PLOS COMPLEX SYSTEMS

RESEARCH ARTICLE

Determining graphlet explanations for machine learning on graphs

Bettina Soós*, Gonzalo Nápoles, Pieter Spronck, Çiçek Güven

Department Cognitive Science and Artificial Intelligence, Tilburg School of Humanities and Digital Sciences, Tilburg University, Tilburg, The Netherlands

* b.soos@tilburguniversity.edu

OPEN ACCESS

Citation: Soós B, Nápoles G, Spronck P, Güven Ç (2025) Determining graphlet explanations for machine learning on graphs. PLOS Complex Syst 2(8): e0000067. https://doi.org/10.1371/journal.pcsy.0000067

Editor: Réka Albert, Pennsylvania State University, UNITED STATES OF AMERICA

Received: March 31, 2025

Accepted: August 4, 2025

Published: August 26, 2025

Data availability statement: All data are in the manuscript and/or supporting information files.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Abstract

This study introduces a post-hoc approach for explainable machine learning on graph-structured data by identifying relevant subgraphs and their corresponding subgraph patterns, the graphlet motifs. Unlike traditional motif detection methods, which rely solely on statistical occurrence frequencies, and therefore can be decoupled from the learning task, our method optimizes important motifs based on fidelity and sparsity, together with sufficiency and necessity, which are deemed key properties when generating explanations for the learning task. The resulting motifs are relevant due to being explanatory in the prediction task and their subgraph coverings correspond to the explanatory subgraphs. The method outperforms state-of-the-art techniques when comparing performance on the explanatory subgraph. Being an NP-hard problem, without constraining motif structure (e.g., fixing motif size), finding subgraph monomorphs of any form, and for all possible motifs, is computationally difficult. Hence, a genetic algorithm is used to search the possible subgraph space with the properties of desirable explanations. Fidelity ensures that the selected subgraphs maintain predictive power, which means that marginalizing or altering the subgraph would significantly impact the model's output. Sparsity guarantees that the explanation is as concise as possible, hence avoiding redundant or overly complex subgraphs while still capturing the core reasoning behind the prediction. Minimizing fidelity on the part of the graph that is not in the explanation (the non-covering subgraph) supports that the explanations are not only sufficient, but also necessary. We frame the search for explanatory graphlets as a multi-objective optimization problem that balances these properties. The methodology is demonstrated with single motifs as building blocks of subgraph explanations and evaluated on two complementary synthetic datasets designed for graph prediction tasks, also in comparison with state-of-the-art methodologies. The motifs found not only cover relevant structural patterns but also contribute meaningfully to the model's decision-making process.

Author summary

In this work we present an explainability method for machine learning on graphs. Taking neural networks as machine learning models obscures the interpretability of the models' prediction, given the complicated functions with large number of parameters they implement. Graphs represent entities and their relations that are increasingly used as primary data structures to represent data for machine learning. However, due to the complex nature of graph-structured data, additional considerations need to be taken when applying explainability methods directly on this domain. We use the genetic algorithm to find relevant substructures in the graph that explain the predictions. The explanations remain higher-order with searching for the combination of nodes and edges, i.e., subgraphs, relevant for the predictions. Furthermore, not only the relevant subgraph in the graph, but the repeating pattern in the subgraph, i.e., the graphlet motif, is analyzed for explainability. The genetic algorithm effectively explores the search space and finds meaningful explanations, which are evaluated with three neural networks on two synthetic datasets.

1. Introduction

Explainability and interpretability are important components of machine learning in applications where the output from the machine learning model has to be reasoned or the model's behavior has to be interpreted [1–3]. For a machine learning model, an explanation could be in form of a series of decisions that are made within the process to come to a specific outcome, or a collection of input components or features that the model focuses on while making a decision, or a description of what could be changed to reach to a different outcome. Explanations are possible to provide within the model, e.g., through activations and weights, or through the input data, e.g., by creating saliency maps and measuring feature importance. Here, explanations are searched in the input data space and the explanation methodology is model-agnostic. That is, different machine learning models can be analyzed with the algorithm to explain the predicted label or the true label.

Machine learning on graphs uses graphs as input data, enabling the model to learn and infer features, properties, and attributes from their structure [4]. There are similar explainability methods present for graphs as for other domains, and there are methods that incorporate graph structural information, feature information, or information of other graph properties (e.g., spectrum, random walk distribution) [5,6]. To explain a label, inference, or prediction from graph data, specific graph components - such as nodes, edges, or subgraphs are highlighted as relevant, contributing, or important elements that support the target value.

Graph *motif*s are graphlets that constitute relevant subgraph patterns in a graph. The relevant subgraph may reoccur in the graph more often than in an arbitrarily randomized version of the graph, and hence may be classified as a traditional graph *motif* [7]. However, the more a graphlet is relevant by itself, the less it has to reoccur to be considered a subgraph pattern as relevant as a more frequent but less important graphlet. Measuring relevance through the relative frequency of isomorphic subgraphs alone is not sufficient to identify symbolic subgraphs that are inherently relevant due to some functional property. Machine learning, however, exactly aims to identify functional relevance, i.e., learns a property given a set of inputs, some of which are more or less relevant for the prediction of the property. Relevant subgraph patterns are symbolic to the problem and can be considered functional motifs due to their inherent relevance rather than their frequency of occurrence. Explanatory power can determine whether a graphlet is a functional motif in the graph and can be used to determine how

prominent the motif in the graph compared to other motifs. The approach provides a motif definition by integrating structural properties and attributes of graphs, implicitly, through optimizing subgraph patterns for the properties listed below using a genetic algorithm.

Metrics used to evaluate good explanations in graph level tasks are similar to those used to explain predictions in other machine learning domains [8,9]. Namely, desirable properties that evaluate successful explanations are,

- **fidelity**: prediction on the explanations are is close to the original,
- **sparseness**: explanations are smaller than the original descriptors,
- **sanity**: sensitivity to model training and different models,
- **contrastivity**: explanations are target and output discriminative.
- **sufficiency and necessity**: the explanation is sufficient to explain the output or target, and necessary, such that there is no other component that can explain similarly.

Fidelity and sparsity are the measures used most often for quantitative evaluation of explanations [10–12]. Explanations with high fidelity are faithful to the original prediction, diverging from the pattern obscures the prediction. That is, fidelity corresponds to the negative prediction error or positive performance. Measure of supervised performance (difference) can estimate fidelity, where the label is replaced by the value the fidelity is measured against (for example, original prediction, or prediction label). Explanations carry different meanings depending on whether fidelity towards the prediction or the label is optimized. Optimizing towards label allows for the interpretation of the data and evaluation of data quality, and optimizing towards prediction allows for the interpretation of the model performance, e.g., analysis of error.

Maximizing sparsity and fidelity enforce concise and accurate representations of the data. These metrics are presented directly in the objective function. Contrastivity is ensured by objectives dependent on the output of every class (which push away from the incorrect classes). Necessity is ensured by adding to the objective function fidelity measurements from the graph components that are not in the explanation, i.e., of the non-covering subgraph. Sanity is checked through the comparison of models of different architectures and performances.

The explanatory graph components are the subgraph coverings of the monomorphisms of a graphlet, and the graphlet itself, or motif [13]. Subgraph covering is defined by the union of all the non-induced edge isomorphisms of the motif within the graph. A genetic algorithm is used to optimize the graphlets and corresponding subgraph coverings which graphlets can be considered functional motifs for explanatory power as described above and in [14].

Fig 1 illustrates the explanatory components in the methodology. For an input graph of some topology, Fig 1A, given a graphlet, Fig 1B, the subgraph monomorphisms are found of the graphlet in the graph. The union of the subraphs determine the coverings and non-coverings. For example, the union of the subgraph monomorphs of the graphlet in Fig 1B is calculated for the three graphs of Fig 1A. Two of the graphs have monomorphisms and thus a subgraph covering, shown in Fig 1C. Subgraph non-coverings are the edge difference of the graph and the covering (i.e., the edges of the graph that are not in any of the monomorphisms), shown in Fig 1D for the graph of Fig 1A and graphlet of Fig 1B. The graphlet with the largest explanatory power on itself and on its subgraph covering, and lowest explanatory power on the non-covering, is the motif explanation. The explanation has small prediction error on the subgraph covering and large on the non-covering subgraph. The size of the explanation is minimized by pushing the motif and covering as small as possible towards maximal sparsity.

