelity measurement used to evaluate the correctness of explanations. This metric was originally defined to provide instance-level explanations for specific input cases, demonstrating its effectiveness in addressing out-of-distribution challenges [64]. We define $Fidelity$ as $Fid_{\alpha_1,\alpha_2,\Delta}$, as outlined in the original paper.In our model-level explanations, it is necessary to generate relevant input graphs derived from the explanatory graphs, as we do not have the specific input instances corresponding to the explanations within the context of model-level explanations. To ensure effective evaluations of $Fidelity$, it is essential that these generated input graphs align with the distribution of the original dataset.

B. Model Settings

Deep Graph Models In this paper, we employed GCNs and GINs for graph classification tasks on synthetic and empirical datasets. We designed the deep graph model in an end-to-end manner, utilizing a multi-layer GNN followed by a classifier. Each GNN layer learns high-level node embeddings through neighborhood message aggregation, and the graph representations are obtained from the embeddings of the final layer using a readout operation, specifically global attention pooling [65]. The classifier consists of fully connected networks with a softmax activation function, which outputs probability estimates for each category.

For the synthetic and molecular datasets, we constructed a four-layer GIN with a hidden dimension of 64. The output dimensions of the fully connected layers in the classifier are 128, 32, and 2. In contrast, for the functional brain network dataset, we developed a three-layer GCN with a hidden dimension of 32, and the output dimensions of the MLP are 64, 32, and 2. We trained the GNN models using 10-fold cross-validation and report the average metrics of F1-Score and ROC AUC in Table I. The results indicate that the predictive models to be explained all have acceptable classification performance. To mitigate the potential bias of a single-fold model, we adopted the $ConsensusModel$, i.e., the average predicting output of the 10-fold models, to guide DGX in generating model-level explanations.

Hyper-parameters As a post-hoc explanatory framework, DGX can be flexibly configured to generate diverse model-level explanations for deep graph models. In our experiments, we empirically set a population size of $k = 10$, an elite size of $\epsilon = 1$, and a tournament size of $\tau = 3$. We set the diversity coefficient $\lambda = 0.8$ for the optimization objective, $\mu = 0.5$ and $p = 1.5$ for the fitness function. Specifically, for brain functional network classification experiments, we adopt a small $\lambda = 0.2$ to motivate DGX to focus more on exploring high-confidence functional connectivity patterns, and we adjust μ to provide explanations under different sparse constraints. The population evolves over a maximum of iterations $epoch = 1000$.

The supplementary materials provide information about the server and software utilized in the experiments. The DGX source code can be accessed publicly on the project's website at https://github.com/largeapp/dgx.

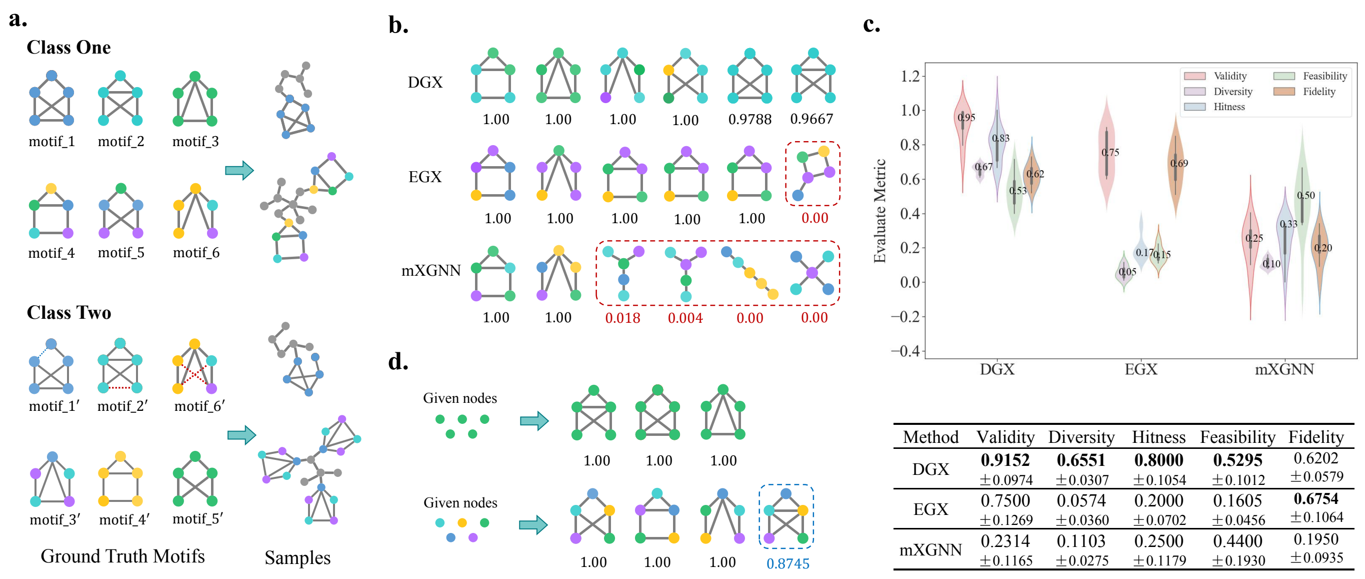
C. Diverse and Customized Explanations

Effective and Diverse Explanation Based on the MOT dataset, a GIN model [35] was trained to classify the graphs into two categories. To validate the effectiveness and diversity of our explanation approach, we explained the trained GIN model and compared it with two methods: (i) mXGNN, we extended the canonical model-level explanation method XGNN [22] to generate multiple graphs by repeatedly calling the original version of XGNN with random initialization at each iteration. We employed a probabilistic strategy for reinforcement learning in mXGNN, randomly selecting one action from the top three actions and using the probability of each action as the weight for random selection to generate more diverse graphs. (ii) EGX, a variant of our approach DGX, adopts only the elite policy without the niching method. We repeated each approach ten times with different random seeds to evaluate the robustness and stability of the explanations.