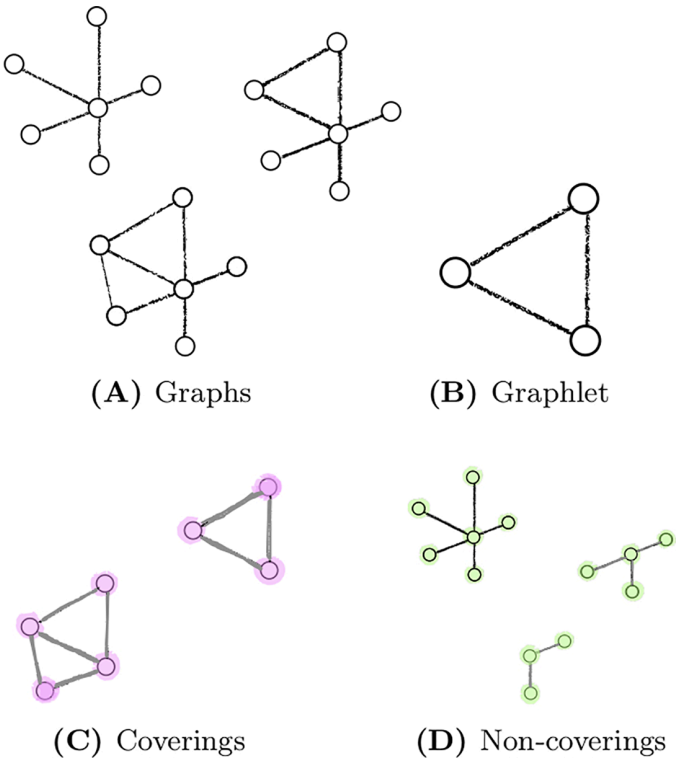

Fig 1. Partitioning graphs by a graphlet motif. The union of the subgraph monomorphs in given graphs constitute the subgraph covering for the graphlet. Edges not present in the coverings constitute the non-covering subgraphs.

https://doi.org/10.1371/journal.pcsy.0000067.g001

Subgraph size and prediction error are not independent variables; on larger subgraphs predictions tend to fall closer to predictions on the original graph, thus, it comes naturally to minimize explanation size. Otherwise, regions close to the original graphs are explored frequently in the subgraph space when searching for maximal fidelity. These regions are not necessarily dense with solutions of large explanatory power but more likely contain many substructures that are shared with the original graph. Similarly, less frequent motifs have larger explanatory power at equal fidelity and, therefore are more desirable under the same predictive power. Motif frequency correlates with the inverse size of the motif (see Supporting Information, S2 Fig). Smaller motifs can be more frequent as having relatively more space to occur in the graph, and more frequent motifs can cover more of the graph if they do not overlap largely. Then, with maximizing sparsity more frequent motifs are enforced, but more frequent motifs produce larger coverings. Hence the drift is balanced through minimizing the covering's size. Nevertheless, not every infrequent motif covers little of the graph. Every possible subgraph contains at least one motif that is infrequent, the subgraph itself (with a number of occurrences as much as the automorphisms of the subgraph). When the motif becomes the subgraph covering itself, it means the explanatory subgraph can not be broken down into smaller explanatory components, and the best explanation for the graph is a subgraph rather than a subgraph pattern.

Subgraph matching, and telling whether a graph contains a specific subgraph, are computationally difficult but not impossible problems. Finding all subgraphs of certain properties, as being beyond the decision problem, is intractable with deterministic algorithm [15]. Therefore, approximate subgraphs of certain properties are found by optimization via genetic

algorithms, as subgraphs span a non-differentiable space. Genetic algorithms are suitable to explore (sub-)graph structures [16,17]. Since subgraphs with multiple properties are of interest, multi-objective optimization is used with non-dominated sorting algorithm [18].

2. Relevant literature

Explainability in graph-based machine learning models rely mostly on depicting informative subgraphs, in a way similar to image segmentation in computer vision. Optimization and reinforcement learning, prototype based interpretability are some of the methods used to find such subgraphs. One of the challenges is in balancing computational feasibility with explanatory power, as subgraph discovery is complex itself. Here we focus on model-agnostic methods that treat the graph structure, features, or embeddings independently of the model specifics. Particular attention is given to methods employing genetic algorithms and subgraph covering strategies, which offer scalable means to explore large combinatorial spaces and uncover functionally relevant graphlets. So the focus is functional coverage and property-driven importance instead of frequency-based motif relevance.

Explainability in graphs through finding subgraphs is conceptually similar to explainability in computer vision through image segmentation. For example, Grau et al. [19] solved an optimization problem to directly account for sparseness in feature importance explanations. They incorporated regularization that maximizes performance decrease after marginalizing for irrelevant features, and used genetic algorithm with multi-objective optimization to generalize the optimization strategy to more than two-objectives.

There are several approaches to explaining a machine learning model on graph-structured data specifically. Some of them look at the graph, feature, or embedding, and use the machine learning model as a parameter (model-agnostic approaches), and that is the explainability the approach used in this work. In the following, relevant literature using similar explainibility methods will be discussed.

Ying et al. [20] and Luo et al. [21] employed graph generation to produce subgraph explanations. Taking a continuum limit of the non-differentiable graph domain, they construct an explainer parametrized by edge's importance weight. They apply regularization on the connectedness of pairs of edges, however, higher-order interactions of the edges are not considered.

Wang et al. [22] used a reinforcement learning agent that constructs explanatory subgraphs by sequentially adding salient edges. For accurately reflecting the original GNN reasoning process instead of offering post-hoc explanations in [23] the authors integrated prototype learning with GNNs for built-in interpretability. Predictions are explained by comparing inputs to learned prototypes in the latent space. Wu et al. [24] used a non-parametric framework to identify explanatory subgraphs by leveraging common motif patterns in graphs. It couples the target graph with similar instances and determines the most crucial joint substructure by minimizing the distance based on node correspondences. They also looked at the non-explanatory part of the graph (non-covering component in our case) and maximized the prediction difference in their pipeline [25]. They constrained the subgraph size with a fixed number. Yuan et al. [25] identified important subgraphs by Monte Carlo tree search and evaluated them using Shapley values.

It seems promising to incorporate subgraphs and subgraph patterns into machine learning models [26]. However, finding all subgraphs, even in one graph, is computationally difficult as being beyond a decision problem [15]. Motif finding and graph learning communities have to thus resort to constrained searches, for example, constrained by the graphlet sizes. Or they use algorithms that only provide the counts of the isomorphic substructures and do not give the

isomorphisms mappings themselves, and by that the subgraphs. This work does not constrain the size of the graphlets, yet it requires all monomorphisms of a given graphlet, thus it has a computational complexity at least as much as the non-decision graph isomorphism problem has. However, using genetic algorithm eases the further complexity arising from the quadratic growth with the graph structure. Due to the combinatorial nature of both the algorithm and data, the search space can be quickly explored through genetic algorithm [16,17,27].

Evolutionary algorithms are used to find graph structures with specific properties [16] and use objective functions with more than one objective [28]. He et al. [27] used a genetic algorithm combined with local search for this purpose. They find subgraphs that optimally cover an input graph by maximizing the total covering of the subgraph unions and minimizing the overlap between the covering subgraphs. Saquib et al. [17] uses genetic algorithm for explaining malware detection by subgraphs using single objective on a measure of fidelity. Here, the genetic algorithm is used to find the graphlet that has subgraph monomorphisms that are the most explaining in the graph by their subgraph covering, using multiple objectives on desirable properties of explanations.

Traditionally, motifs are defined by the number of occurrences compared to that in a random null model of the graph [7]. However, using a specific graph model for comparison skews the results towards the model, producing biased estimates of relevant motifs [29], hence further definitions of relevant motifs are sought for [30]. Especially when looking at functional relevance, not the most frequent substructures will be necessarily the most important. The high frequency of a subgraph alone does not indicate its importance or functional relevance, as it may sometimes result from trivial or redundant structures; structure does not imply function [31]. On the other hand, in biological networks, for example, certain frequent motifs correspond to specific functional units, like regulatory circuits or signaling pathways, and are important in explaining and interpreting biological processes [7].

When relevance is measured against a property or attribute, unlike relevance measured as a structural constituent, less frequent and smaller sub-structures can be as important. For example, a machine learning model may focus more on a small part of an input graph to predict a property, in which case a smaller pattern is responsible for determining the property, and a less frequent motif can be determined relevant for the pattern's smaller covering, or more frequent but smaller. The importance of a frequent motif can be due to the increased covering, and not due to the specific substructure being more important.