In Fig. 2b, the top six of ten explanatory graphs generated for Class One by DGX, EGX, and mXGNN are shown respectively. For DGX, the six explanatory graphs yield high predicted probabilities (nearly 1.00) with the GIN model and are inline with the six ground-truth patterns of the motifs in Class One. For EGX and mXGNN, five and two of the six graphs, respectively, yield high predicted probabilities with the GIN model, while one and four of the six graphs do not yield high predicted probabilities; the high-prediction graphs are inline with only two ground-truth motifs. The results demonstrate that all three methods can generate high-prediction explanatory graphs in terms of effective explanation. However, in terms of diverse explanation, only DGX generates sufficiently distinguishable explanatory graphs that include all six motifs of Class One, while both EGX and mXGNN only generate two motifs of Class One.

To quantitatively evaluate the diversity of explanations, we employed the five metrics mentioned above. The results are presented in Fig. 2c. We conducted ten explaining experiments and generated ten explanatory graphs for each method. $Validity$ refers to the average prediction probability of the ten generated graphs given by the trained GIN, while $Diversity$ measures the mean dissimilarity between each pair of generated graphs. $Hitness$ represents the proportion of unique true positive motifs relative to six, which is the ground-truth size of Class One. $Feasibility$ is defined as the proportion of unique true positive motifs among the total number of generated positive graphs. The pairwise dissimilarity is calculated as one minus the similarity between two corresponding graphs, with graph similarity measured using the well-established kernel method, specifically the subgraph matching kernel [57]. For $Fidelity$, we follow the construction rules of the MOT dataset and utilize each explanatory graph as a motif to randomly generate a corresponding scale-free Barabási-Albert graph as the input graph. We then calculate the $Fidelity$, referring to $Fid_{\alpha_1,\alpha_2,\Delta}$ as defined in [64], with $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$.

The results indicated that DGX achieved the best performance among the first four metrics. DGX was slightly inferior to EGX and significantly outperformed mXGNN in terms of fidelity. This may be attributed to the fact that EGX does not face the diversity limitations that DGX encounters when generating explanatory graphs. In general, DGX demonstrated the greatest diversity in its explanations while achieving excellent performance in other metrics.



Method	Validity	Diversity	Hitness	Feasibility	Fidelity
DGX	**0.9152** ±0.0974	**0.6551** ±0.0307	**0.8000** ±0.1054	**0.5295** ±0.1012	0.6202 ±0.0579
EGX	0.7500 ±0.1269	0.0574 ±0.0360	0.2000 ±0.0702	0.1605 ±0.0456	**0.6754** ±0.1064
mXGNN	0.2314 ±0.1165	0.1103 ±0.0275	0.2500 ±0.1179	0.4400 ±0.1930	0.1950 ±0.0935

Fig. 2. **Explanations with synthetic dataset**. **a**, the examples with two graph-level labels in the synthetic dataset MOT, we define Class One as the positive class, which includes six motif patterns that satisfy the mutual exclusion definition. Specifically, the first three patterns consist of specific structures containing only a single node type, while the last three represent distinct structures containing multiple node types. Class Two, as the negative class, comprises motifs generated by applying structural or nodal perturbations to the positive motifs. Each sample is created by connecting a random scale-free network to a positive or negative motif up to three times. **b**, the explanatory graphs for Class One generated by three methods for the trained GIN classification model, comparing DGX with EGX and mXGNN. We selected the top six out of the ten explanations produced by each method, and their validity scores (prediction probabilities) are displayed below. The graphs enclosed in the red dashed box represent invalid explanations, indicating that EGX and mXGNN were unable to generate six valid explanatory graphs for the GIN model. **c**, the quantitative evaluation of DGX, EGX, and mXGNN, includes ten rounds of explanation and assessment for each method. The tabular data calculates the mean and standard deviation of the five explanatory metrics, while the violin plot presents the statistical distribution of these metrics as a whole. **d**, the explanatory graphs generated by DGX for Class One under two sets of customized configurations: the first row features five green nodes, and the second row includes five nodes of distinct types. The graph within the blue dashed box is not derived from the ground-truth motifs but has a high prediction probability for Class One.

Customization Explanation The results presented in Fig. 2d demonstrate the effectiveness of our approach in generating customized explanations on the MOT dataset. We carried out two experimental cases with customized explanations using initial nodes provided by the user.