Homogeneous clusters of motifs and their connected collection have been found to be important in graph analysis, both structurally and functionally [32]. These findings suggest looking for a graphlet's covering (motif pattern), i.e., the union of the subgraph monomorphs of the graphlet in the graph, as the explanatory component in the graph, and the graphlet as a functional motif for a graph property.

2.1. Contribution

While these works study motifs and subgraph explanations separately, none of them uses subgraphs together with their motif pattern as explanatory components. Also this methodology is general, there are no restrictions on a graphlet's form or size, while none of the above works identifies arbitrary motifs. Furthermore, this study initiates a theoretical analysis of the objective functions, i.e., the properties of desirable graphlet explanations, in a rigorous framework through multi-objective optimization. Explanation properties are separate objectives for the optimization. The analysis is supported by experimental results of qualitative and quantitative analysis of the resulting explanations. The explanations highlight parts of the graph that can increase prediction accuracy greatly, and can find the known explanations

from synthetic data. Explanations are sensitive to label and the quality of the machine learning model, hence fulfill desirable properties of explanations. Limitations are discussed of current implementation, namely, the usage of single motif, unlabeled subgraph matches, as this work constitutes mainly a basis for the development of rigorous pipeline also theoretically.

Explanations provided by the methodology of Ying et at. [20] (GNNE), Luo et al. [21] (PGE), and Yuan et al. [25,33] (SubgraphX) are evaluated in comparison with the genetic algorithm based explanation methodology described here (motifGA). Results show SubraphX and motifGA have superior performance that aligns with the desired properties of explanations, providing necessary and sufficient explanations, and that motifGA outperforms SubgraphX on these small-scaled experiments. motifGA replicates and shows additional good explanations to those provided by state-of-the-art subgraph explanation algorithms, while is conceptually simple; using genetic algorithm with multi-objective optimization for the optimization of difficult functions. In addition, motifGA provides a break-down of the explanatory subgraphs, and uses this information during optimization, in the form of reoccurring graph patters (motifs), which is novel in relation to any of the subgraph explanation methodologies.

3. Materials and methods

Fig 2 depicts the overall methodology; given a graph G^i, and a motif M, the union of the monomorphic graphlets constructs a subgraph covering G^i_M, and non-covering $G^i_{\bar{M}}$. Predictions are made by a graph learning model on all three graphs yielding y_G, y_M, and $y_{\bar{M}}$. Prediction differences are calculated to determine fidelity (and see Sect 3.3).

3.1. Motif projection

A motif in a graph $G = (\mathcal{V}, \mathcal{E})$, $\mathcal{E} = (i,j) \mid i,j \in V$ is defined as the set of all the graphlets, $\mathcal{M} = \{g\}_G$, where the graphlet $g = (\{v\}, \{e\})$, $\{e\} = (i,j) \mid i,j \in \{v\}$, is monomorphic to the motif, i.e., for every $g \in \mathcal{M}$ there is a mapping $m : (i,j) \longleftrightarrow (m(i), m(j))$ for every $(i,j) \in \{e\}$ and $(m(u), m(v)) \in \mathcal{E}$, and is a subgraph of G, $g \subseteq G$. Monomorphisms [34,35] are chosen for the mappings because the edges are more detailed descriptors of the space of possible subgraphs as more non-induced subgraphs can be generated from a given set of nodes. Symmetric and

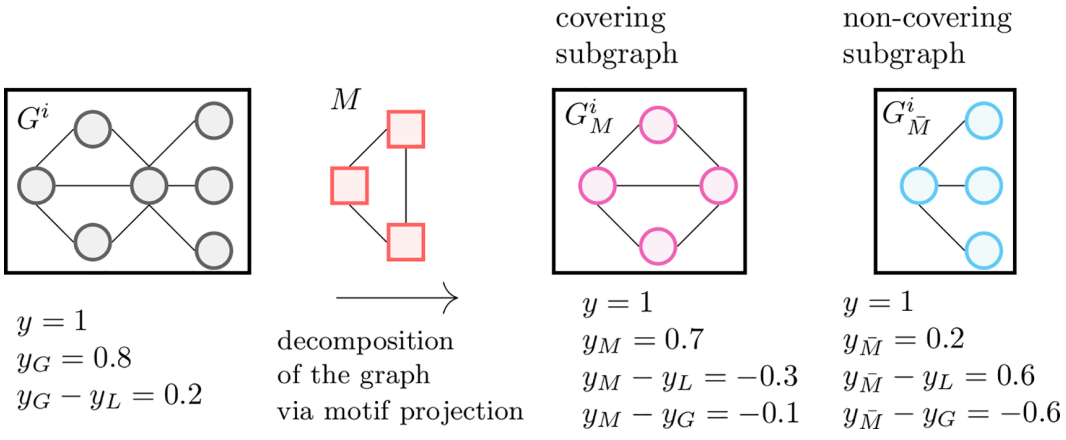

Fig 2. Evaluating motif projections. Prediction differences ($y_* - y_{L,G}$) to the original label (y_L) is calculated for subgraph covering and non-covering after motif projection to evaluate fidelity. Larger absolute deviation indicate smaller fidelity to the label (or prediction, y_G).

https://doi.org/10.1371/journal.pcsy.0000067.g002

connected graphs contain many monomorphisms of the matched pattern, then, the subgraph matching can be computationally intensive.

Then, the motif is projected on the graph by the union of the subgraph monomorphs, i.e., $C_M = \bigcup_k g_k \mid \forall g_i \subset G \wedge g_k \in \mathcal{M}$ and $C_M = (V_C, E_C)$ is the subgraph covering for the motif projected on the graph. Removing the edges from G that are in the subgraph covering produces the non-covering subgraph, $nC_M = (V_{nC}, E_{nC})$, where $E_{nC} = E \setminus E_C$ and $V_{nC} = (i,j) \mid i,j \in E_{nc}$.

3.2. Genetic algorithms

Eqs (1)–(4) describe the genetic algorithm. The algorithm E finds the explanation $\{m, c\}$ of the label or prediction through a graph learning model, A, on a graph instance G (see Eq (1)). The algorithm refines the solutions $\{m\}^t$ iteratively with increasing generations t. An explanation consists of the motif and subgraph covering, m and c. The graph learning model, the hyperparameters of the genetic algorithm h, the objective function f, and the graph instance G are given for the execution of the algorithm. While the algorithm iterates through the generations it changes the encoding of m (genes). m is encoded through a binary vector with the size of G in terms of edges. Positive values represent the presence of an edge, and the largest connected component is taken from the vector representation as m in the experiment with the different datasets. In the experiments with the state-of-the-art an improved version selects the most predictive component, i.e., the component with the largest fidelity.

$$E(\{m\}^t | h, f, G, A) = \{m, c\}^{t+1} \tag{1}$$

There are three basic operations in the algorithm that are performed at every generation (see Eq (2)). First, parents are selected from the population $\{m_*^t\} \subseteq \{m^t\}$ such that they are Pareto-optimal solutions of the objective function f, i.e. m_* are non-dominated solutions [18] of the maximization problem of the objective function $f(m, c, nc | G, A)$. After the selection of parents given the objective values, the crossover operation is applied that combines the genes of two of the solutions from the selected parents, denoted by $\times$ in Eq 2. Following crossover, the mutation operator is applied on each recombined solution, $\circ$ in Eq (2). The type of crossover and probability of mutation are hyperparameters of the genetic algorithm (see S3 Appendix for further details on the hyperparameters).

$$\{m\}^{t+1} = \circ\{m_*^t \times m_*^t\} \mid h, f(\{m_*^t\}) = \underset{m \in \{m\}^t}{\operatorname{pmax}} f(m, c, nc \mid G, A) \tag{2}$$

Subgraph covering (c) and non-covering (nc) are derived from the motif (Eq (3)) and are also used in the objective function.

$$\{m\}^{t+1} \rightarrow \{c\}^{t+1} \rightarrow \{nc\}^{t+1} \mid G \tag{3}$$

The initial edges of the motif (Eq (4)) are the largest connected component (CC) of the edges sampled from the original graph G, with a Bernoulli distribution and parameter p_I, the percentage of edges to remove initially.

$$\{m\}^0 = \{E_G \setminus U_E\} \mid U_E = \{u_E \sim \text{Bernoulli}(p_I)\}_{CC}^{|E_G|} \tag{4}$$

Motif and covering subgraph are edge subsets of the graph G and the graphlet is a subset of the covering subgraph by construction, $m \subseteq c \subseteq \{e\}_g$. It is computationally more feasible to find all subgraph isomorphs of a given graphlet found through the genetic algorithm (NP-complete), and construct the union, then to evolve the covering subgraph and decompose it to isomorph subgraphs of any form (NP-hard). Hence, the graphlet is the structure that is encoded in the genes within the algorithm as a binary vector representation of an edge subset of the graph instance.

The explanations optimized through the objective functions f directly to the evaluation metrics, which together with the hyperparameters of the genetic algorithm, h, determine the properties of the resulting explanations. Fidelity and sparsity are among the properties good explanations have and hence are optimized through the prediction accuracy and the size of the subgraphs. The PyGAD python package [36] is used for the implementation of the genetic algorithm with multiobjective optimization by the NSGA-II algorithm.

3.3. Objective functions

For a single objective, the best solution is found by maximizing the fitness value of the objective function $f \to \mathbb{R}$, or $\mathbb{N}$. In multi-objective optimization, $f \to \mathbb{R}^d$, $\mathbb{N}^d$, $d > 1$, the number of objectives can be more than one. Taking the objectives separately, there can be d different optimal solutions. Choosing a solution with the largest sum, or similar single-value summary of the fitness function, may not result in an optimal solution for any of the objectives individually. The NSGA-II algorithm [18] finds solutions on the Pareto front of the objective function that are considered equally optimal. Thus, the algorithm provides a set of optimal solutions, and even though the execution of the genetic algorithm, k solutions, possibly less than the number of optimal solutions, have to be selected ($k = 1$ from the final set of solutions for the selection of the explanation at evaluation). The selection of the k "best" optimal solution(s) can be made based on preferred objectives. For simplicity, here normalization and min-max scaling are applied on the selected objective and, after the scaling, the k largest of the average of the preferred objectives is selected. Here it is also examined which (combination) of the objectives provides the best qualitative and quantitative explanations.

Eqs (5a)–(5f) show the objectives of the fitness function used in this article. The function contains six objectives $f \to \mathbb{R}^6$, that are maximized through the algorithm. Fidelity, f, and sparsity, s, are directly used in the objective function (see Sect 3.4.2 for the elaboration of fidelity and sparsity measures). The sparsity of the motif, s_m, and covering subgraph s_c, is used in the objectives $O_s(m)$, and $O_s(c)$. $O_f(m)$ optimizes the fidelity of the graphlet motif f_m, $O_f(c)$ of the covering and, $O_f(nc)$ of the non-covering subgraph. e_m, e_c and e_{nc} are prediction errors used for the objectives, Eq (6) describes the regression error, Eq (7). Smaller motifs can fit more often into a given graph (see S2 Fig for an empirical evidence). Hence, the subgraph covering of smaller motifs is more likely to be bigger, and therefore can produce predictions closer to the prediction on the graph more easily. Less frequent motifs, with the same predictive power, are more desirable explanations hence. Furthermore, when minimizing the covering size, motif frequency is minimized indirectly. Nevertheless, for completeness, motif frequency is also included among the objectives in the form of $O_c(m)$.

$$f(m) = \begin{cases} O_f(m) = \frac{1}{e_m} \sim \frac{1}{(1-f_m)^2}, & (5a) \\ O_f(c) = \frac{1}{e_c} \sim \frac{1}{(1-f_c)^2}, & (5b) \\ O_f(nc) = e_{nc}, & (5c) \\ O_c(m) = \frac{1}{C_m}, & (5d) \\ O_s(m) = \frac{1}{1-s_m}, & (5e) \\ O_s(c) = \frac{1}{1-s_c} & (5f) \end{cases}$$

For regression, the individual prediction error $e_{x=\{m,c,nc\}}$ on a graph x is calculated as the squared prediction difference, $(p_x - p_L)^2$, on the graph to the label, relative to the original graph prediction difference $(p_G - p_L)^2$. The prediction output $p_{x=\{m,c,nc,G,L\}}$ on a graph x is given by the output of the graph learning model A, $p_{x=\{m,c,nc,G,L\}} = A(x) \in \mathbb{R}$ for regression, and $p^c_{x=\{m,c,nc,G,L\}} = \mathcal{A}(x) \in [0,1]^d$ for classification, with c categorical labels and d is the class dimension. For classification, the error for class c is the squared L2 norm $\|\cdot\|_2$ of predicted probability p^c_x to true label vector p^c. Such that predictions are not only pushed towards the true label but also pushed away from the incorrect label.

$$e_{x=\{m,c,nc\}} = \frac{(p_x - p_L)^2}{(p_G - p_L)^2} \quad (6)$$

$$e_{x=\{m,c,nc\}} = \|p^c_x - p^c\|_2^2 \quad (7)$$

There is a trade-off between the fidelity of an explanation (f_m, f_c, f_{nc}) and its size $(\frac{1}{s_m}, \frac{1}{s_c})$. Smaller explanations are more desirable at constant fidelity because they carry the same amount of information about the prediction but with less encoding of the information in a smaller space. The fitness function $f(m)$, Eq (5) is constructed such that the objectives account for the trade-off between size and fidelity. The reciprocal function $1/x$ together with the squared differences ensure that smaller prediction errors can get larger weight. Fidelity is optimized to the opposite direction on the non-explanatory component of the graph, i.e., the non-covering subgraph *nc*.

It is analyzed whether all objectives are equally important, and what it means for the explanation quality if the solutions are better in some of the objectives than in the others. For example, if covering fidelity O_{fc} is large, it might be difficult to find solutions with large motif fidelity O_{fm}, unless the motif is a **full motif**, i.e., it spans the entire covering, $O_{fc} = O_{fm}$. Then, the explanation is a *subgraph explanation* and there is no single explanatory pattern contained in it.

Fidelity can have two understanding. One interpretation is faithfulness of the explanation to the label, the other is is faithfullnes toward original prediction (explain phenomena and explain model [8]). Thus, when the prediction is $\mathcal{A}(X) = P_X$, the label $L_G \in G$, and the prediction on the graph $\mathcal{A}(G) = P_G$, f_m, f_c and f_{nc}, can be constructed either by $|L_G - P_X|$ or $|P_G - P_X|$, which result in explanations of different meanings. Optimizing $|L_G - P_X|$ reveals which parts of

the graph make the prediction better or worse when the model generally predicts well. Optimizing $|P_G - P_X|$ can reveal substructures that are responsible for the prediction of errors with inaccurate models.

3.3.1. Hyper-parameters and experiments with the Pareto front. The genetic algorithm results in a set of solutions, i.e., successful genes, that are in the Pareto-front found by selection through NSGA-II. Graphlets are encoded in the genes that represent the edges of the graph, and the graphlet's subgraph covering and non-covering is implied by projecting the graphlet on the graph (calculating the union the subgraph monomorphs). Given the graphlet, covering, and non-covering, the objective values corresponding to the solutions can be calculated by passing the subgraphs through the model and calculating their size. These solutions are equally optimal, given they are in the Pareto-front, however, one of them needs to be selected in order to evaluate the explanation method, which selected solution will be the explanation for the model and graph. Different solutions in the Pareto-front represent explanations with varying size and fidelity. To decide which solution to select as the explanation one can resort to domain knowledge by specifying the preferred objectives and select the solution that has the largest value on the preferred objectives (for example, one may prioritize fidelity of a solution rather than sparseness). Generally, weights can be assigned to the objectives and the Nash product can be calculated [37–39], and see Eq (8).

$$m^* = \max_{m \in \{m\}^T} \prod_i (f_i(m) - d_i)^{w_i} \tag{8}$$

The Nash product selects a balanced solution, m, from the Pareto-front $\{m\}^T$, of the non-convex objective function $f(m)$ with multiple objectives i given these objectives are conflicting, as it is the case in this optimization problem (see S1 Fig and S2 Fig). Every objective has a reference point d_i, optimally the worst value of f_i in the Pareto front, i.e., $\min_{f_i}\{f_i\}^T$, but can be approximated by the global minimum of $f_i(m)$ if it is better known. In the absence of weights all weights are set to be one. Preferred objectives can be selected with specifying a weight vector w, e.g., of zeros and ones, where w_i is one for a preferred objective and zero otherwise.