In the first case, the user aimed to investigate the structural patterns associated with five nodes of the same type, which were of particular interest to them, to determine how these patterns would lead the trained model to classify the graphs as Class One. As illustrated in the first row of Fig. 2d, the user provided five nodes of the same type (highlighted in green) to the DGX explainer. The explainer identified three graphs containing these five green nodes that corresponded to the ground-truth motifs of Class One.

In the second case, leveraging existing knowledge about various nodes, the user aimed to explore the structural patterns associated with five nodes of different types to determine whether the model would classify them as Class One. As illustrated in the second row of Fig. 2d, the user provided five nodes of different types (represented by various colors) to the DGX explainer. The explainer found three graph patterns that feature five distinct nodes that corresponded to the ground-truth motifs of Class One. Furthermore, DGX discovered another structure that, while not in accordance with the definition of ground truth, exhibited a high prediction probability of 0.8745 (the fourth graph in the second row of Fig. 2d). This finding suggests that DGX can help reveal potential insights within the decision boundaries of the trained model.

D. Explaining Deep Graph Models on Molecule Graphs

We trained a deep graph model on the Mutagenicity dataset to classify compounds as either mutagenic or non-mutagenic. We applied DGX to explain the model's decisions regarding mutagenic effects at two levels. On the one hand, our aim was to explain which atoms and structures could lead the model's predictions of mutagenicity with high probability. Specifically, we utilized DGX to generate explanatory graphs using 3 to 5 nodes selected from the 14 types of atoms in the dataset. The atomic types and bond connections of the nodes were optimized according to valency rules. We obtained three groups of chemical compounds containing 3, 4, and 5 atoms, respectively, as illustrated in Fig. 3b. The estimated mutagenic probabilities assigned by the trained GIN for these structures are close to 1, with functional groups such as amino, nitro, azide, and aldehyde particularly favorable in enhancing the model's predictions of mutagenicity. Most of these explanations were consistent with the findings of toxic groups in previous studies [66]–[69], although some remain questionable due to unverified structural authenticity or chemical properties.

Furthermore, we present a case study of the DGX explainer utilizing customized explanations for the deep molecule graph model. Researchers are interested in understanding how the general compounds attached to functional groups can lead the model to predict mutagenicity with high probability [71]. Functional groups are atoms or groups of atoms that determine

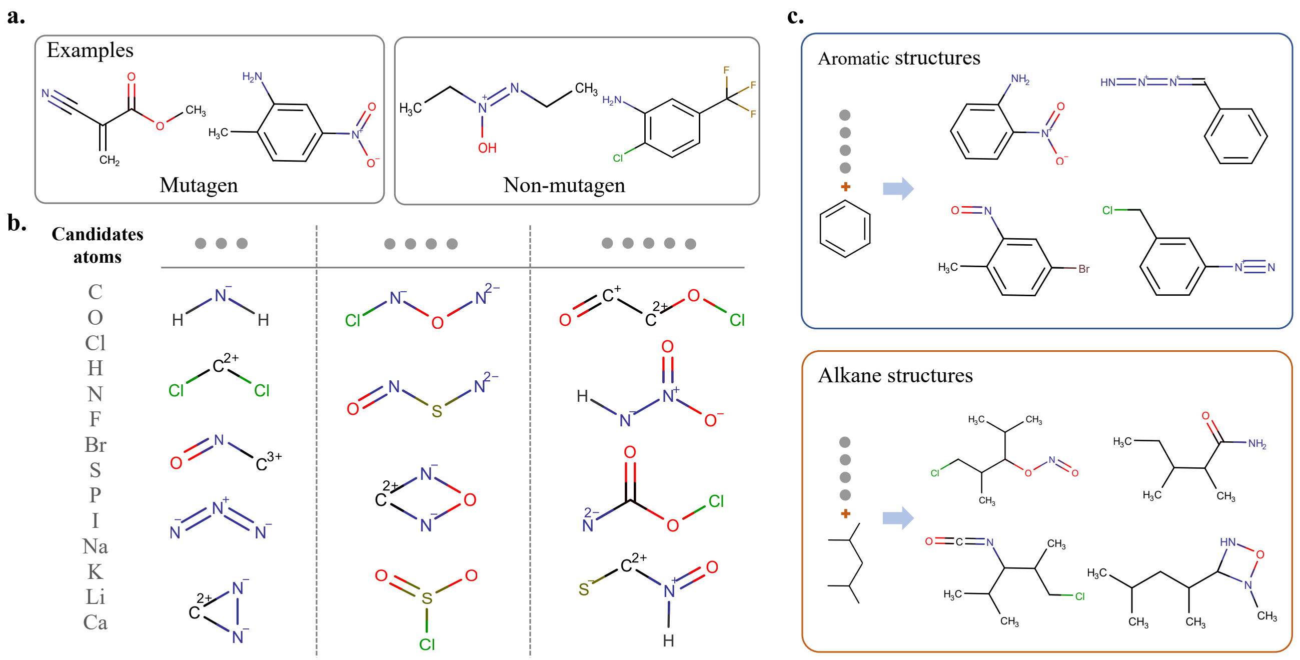


Fig. 3. **Explaining deep graph model on molecule graphs**. **a**, examples of two classes in the Mutagenicity dataset: mutagenic and non-mutagenic molecule graphs. These molecular graphs consist of nodes representing 14 different types of atoms and edges representing 3 types of chemical bonds, with each graph labeled according to its mutagenic effect. **b**, examples of customized diversity explanation results, with the number of nodes/atoms restricted to 3, 4, and 5 respectively, while the types of atoms and their combinations are not limited. All the generated graphs exhibit a high probability of predicting mutagenic toxicity using the GIN model. **c**, the two typical types of explanatory graphs (molecules) with a high probability of mutagenic toxicity ($Prob = 1.0$) contain aromatic ring or alkane chain structures. These chemical structures were plotted using the MarvinSketch tool [70].