For tuning the hyper-parameters h of the genetic algorithm $\mathcal{E}$ the Optuna framework [40] is used with multiple objectives and the aim of maximizing the average output of each of the objectives of Eq (5) on the validation set. The hyper-parameters used for the optimization of the genetic algorithm are listed in S3 Appendix. The optimization method allows the exploration of the hyperparameter space that provides a set of solutions that can be considered equally optimal, yet having relative different fitness values in terms of different properties. I.e., equally optimal hyperparameter solutions generate explanations with different properties favored, quantified by the relatively different average O_{fm}, O_{fc}, O_{sc}, O_{fnc}, O_{sm} values. The parameter set h^* can be selected for the the genetic algorithm such that $E(f, h^*)$ results in explanations which favor some (combination) of the objectives. For example, solutions can be selected specifically for large fidelity with relatively large $\{O_{fm}, O_{fc}, O_{fnc}\}$ and for large sparsity relatively, with large $\{O_{sc}, O_{sm}\}$.

In addition to the selection of the hyperparameters that favor certain objectives, an operator was implemented that takes effect through the multi-objective optimization of the genetic algorithm, similarly, to favor specific objective(s). The operator ranks the Pareto-optimal solutions based on the fitness values of the specified objective(s) (takes effect at every generation, and at the selection of the final solution). Normalization and scaling of the fitness values were added as hyperparameters to ensure fitness values of different distributions can be comparable, too at the selection *from* the Pareto-optimal solutions.

He et al. [27] used node identities to fill the gene space of subgraph chromosomes, Saquib et al. [17] used the identities of the edges. That way the gene space is spanned by the node and edge identities, i.e., each node or edge in the graph is encoded by a unique phenotype. Then, the gene locations are permutation invariant and a gene can be present at any place in the chromosome to represent the presence of an edge in the solution. He et al. fixed the size of the subgraphs and bound gene locations to the subgraph's node. With an additional phenotype, which is used to represent the absence of an edge, however, chromosomes larger than the subgraph can be used to find variable-sized subgraph solutions. Nevertheless, several crossover operations assume fixed locations of genes. When phenotypes can permute freely across loci, and result in the same representation, the same subgraph can be represented twice in the chromosome, while some parts of the graph none. When using this kind of encoding, through mutation and recombination it is easier for the algorithm to explore smaller structures, because duplicate edge phenotypes map to the same edge. When a pair of solutions are recombined (both are a sampling of the same graph's edges with replacement), there likely will be overlapping among the edges given to the offspring, and the phenotypes representing the overlapping edges will occupy the genes. Furthermore, a percentage of the genes (edges) are mutated, possibly resulting in phenotypes that represent edges already present at other loci, introducing redundancy in the genes again. Overall, the encoding reduces the variation further among the genes (further than how it is already reduced through crossover). In this work, gene locations are fixed to the input graph's edges and the size of the chromosomes equals the number of edges in the graph. The gene space is binary and represents whether an edge is present in the subgraph solution or not.

3.4. Evaluation

Evaluation consists of two parts; qualitative evaluation through experiments on different datasets and models with known explanations, and quantitative evaluation with metrics used in machine learning and explainable AI. Quantitative metrics are used for comparison with state-of-the-art subgraph explanation methods.

3.4.1. Data and graph learning model. Two synthetic graph datasets are used for the prediction tasks which have tasks that are fairly well understood, such that the resulting explanations can be evaluated qualitatively, and contain graphs with different topological properties. Dataset I. [41] contains a regression task on random graphs which contain many symmetric small sub-structures. The task is to predict the proportion of 4-cycles to other substructures in the graph, and the SSWL subgraph neural network [42] is used for the prediction task. Dataset II. [21] contains Barabasi-Albert random graphs, with either a 5-house or a 5-cycle motif attached to one of the nodes. The task is a binary classification of the motif attached to the graph, and graph convolutional networks are used for the model [43].

Dataset II. is further used to evaluate the difference between explanation methods and machine learning models. Machine learning models are quite similar but differ in hyperparameters and training, and hence evaluate differently in terms of predictive power and expected to produce different explanations. For this purpose, two models are trained, one with 3 layers (model A) and one with 5 layers (model B), tuned separately for optimal hyperparameters. Details on the data, models, and training can be found under S1 Appendix.

Two synthetic datasets are selected, because they contain known explanations and therefore can be analysed theoretically [5,9]. Both datasets are important to evaluate explanatory and expressive power of graph learning models, and contain difficult tasks (that cannot be solved by 1-WL or vanilla graph convolutional networks). The two datasets is contrasted for this task because one is explained by specific structural element (dataset II. contains house

or a circle graphlet attachment) and the other is explained by the whole graph connectivity (dataset I. contains graphs with different proportion of 4-cycles). While dataset II. is expected to be well-explained by a subgraph explanation method, dataset I. is expected to work well for recognizing subgraph patterns.

3.4.2. Evaluation metrics. There are two ways, at least, to evaluate explanations, depending on whether a ground-truth explanation is present. Ground-truth explanations are often represented as node or edge masks indicating the presence of the node or edge in the true explanation. Groud-truth explanation is rare to have, especially for real-world datasets. Nevertheless, when it is present, it can be used to compare the ground-truth mask to the one provided by an input-level explanation. However, it is questionable even in the case of well-known synthetic datasets, if the model applies the same rules that determines the ground truth as it was prescribed by human construction. Taking the BA2motif dataset as an example, the presence of a house or a 5-cycle graphlet determines the class of a Barabasi-Albert graph by the construction of the dataset. However, these graphs have low clustering coefficient, and taking reasonably sized networks ($|V| = 25$), triangles or cycles will be very rare or not present. Then, the ground-truth explanations of this dataset does not have to align with the actual explanation, since the model can learn to detect only the triangles or the 4-cycles, both present in the house graphlet, in order to differentiate the classes. Such intricacies may be present in many of the human-designed explanations. Without knowing the entire decision-making process (the explanation), a well-aligned ground-truth explanation is difficult to construct. Another approach is to use evaluation metrics that are based on the desirable properties of explanations, such as fidelity and sparsity, that is the approach taken throughout this work, along with the quialitative evaluation of the explanations.

The genetic algorithm maximizes directly the metrics of evaluation that are based on the properties that define good explanations. While due to the direct maximization the resulting explanations expected to have large values in these metrics, sparsity and fidelity are opposing properties, and can counterbalance each-other, hence it is not trivial to find both of these values large at the same time for an explanation. For further details on the calculation of fidelity and sparsity metrics, see S2 Appendix. Contrastivity is maximized through optimizing a fidelity vector indexed by the output classes, i.e., a vector of the predictions is used to calculate the error, which is minimized through all classes. Since the machine learning model is a parameter of the explanation algorithm, the explanations should inherently posses the property of sanity, i.e., a model which performs differently on the machine learning task should have different explanations.

When optimizing for graphlet explanation $X_G = \{m, c\}$, the explanation constitutes both the graphlet $m = M_G$, and the covering subgraph $c = C_{M_G}$ of the graphlet motif. Then the graphlet explanation can be expressed as,

$$X_G = \begin{cases} M_G & = S^* \subseteq G \\ C_{M_G} & = \bigcup_k \hat{S}_k^*, \end{cases} \tag{9}$$

with the union of the graphlet isomorphs $\hat{S}_k$ in the graph (see Sect 3.1), and the quantitative evaluation metrics can be calculated for both components.

The datasets contain tasks that are well understood and therefore allow for the qualitative evaluation of the explanations (see Sect 3.4.1). Synthetic datasets contain graphs that have true labels generated by a known and human-intelligible function, which means X_G^{true} can be inferred from G. For example, one expects to find a "house" structure for one class (which

consists of a triangle and a 4-cycle), or a 5-cycle structure for the other class, as the explanation on the MotifBA dataset [21], which is known to contain classes generated by attaching these motifs to random graphs.

4. Results

Depending on the dataset and trained model, the algorithm can find subgraph explanations that are 1a) spars and represent the original prediction well, or 2a) predict much more accurately than the original graph. Table 1 and Fig 3 shows the results of the experiments. Experiments on Dataset I. was conducted with a well performing machine learning model (graph fidelity is 0.9988 on a scale which can reach at most 1. Here, covering fidelity was selected as an interesting objective, i.e., solutions large with this objective were selected from the Pareto-optimal solutions. Despite the good performance on the original graph, on 1436 of the 2500 instances a better prediction could be found on the explanation method (on average 0.9988 → 0.9998). Fig 3G shows an example of this on one of the instances, the prediction error is reduced on this subgraph covering greatly ($\Delta P = 0.023 \rightarrow 0.006$). On the other dataset (Dataset II.), 0.88 → 0.96 is the average performance increase with model A, and 0.74 → 0.87 with model B on the motifs, on which predictions are slightly better than on the covering subgraphs. However on this dataset with these models, predictions are better on average both on the motif and subgraph covering. Fig 4 shows examples of motifs found by optimization on Dataset II.

Some explanations, however, resulted in 1b) the complete overlap of covering subgraph and graph $C_{M_G} = G$, i.e. **full covering**, or 2b) complete overlap of the graphlet with covering $M_G = \bigcup_k \hat{S}_k^* = S^* = C_{M_G}$, **full motif**. The graphlet and its subgraph cover can take any form in the graph. While $\mathcal{M} \subseteq \mathcal{C} \subseteq \mathcal{G}$ holds, graphlet can cover the entire graph, and graphlet can be equal to the covering.

1b) and 2b) occur despite including objectives O_{sm} and O_{sc} that pushes solutions away from these regions. 1b) indicates that there is no better predictor constrained by the structure of the graph, than the graph itself, of the given prediction. Then, better prediction of the label

Table 1. Average target value and evaluation metrics of the optimized explanations with different optimization strategies and machine learning models.

condition	partition	graph fidelity	motif fidelity	covering fidelity	non-covering fidelity	motif sparsity	covering sparsity	*nr.o. inst.*
data: I.	all	0.9988	0.6984	0.7878	−1.1610	0.0110	0.1062	2500
obj.:	prediction better	0.9981	0.9353	0.9998	−0.7873	0.0069	0.0890	1436
covering fidelity	full motif	0.9989	0.8647	0.8647	−1.1358	0.0	0.0897	2258
model: regr.	full covering	0.9985	0.9059	0.9985	-	0.0037	0.0	209
data: II.	all	0.88	0.96	0.92	0.5	0.15	0.14	255
obj.: none	prediction better	0.79	0.94	0.88	0.5	0.17	0.15	144
model:	full motif	0.92	1.0	1.0	0.18	0.0	0.16	4
class., A	full covering	1.0	1.0	1.0	-	0.18	0.0	5
	class 0/1	0.85/0.92	0.95/0.97	0.92/0.93	0.29/0.74	0.13/0.18	0.15/0.12	133/122
data: II.	all	0.74	0.87	0.81	0.41	0.43	0.25	245
obj.: none	prediction better	0.7	0.88	0.8	0.43	0.44	0.27	192
model:	full motif	1.0	1.0	1.0	0.67	0.0	0.33	3
class., B	full covering	0.13	0.5	0.13	-	0.85	0.0	6
	class 0/1	0.62/0.89	0.79/0.96	0.67/0.97	0.18/0.68	0.42/0.44	0.23/0.28	132/113

Explanations are partitioned into groups where the motifs completely overlap with their covering subgraph and where not (full motif), where the covering subgraphs overlap with the graph and don't (full covering), where the model performs better on the explanation (prediction better), and, in case of classification, based on which class they belong to (class 0/1). Optimization condition refers to the selection preference from the Pareto-optimal solutions.

https://doi.org/10.1371/journal.pcsy.0000067.t001

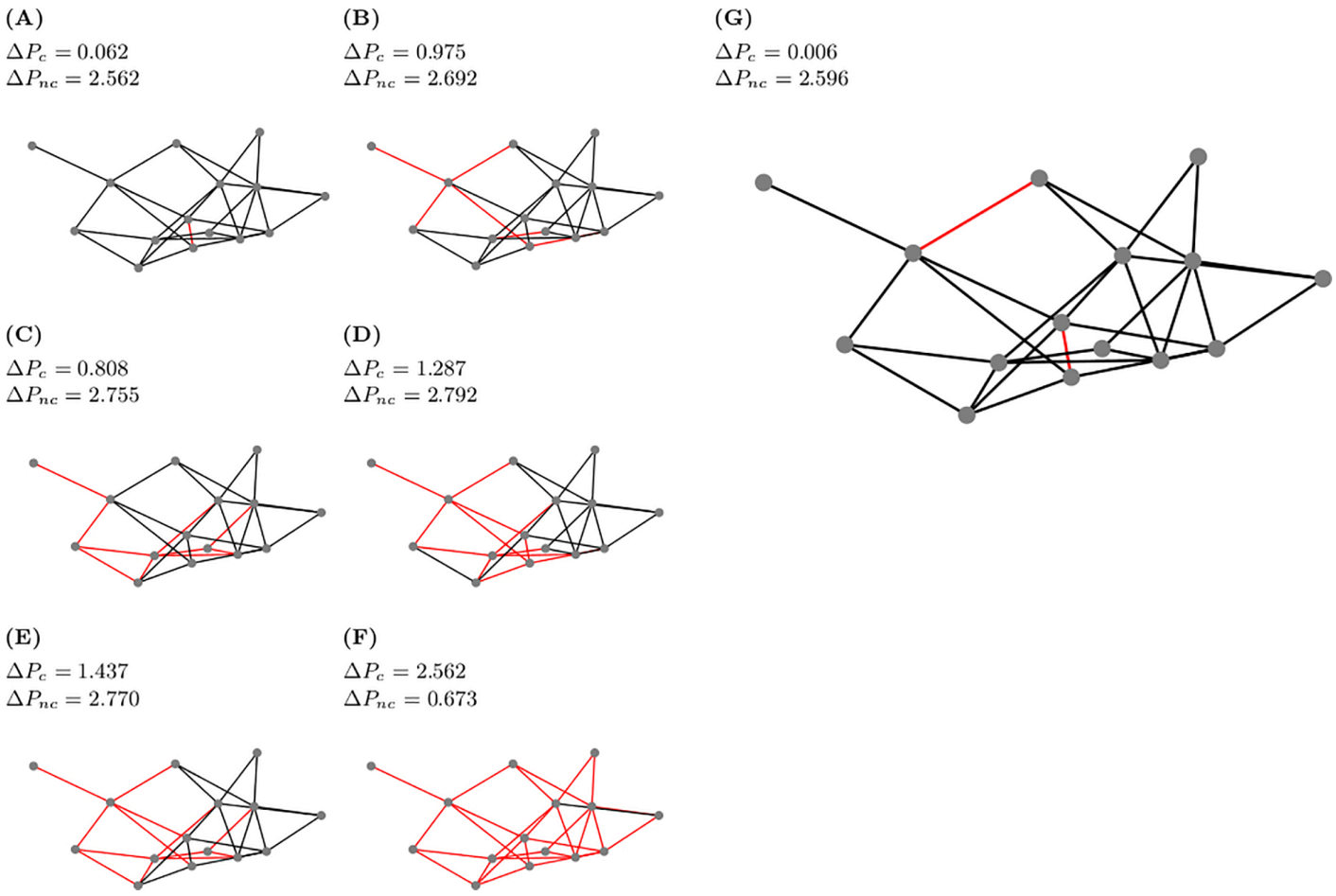

Fig 3. Covering solutions from running the genetic algorithm with different optimal hyperparameters. Subgraph coverings are shown by grey edges, non-covering components by red. The absolute prediction difference is shown above each figure, $\Delta P_x = |L - P_x|$. L, the true regression label of the graph is $L = 2.882$ (proportion of 4-cycles in the graph), and the prediction error on the original graph is $\Delta P_G = 0.023$. $P_{x \in \{c,nc\}}$ are the predictions on the covering and non-covering components. Each figure presents a solution of the genetic algorithm, with hyperparameters selected such that solutions are drifted in favor of some of the objectives (high fidelity, high sparsity, or both).

https://doi.org/10.1371/journal.pcsy.0000067.g003

cannot be achieved on any subset of the graph, perhaps because the entire graph is required for the accurate inference of the label, i.e., the entire graph constitutes the explanation. 2b) indicates that the explanatory subgraph cannot be broken down to a single smaller repeating explanatory substructure.

It can be concluded that the algorithm explores well the graphlet space. Even removing 90% of the edges from the graph as initialization of the graphlet, the algorithm finds instances of 1b) and 2b). Removing a large percentage of the edges in the initial population $\{S\}$ results in small graphlets to start the genetic algorithm with. Yet, the algorithm converges to complete coverings which require every edge to be present of G in $\bigcup_k \hat{S}_k^* = C_{M_G}$ in case of 1b), or S^* in C_{M_G} in case of 2b), which are distant solutions from small starting $\{S\}$-s. The explanations are capable of covering the entire domain, as in 1b) and 2b), even when the parameters are unfavorable for these to occur, furthermore, similar explanations are found with different initial populations and mutation operations. However, the methodology is sensitive to the model's performance, shown by Fig 4, and target, Table 1 class differences.