the chemical properties of organic compounds. Therefore, we employed DGX to generate explanatory graphs based on two categories of compounds: aromatics and alkanes. We initialized the explanatory graphs with ten atoms, ensuring that six of them were carbon atoms while preserving their benzene or hexane structure throughout the optimization process. Two groups of the generated chemical compounds are presented in Fig. 3c. The results indicate that aromatic nitro, aromatic amino, and alkyl nitrite groups are more likely to lead to mutagenic predictions by the trained GIN model. More examples of explanations that utilize a priori information can be found in the supplementary materials.

E. Explaining Deep Graph Models on Functional Brain Networks

In the field of brain science, deep graph models have been developed to analyze brain functional networks, including the identification of brain disorders and enhancing our understanding of their functional connectivity patterns [72], [73]. The sparsity or density of these functional networks plays a crucial role in brain network analysis [74], where significant functional connections are expected to be both explicit and sparse [75]. Weak connections—often referred to as spurious or noisy—are typically eliminated using various thresholds to construct sparse brain networks [76].

In this study, we developed a simple graph convolutional model [32] to classify functional brain networks from Autism Spectrum Disorder (ASD) and Healthy Controls (HC) using the ABIDE-I dataset. We applied DGX to explain the predictions made by the trained GCN model regarding ASD in a customized manner, incorporating prior knowledge. The objective of the functional brain network explanation is to generate two sets of explanatory graphs with varying levels of sparsity. Specifically, we initialized a population of 116 ROI nodes with sparse connectivity and evolved it using discrete recombination and segment-moving mutation operators [77], [78]. As illustrated in Fig. 4, DGX produces two sets of explanatory graphs with different sparsity levels, along with high probability estimates for ASD, which represent distinct functional connectivity patterns derived from the model's cognition at various levels of brain region co-activation. In practice, two sets of brain networks were generated using different sparsity factors in the fitness function outlined in formula (1), specifically 0.2 and 0.1, respectively.

From the generated explanatory graphs, the functional connectivity patterns exhibit notable differences. Three connections (edges) frequently appear in these explanatory brain networks: the connection between Frontal_Inf_Oper_R and Cingulum_Mid_L, the connection between Frontal_Sup_Medial_L and Occipital_Inf_R, and the connection between Frontal_Sup_Medial_L and Cerebelum_4_5_L. There are eight ROIs (nodes) frequently involved in the generated brain networks: Frontal_Inf_Oper_R, Frontal_Inf_Tri_L, Olfactory_R, Frontal_Sup_Medial_L, Frontal_Sup_Medial_R, Hippocampus_R, ParaHippocampal_L, and ParaHippocampal_R. Some of these findings are consistent with previous findings, such as the significant ROIs of the superior frontal gyrus and ParaHippocampal_R

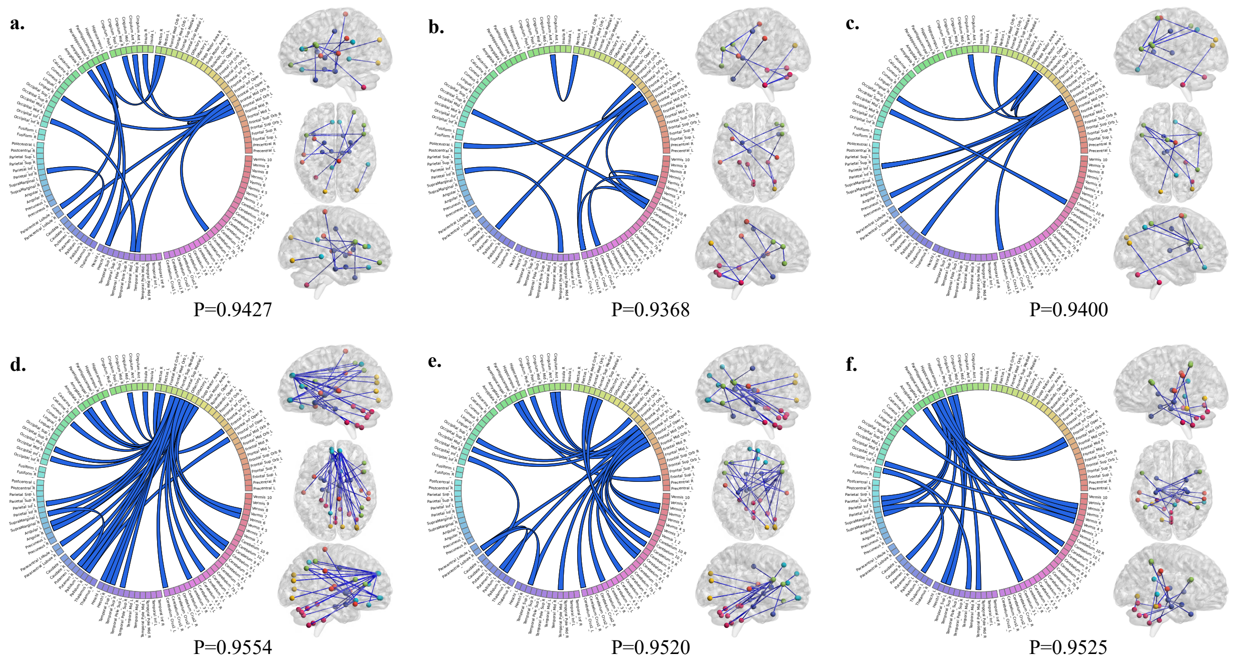


Fig. 4. **Explaining deep graph models for predicting ASD on brain functional networks**. Two level of sparsity of brain functional networks are generated, the sets of explanatory graphs (**a**, **b**, and **c**) and (**d**, **e**, and **f**) are generated by using 0.2 and 0.1 as the sparsity factor μ of the fitness function, respectively. Each sample demonstrates the explanatory brain functional networks with Circos diagram [79], and demonstrates these brain functional networks in the three views (left, top, and right) of brain with the BrainNet-Viewer tool [80]. For Circos diagrams, the 116 brain regions of AAL atlas are represented as ideogram squares on the circumference of the big circle, and the ribbons connecting two squares represent the functional connections between the corresponding brain regions.

[81], while others are new to previous studies, including three specific connections [73]. The results indicate the diversity of explanations of DGX, and its potential to validate prior findings and reveal new insights (graph patterns) that significantly contribute to the model's prediction.

V. Conclusion

In this study, we propose an evolutionary algorithm based approach to explain deep graph learning models at the model level. By combining an elitic niching model with evolutionary optimization, our approach is able to generate a diverse set of model-level explanations in a flexible manner, as well as personalized explanations according to user preferences. Our approach is model-agnostic, applicable to any single model or group of consensus models, and provides general insights into the model's class perception without requiring access to any parameters. Model-level explanations enable users to explore which significant nodes and their connections discriminatively influence predictions. The pursuit of diversity motivates the explainer to comprehensively reveal the various potential attributions behind model decisions. Additionally, diverse explanations can help diagnose potential model pitfalls, thereby offering reliable guidance for improving the predictive performance of the model [23]. Experiments on synthetic and real-world datasets validated DGX's global explaining capability and the power in terms of diversity and personalized explanations.

We frame the challenge of finding global explanations for DGMs as an optimization problem. Our approach utilizes an evolutionary algorithm that integrates elitist and niching strategies to tackle this issue, thereby avoiding the introduction of another black box in the process of explaining deep models [22], [25]. This methodology also facilitates the provision of diverse and customized explanations. However, it is not without drawbacks as it is influenced by the inherent limitations of evolutionary algorithms, including substantial computational demands and the risk of premature convergence. Therefore, it is essential to evaluate the effectiveness of our explanation approach, particularly when explaining models with input graphs that contain a large number of nodes. In practice, the employing of efficient chromosome encoding and genetic operators, along with the implementation of parallel strategies, can help alleviate the inherent shortcomings of evolutionary algorithms [82]. Furthermore, weakly constrained high-probability explanations tend to provide the rationale for mathematical reasoning rather than offering faithful attribution based on actual data distributions [23], [28]. In instances where the model has not accurately learned the true patterns due to insufficient or biased data, the generated explanatory graphs may struggle to accurately depict the model decision-making mechanisms [23], [71].

We emphasize that addressing the challenges of model-level explanation and its application to real-world domains simultaneously is difficult [83]. Interactive explorations that incorporate domain knowledge and specific constraints significantly improve the understanding of the behavior of DGMs. Consequently, DGX is designed as a straightforward and extensible explanatory framework that allows users to flexibly select specific operations, such as chromosome encoding, genetic operators, dissimilarity assessment, and fitness functions with constraints. This flexibility enables comprehensive explanations for extensive graphs with complex specifications and pattern definitions.

Acknowledgments

This research was supported by the Natural Science Foundation of Guangdong Province, grant numbers 2021A1515011942 and 2025A1515011595; and the Innovation Fund of the Zhongshan High-End Scientific Research Institutions, grant number 2019AG031. The authors thank Tenghui Li for his contribution to the initial work when he was at South China University of Technology.

References

[1] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 249–270, 2020.

[2] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu, "Graph learning: A survey," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 2, pp. 109–127, 2021.

[3] M. M. Li, K. Huang, and M. Zitnik, "Graph representation learning in biomedicine and healthcare," *Nature Biomedical Engineering*, vol. 6, no. 12, pp. 1353–1369, 2022.

[4] X.-M. Zhang, L. Liang, L. Liu, and M.-J. Tang, "Graph neural networks and their current applications in bioinformatics," *Frontiers in genetics*, vol. 12, p. 690049, 2021.