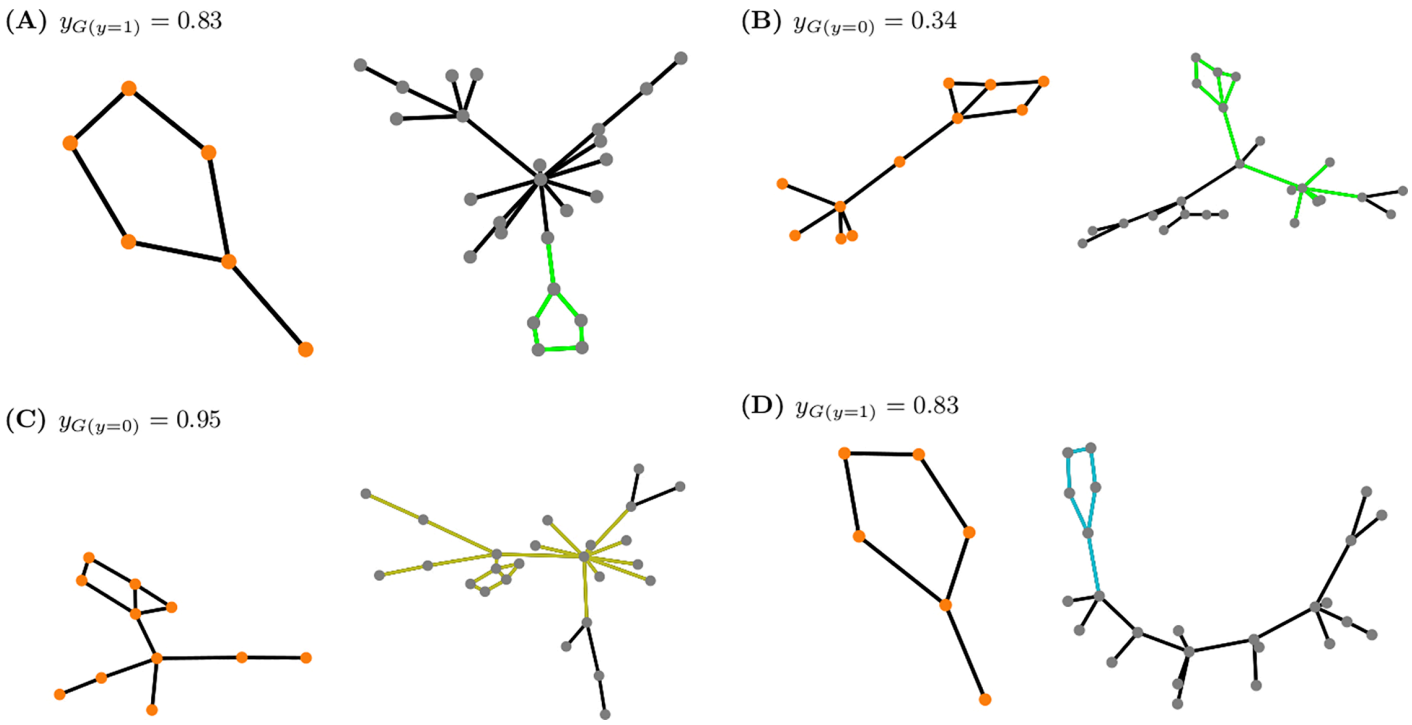

Fig 4. Explanations when optimized towards prediction.

https://doi.org/10.1371/journal.pcsy.0000067.g004

4.1. Subgraph covering

When the motif spans the entire covering, the monomorphs in the graph equal the automorphs of the graphlet, i.e., the motif can be found in the graph as many times as the motif can be found in itself. Then, the sparsity of the motif, $1 - |E_C|/|E_M|$ will be zero. In those cases, the best explanation is a subgraph explanation since no breakdown of the explanatory subgraph exists that explains the prediction better and possesses optimal properties.

Average motif sparsity values are shown by Table 1 "motif sparsity" column. Motif sparity is low for one of the datasets (Dataset I.), where also more "full motif" explanations can be found (2258 out of 2500). These results show, that in this dataset and with this specific model, the majority of the instances are better explained by a subgraph explanation, rather than a one-motif pattern. On Dataset II, however, with the exception of a few instances (4 out of 225 with model A, and 3 out of 245 with model B), combination of motif and subgraph can produce better explanation than the subgraph alone. Nevertheless, motif explanations carry additional values, they are the repeating pattern that breaks down the subgraph explanation.

Considering the subgraph covering of the motif, explanatory coverings can be compared to subgraph explanations from other methodologies. Table 2 and S1 Table show the evaluation results of GNNE, PGE, SubgraphX, and the proposed methodology motifGA. Results show motifGA outperforms the rest of the methods (with SubgraphX also scoring high), providing explanations that are more sufficient (scoring high on E) and necessary (scoring low on $G\backslash E$), see S1 Table. Fidelity Δ in Table 2 shows the difference of the fidelity scores on E and $G\backslash E$. motifGA produces the most explanation which predicts the label better than the original graph. On average, the explanations are about half the size of the original graph of motifGA, a

Table 2. Average evaluation metrics of explanation methods ($n = 57$).

fidelity Δ metric (↑)	random	GNNE	PGE	SubgraphX	motifGA
1 – MA_diff	0.007	0.041	−0.419	0.432	0.576
1 – MS_diff	0.006	0.040	−0.404	0.363	0.490
accuracy	0.000	0.000	−0.421	0.403	0.579
ROC-AUC	0.049	0.560	−0.413	0.5065	0.636
# prediction better	6	4	4	24	28
class 0 house (1 – MA_diff)	0.008	0.037	0.114	0.288	0.489
class 1 cycle (1 – MA_diff)	0.006	0.045	−0.867	0.592	0.671
sparsity (↑)	0.508	0.603	0.913	0.615	0.510

Dataset II. is used to benchmark explanation methods. Random explainer produces random explanation edge masks with thresholding at 0.5. The 0.5 threshold is also used to select edges from GNNE's and PGE's explanation edge masks. Details on the calculation of fidelity metrics can be found in S2 Appendix and under S1 Table.

https://doi.org/10.1371/journal.pcsy.0000067.t002

bit smaller of SubgraphX, and the smallest of PGE. Visual examples are provided under Supporting Information, S4 Fig–S8 Fig. GNNE and PGE are lower-order explanation methods, focusing on nodes and edges, while SubgraphX and motifGA incorporate interaction between sets of nodes and edges. Hence, it is not surprising that there is a great difference between these explanation methodologies.

4.2. Selecting solutions from the Pareto front

Through the hyperparameter optimization a number of "trials" are executed which test the algorithm with different hyperparameters on a few data points (see Sect 3.3.1, S3 Appendix–S3 Fig). This allows the analyses of a (biased) distribution of the algorithm outcomes. The hyperparameter optimization method (Optuna, see [40]) provides parameter importance based on the parameter values of the Pareto-optimal solutions. Parameter importance values were extracted for each objective separately (Eq (3.3)), and for the average of all, see S3 Fig for some examples. By the observation of several runs containing >30 trials, it can be concluded that similar parameters are more influential to some of the objectives. Specifically, objectives which correspond to fidelity properties, $\{O_{fm}, O_{fc}$, and $O_{fnc}\}$ in Eq (5), have similar feature importance profile, while structural objectives $\{O_{cm}, O_{sm}$, and $O_{fsc}\}$ feature importance profiles that resemble each other more. Further indicating the importance of hyperparameter selection in the properties of solutions resulting from the algorithm.

Fig 3B–3G represent increasingly sparser solutions (on the cost of reduced fidelity and increased error). Fig 3B shows the single edge that is the least important for the prediction. Via selecting hyperparameter solutions based on preferred objctives, different quality explanations can be found, such as the ones on the scale of Fig 3.

5. Discussion

Overall the explanations contribute to the understanding of the data, model and predictions. Graphs grow with $\mathcal{O}(n^2)$ when increasing the number of nodes n, while the number of possible subgraphs defined by the edges grows with $\mathcal{O}(\sum_{k=1}^{n} n^{2k}) = \lim_{x\to\infty} \mathcal{O}(n^n)$. Hence, finding all possible subgraphs is non-polynomial in the number of computations and infeasible with increasing graph sizes. Through optimization with the genetic algorithm, the input space can

be explored inexhaustibly without evaluating all possible subgraphs. Moreover, genetic algorithms can optimize non-differentiable functions and therefore are suitable for the subgraph domain.