[5] F. Wong, E. J. Zheng, J. A. Valeri, N. M. Donghia, M. N. Anahtar, S. Omori, A. Li, A. Cubillos-Ruiz, A. Krishnan, W. Jin *et al.*, "Discovery of a structural class of antibiotics with explainable deep learning," *Nature*, pp. 1–9, 2023.

[6] J. Chen, J. Gao, T. Lyu, B. M. Oloulade, and X. Hu, "Automsr: Auto molecular structure representation learning for multi-label metabolic pathway prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2022.

[7] B. Zhang, C. Luo, H. Jiang, S. Feng, X. Li, B. Zhang, and Y. Ye, "Adaptive transfer of graph neural networks for few-shot molecular property prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2023.

[8] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability methods for graph convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10 772–10 781.

[9] H. Yuan, H. Yu, S. Gui, and S. Ji, "Explainability in graph neural networks: A taxonomic survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5782–5799, 2023.

[10] J. Kakkad, J. Jannu, K. Sharma, C. Aggarwal, and S. Medya, "A survey on explainability of graph neural networks," *arXiv preprint arXiv:2306.01958*, 2023.

[11] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[12] J. Jiménez-Luna, F. Grisoni, and G. Schneider, "Drug discovery with explainable artificial intelligence," *Nature Machine Intelligence*, vol. 2, no. 10, pp. 573–584, 2020.

[13] I. Ponzoni, J. A. Páez Prosper, and N. E. Campillo, "Explainable artificial intelligence: A taxonomy and guidelines for its application to drug discovery," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 13, no. 6, 2023.

[14] R. Alizadehsani, S. S. Oyelere, S. Hussain, S. K. Jagatheesaperumal, R. R. Calixto, M. Rahouti, M. Roshanzamir, and V. H. C. De Albuquerque, "Explainable artificial intelligence for drug discovery and development: A comprehensive survey," *IEEE Access*, vol. 12, pp. 35 796–35 812, 2024.

[15] M. Kiani, J. Andreu-Perez, H. Hagras, S. Rigato, and M. L. Filippetti, "Towards understanding human functional brain development with explainable artificial intelligence: Challenges and perspectives," *IEEE Computational Intelligence Magazine*, vol. 17, no. 1, pp. 16–33, 2022.

[16] S. Nazir, D. M. Dickson, and M. U. Akram, "Survey of explainable artificial intelligence techniques for biomedical imaging with deep neural networks," *Computers in Biology and Medicine*, vol. 156, p. 106668, 2023.

[17] J. Hu, J. Luo, Z. Xu, B. Liao, S. Dong, B. Peng, and G. Hou, "Spatio-temporal learning and explaining for dynamic functional connectivity analysis: Application to depression," *Journal of Affective Disorders*, vol. 364, pp. 266–273, 2024.

[18] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "Gnnexplainer: Generating explanations for graph neural networks," *Advances in neural information processing systems*, vol. 32, 2019.

[19] D. Luo, T. Zhao, W. Cheng, D. Xu, F. Han, W. Yu, X. Liu, H. Chen, and X. Zhang, "Towards inductive and efficient explanations for graph neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5245–5259, 2024.

[20] M. Vu and M. T. Thai, "Pgm-explainer: Probabilistic graphical model explanations for graph neural networks," *Advances in neural information processing systems*, vol. 33, pp. 12 225–12 235, 2020.

[21] J. Hu, T. Li, and S. Dong, "Gcn-lrp explanation: exploring latent attention of graph convolutional networks," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[22] H. Yuan, J. Tang, X. Hu, and S. Ji, "Xgnn: Towards model-level explanations of graph neural networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 430–438.

[23] X. Wang and H.-W. Shen, "Gnninterpreter: A probabilistic generative model-level explanation for graph neural networks," *arXiv preprint arXiv:2209.07924*, 2022.

[24] Z. Huang, M. Kosan, S. Medya, S. Ranu, and A. Singh, "Global counterfactual explainer for graph neural networks," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 2023, pp. 141–149.

[25] S. Azzolin, A. Longa, P. Barbiero, P. Liò, and A. Passerini, "Global explainability of gnns via logic combination of learned concepts," *arXiv preprint arXiv:2210.07147*, 2022.

[26] H. Xuanyuan, P. Barbiero, D. Georgiev, L. C. Magister, and P. Liò, "Global concept-based interpretability for graph neural networks via neuron analysis," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, 2023, pp. 10 675–10 683.

[27] G. Lv and L. Chen, "On data-aware global explainability of graph neural networks," *Proceedings of the VLDB Endowment*, vol. 16, no. 11, pp. 3447–3460, 2023.

[28] T.-C. Bui, W.-s. Li, and S.-K. Cha, "Pgx: A multi-level gnn explanation framework based on separate knowledge distillation processes," *arXiv preprint arXiv:2208.03075*, 2022.

[29] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas *et al.*, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," in *International conference on machine learning*. PMLR, 2018, pp. 2668–2677.

[30] R. M. Cichy and D. Kaiser, "Deep neural networks as scientific models," *Trends in Cognitive Sciences*, vol. 23, no. 4, pp. 305–317, 2019.

[31] Y. Rong, T. Xu, J. Huang, W. Huang, H. Cheng, Y. Ma, Y. Wang, T. Derr, L. Wu, and T. Ma, "Deep graph learning: Foundations, advances and applications," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 3555–3556.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[33] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[34] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio *et al.*, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.

[35] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.

[36] D. Luo, W. Cheng, W. Yu, B. Zong, J. Ni, H. Chen, and X. Zhang, "Learning to drop: Robust graph neural network via topological denoising," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, p. 779–787.

[37] C. Zheng, B. Zong, W. Cheng, D. Song, J. Ni, W. Yu, H. Chen, and W. Wang, "Robust graph representation learning via neural sparsification," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, 13–18 Jul 2020, pp. 11 458–11 468.

[38] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions*

on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 4–24, 2021.

[39] H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji, "On explainability of graph neural networks via subgraph explorations," in *International conference on machine learning*. PMLR, 2021, pp. 12 241–12 252.

[40] Q. Huang, M. Yamada, Y. Tian, D. Singh, and Y. Chang, "Graphlime: Local interpretable model explanations for graph neural networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 7, pp. 6968–6972, 2023.

[41] A. Lucic, M. A. Ter Hoeve, G. Tolomei, M. De Rijke, and F. Silvestri, "Cf-gnnexplainer: Counterfactual explanations for graph neural networks," in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Camps-Valls, F. J. R. Ruiz, and I. Valera, Eds., vol. 151. PMLR, 28–30 Mar 2022, pp. 4499–4511. [Online]. Available: https://proceedings.mlr.press/v151/lucic22a.html

[42] D. Dumitrescu, B. Lazzerini, L. C. Jain, and A. Dumitrescu, *Evolutionary Computation*. CRC Press, 2000.

[43] G. Rudolph, "Convergence analysis of canonical genetic algorithms," *IEEE transactions on neural networks*, vol. 5, no. 1, pp. 96–101, 1994.

[44] D. Bhandari, C. Murthy, and S. K. Pal, "Genetic algorithm with elitist model and its convergence," *International journal of pattern recognition and artificial intelligence*, vol. 10, no. 06, pp. 731–747, 1996.

[45] S. W. Mahfoud, *Niching methods for genetic algorithms*. University of Illinois at Urbana-Champaign, 1995.

[46] e. Jazzbin, "geatpy: The genetic and evolutionary algorithm toolbox with high performance in python," 2020. [Online]. Available: http://www.geatpy.com/

[47] C. W. Ahn and R. Ramakrishna, "Elitism-based compact genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 4, pp. 367–385, 2003.

[48] J. Horn, *The nature of niching: Genetic algorithms and the evolution of optimal, cooperative populations*. University of Illinois at Urbana-Champaign, 1997.

[49] D. Gupta and S. Ghafir, "An overview of methods maintaining diversity in genetic algorithms," *International journal of emerging technology and advanced engineering*, vol. 2, no. 5, pp. 56–60, 2012.

[50] O. J. Mengshoel and D. E. Goldberg, "The crowding approach to niching in genetic algorithms," *Evolutionary computation*, vol. 16, no. 3, pp. 315–354, 2008.

[51] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[52] L. A. Zager and G. C. Verghese, "Graph similarity scoring and matching," *Applied mathematics letters*, vol. 21, no. 1, pp. 86–94, 2008.

[53] P. Yanardag and S. Vishwanathan, "Deep graph kernels," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1365–1374.

[54] G. Ma, N. K. Ahmed, T. L. Willke, D. Sengupta, M. W. Cole, N. B. Turk-Browne, and P. S. Yu, "Deep graph similarity learning for brain data analysis," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2743–2751.

[55] N. M. Kriege, F. D. Johansson, and C. Morris, "A survey on graph kernels," *Applied Network Science*, vol. 5, no. 1, pp. 1–42, 2020.

[56] G. Ma, N. K. Ahmed, T. L. Willke, and P. S. Yu, "Deep graph similarity learning: A survey," *Data Mining and Knowledge Discovery*, vol. 35, pp. 688–725, 2021.

[57] N. Kriege and P. Mutzel, "Subgraph matching kernels for attributed graphs," *arXiv preprint arXiv:1206.6483*, 2012.

[58] K. Riesen and H. Bunke, "Iam graph database repository for graph based pattern recognition and machine learning," in *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, SSPR & SPR 2008, Orlando, USA, December 4-6, 2008. Proceedings*. Springer, 2008, pp. 287–297.

[59] J. Kazius, R. McGuire, and R. Bursi, "Derivation and validation of toxicophores for mutagenicity prediction," *Journal of medicinal chemistry*, vol. 48, no. 1, pp. 312–320, 2005.

[60] A. Di Martino, C.-G. Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, J. S. Anderson, M. Assaf, S. Y. Bookheimer, M. Dapretto *et al.*, "The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism," *Molecular psychiatry*, vol. 19, no. 6, pp. 659–667, 2014.

[61] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot, "Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain," *Neuroimage*, vol. 15, no. 1, pp. 273–289, 2002.

[62] D. Ahmedt-Aristizabal, M. A. Armin, S. Denman, C. Fookes, and L. Petersson, "Graph-based deep learning for medical diagnosis and analysis: past, present and future," *Sensors*, vol. 21, no. 14, p. 4758, 2021.

[63] A. Bessadok, M. A. Mahjoub, and I. Rekik, "Graph neural networks in network neuroscience," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5833–5848, 2022.

[64] Z. Xu, S. Farhad, W. Tianchun, C. Wei, C. Zhuomin, C. Haifeng, W. Hua, and L. Dongsheng, "Towards robust fidelity for evaluating explainability of graph neural networks," *arXiv preprint arXiv: 2310.01820*, 2024.

[65] Y. Li, C. Gu, T. Dullien, O. Vinyals, and P. Kohli, "Graph matching networks for learning the similarity of graph structured objects," in *International conference on machine learning*. PMLR, 2019, pp. 3835–3845.

[66] C. Xu, F. Cheng, L. Chen, Z. Du, W. Li, G. Liu, P. W. Lee, and Y. Tang, "In silico prediction of chemical ames mutagenicity," *Journal of chemical information and modeling*, vol. 52, no. 11, pp. 2840–2847, 2012.

[67] Z. Wu, D. Jiang, J. Wang, C.-Y. Hsieh, D. Cao, and T. Hou, "Mining toxicity information from large amounts of toxicity data," *Journal of Medicinal Chemistry*, vol. 64, no. 10, pp. 6924–6936, 2021.

[68] N. G. Bakhtyari, G. Raitano, E. Benfenati, T. Martin, and D. Young, "Comparison of in silico models for prediction of mutagenicity," *Journal of Environmental Science and Health, Part C*, vol. 31, no. 1, pp. 45–66, 2013.

[69] K. Hansen, S. Mika, T. Schroeter, A. Sutter, A. Ter Laak, T. Steger-Hartmann, N. Heinrich, and K.-R. Muller, "Benchmark data set for in silico prediction of ames mutagenicity," *Journal of chemical information and modeling*, vol. 49, no. 9, pp. 2077–2081, 2009.

[70] P. Csizmadia, *MarvinSketch and MarvinView: molecule applets for the World Wide Web*. Molecular Diversity Preservation International, 1999.

[71] Z. Wu, J. Wang, H. Du, D. Jiang, Y. Kang, D. Li, P. Pan, Y. Deng, D. Cao, C.-Y. Hsieh *et al.*, "Chemistry-intuitive explanation of graph neural networks for molecular property prediction with substructure masking," *Nature Communications*, vol. 14, no. 1, p. 2585, 2023.

[72] M. M. Li, K. Huang, and M. Zitnik, "Graph representation learning in biomedicine and healthcare," *Nature Biomedical Engineering*, vol. 6, no. 12, pp. 1353–1369, 2022.

[73] J. Hu, L. Cao, T. Li, S. Dong, and P. Li, "Gat-li: a graph attention network based learning and interpreting method for functional brain network classification," *BMC bioinformatics*, vol. 22, no. 1, pp. 1–20, 2021.

[74] P. A. Taylor, R. C. Reynolds, V. Calhoun, J. Gonzalez-Castillo, D. A. Handwerker, P. A. Bandettini, A. F. Mejia, and G. Chen, "Highlight results, don't hide them: Enhance interpretation, reduce biases and improve reproducibility," *NeuroImage*, vol. 274, p. 120138, 2023.

[75] A. Fornito, A. Zalesky, and M. Breakspear, "Graph analysis of the human connectome: promise, progress, and pitfalls," *Neuroimage*, vol. 80, pp. 426–444, 2013.

[76] J. D. Power, A. L. Cohen, S. M. Nelson, G. S. Wig, K. A. Barnes, J. A. Church, A. C. Vogel, T. O. Laumann, F. M. Miezin, B. L. Schlaggar *et al.*, "Functional network organization of the human brain," *Neuron*, vol. 72, no. 4, pp. 665–678, 2011.

[77] M. Zbigniew, "Genetic algorithms+ data structures= evolution programs," *Comput Stat*, pp. 372–373, 1996.

[78] H. Mühlenbein and D. Schlierkamp-Voosen, "Predictive models for the breeder genetic algorithm i. continuous parameter optimization," *Evolutionary computation*, vol. 1, no. 1, pp. 25–49, 1993.

[79] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra, "Circos: an information aesthetic for comparative genomics," *Genome research*, vol. 19, no. 9, pp. 1639–1645, 2009.

[80] M. Xia, J. Wang, and Y. He, "Brainnet viewer: a network visualization tool for human brain connectomics," *PloS one*, vol. 8, no. 7, p. e68910, 2013.

[81] C. S. Monk, S. J. Peltier, J. L. Wiggins, S.-J. Weng, M. Carrasco, S. Risi, and C. Lord, "Abnormalities of intrinsic functional connectivity in autism spectrum disorders,," *NeuroImage*, vol. 47, no. 2, pp. 764–772, 2009.

[82] N. Nedjah, E. Alba, and L. d. M. Mourelle, *Parallel Evolutionary Computations (Studies in Computational Intelligence)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[83] Y. Nian, W. Jin, and L. Lin, "In-process global interpretation for graph learning via distribution matching," *arXiv preprint arXiv:2306.10447*, 2023.