5.1. Improving model performance on noisy data

In this paper, the predictions on the explainable components were optimized towards the true labels and target values of the graph p_L, rather than the prediction p_G. Optimizing towards the predictions were explored in the precursor of this work [14] and Fig 4. This way the explanations are of the labels, i.e. of some attribute of the graph, given the machine learning model, and allow for the evaluation of the data. For example, analyzing noisy components, and extracting informative components about the prediction task. Since the methodology shows components of graphs on what the prediction can be better, it can be useful when working with noisy data. When the model performs well, but not perfectly, it can show the subgraph that does have a better prediction (see Fig 3G).

5.2. Extracting components explaining model performance

Due to the trade-off between explanation size and fidelity, the size of the covering subgraph and graphlet is included in the objective function through O_{sc} and O_{sm}. Including additional terms may be more restricting on the solution space and make the multi-objective optimization more difficult to converge to the optimal solution rather than contributing to the explanatory power of the solution.

In graphs, traditionally, motifs are found and defined by specific criteria of the statistically significant abundance of subgraphs of the motif pattern, compared to a random model of the graph [7]. The criterion which determines the prevalence of the given pattern in the data, i.e., which determines whether the pattern is a motif, should provide an independent basis for determining what is a motif. However, the use of random model criteria affects which subgraph pattern is determined to be relevant [29], which criteria are heuristic since they are not from the data but determined arbitrarily up to some structural properties, e.g. degree distribution.

The presented explanations are all local in the space of the input data, meaning that every graph is explained individually. However, the common machine learning model connects the explanations, since it was trained on data representative over all the instances. Nevertheless, it is termed as local explanation [2], and there is value in looking at global explanations, i.e., at explanations that correspond to a set of the data points [21]. For the genetic algorithm that means to preserve the genes across instances, and to match similar edges, or nodes, across different graphs, or to match the prototype graphlet represented by the genes approximately [44], instead of finding exact monomorhpisms. Implementing the latter is useful anyway, considering graph data with many features. Another approach to extracting global explanations is the blending of individual explanations, for example, by fusing explanatory motifs [45].

In this work single explanations were analyzed, i.e., only one motif, or graphlet pattern, were looked at once. That is a limiting factor, a low-resolution description of the graph. Multi-motif explanations are required for further detail of description and would be useful in real-world application, e.g., looking at properties of molecules Fig 5, where more than one motif may be relevant for a property. The choice of relevant/interesting motifs from a dictionary of motifs is itself interesting to explore and relevancy of motifs could be related to the problem at hand or the application domain. That means constructing coverings from multiple motifs, and inputting them to the genetic algorithm together. There are some aspects which have to

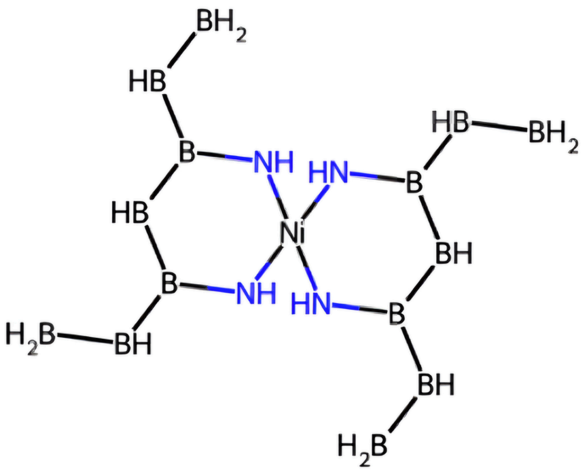

Fig 5. Graph of a molecule.

https://doi.org/10.1371/journal.pcsy.0000067.g005

be considered additionally when such, e.g., whether overlap in the constituting graphlets is allowed. Hence, multi-motif optimization were considered as part of the future work.

6. Concluding remarks

This work lays down the cornerstones of explanatory graphlet motif analysis for explainable graph learning. The article demonstrates the potential of using genetic algorithms for optimizing substructures for explainability. The optimization objectives presented here as determinants of relevant substructures for machine learning prediction, graphlet, and subgraph covering, are justified in theory and supported with empirical experiments on two datasets that contain different graphs and prediction tasks. The resulting "explanations" are informative over the predictions, model and dataset. The genetic algorithm explores well the subgraph space, and provides state-of-the-art quality explanations, hence is suitable for the optimization of this discrete non-differentiable domain.

The resulting explanations fulfill multiple desirable properties of good explanations, which are optimized through multi-objective optimization using genetic algorithm. The method provides subgraph explanations and the motif that breaks down the subgraphs, hence makes the subgraph explanation more intelligible. The method provides higher-order subgraph explanations and hence outperforms egde- and node-based explanations with a large margin, and outperforms a state-of-the-art subgraph explanation method, while providing (and optimizing) the constituting motif building-blocks of the subgraph as well.

This work provides the basis for future work involving multiple graphlets as building-blocks of the explanations. Hence, here two synthetic datasets were used that are important for expressive and explainable graph learning, which justifies the usability of the methodology and provides a good conceptual baseline for future work. Using multiple graphlets allows for a more detailed breakdown of a subgraph explanation, aiming for generality. Future work involves the usage of more complex, real-world datasets that are explained with multiple motifs for a more complete symbolic description of subgraph explanations.

Supporting information

S1 File. Supporting information file.
(PDF)

S1 Appendix. Data and model. The two datasets (Cycle-4 [42,46] and BA2 [21]) both contain graphs and the graph's property for prediction. Both datasets are important to evaluate the expressive power of GNNs as they contain tasks that require recognition of cycles and simple GNNs have difficulties differentiating between cycle- and tree-like structures. Models (SSWL+ [42] and GCN [43]) used for a dataset are listed below the dataset. Both types of models use (different) techniques to enhance the expressive power of GNNs and can thus effectively solve the regression and classification tasks.
(PDF)

S2 Appendix. Calculation of fidelity and sparsity. Details on the calculation of the evaluation metrics.
(PDF)

S3 Appendix. Hyperparameters of the genetic algorithm and tuning. Hyperparameters are tuned with multi-objective optimization using Optuna [40]. The genetic algorithm, Eq (1), have these parameters tuned.
(PDF)

S1 Fig. Trade-off between sparsity and fidelity. Fitness value of the objective optimizing covering sparsity (cov_size) plotted against the fitness value which optimizes fidelity (cov_pred), measured during different runs of the hyperparameter tuning [40] (using different sets of parameters). Fig (A–C) corresponds to Dataset II, Fig (D–F) is of Dataset I. Size is minimized, decreasing with increasing the y-axis (fintess values are maximized).
(PDF)

S2 Fig. Motif size and number of occurrences. Fitness value of the objective optimizing motif count plotted against the fitness value optimizing motif size, measured during different runs of hyperparameter tuning [40] (using different sets of parameters). Fig (A–F) corresponds to Dataset II, Fig (G–L) is of Dataset I. Motif count and size are minimized, decreasing with increasing the x-, and y-axes (fitness values are maximized). The tendency, that motif size is inversely proportional to motif count is visible, especially on the Cycle-4 dataset.
(PDF)

S3 Fig. Feature importance of hyperparameters per objective and hypervolume plot. Plots show feature importances (A–G), and the hypervolume plot (H) provided by Optuna [40]. (A), (C), and (E), are hyperparameter importances of objectives depending on functional properties of the subgraphs (motif prediction, covering prediction, and non-covering prediction). (B), (D), and (F), are importances of objectives depending on structural properties (motif size, and covering size, and motif count). (G) shows the average importances of all properties. For different objectives, different hyperparameters are important, parameter's importance vary with the type of objective (functional or structural).
(PDF)

S1 Table. Average evaluation metrics of explanations coming from different subgraph explanation methodologies ($n = 57$). Dataset II. is used to benchmark explanation methods. On all fidelity metrics = one minus mean absolute (MA), squared (MS), difference, accuracy, and area under the ROC-curve, motifGA outperforms the rest of the explanation methods, even when looking at the performance on the different classes (house or cycle) separately.

Δ corresponds to the difference between the fidelity metric on the explanation E, ($fid_E \sim fid_-$), and the fidelity metric on the graph after the explanation removed $G\backslash E$, ($fid_{G\backslash E} \sim fid_+$), i.e., $fid_\Delta = fid_E - fid_{G\backslash E}$, the larger fid_Δ, the better the explanation in terms of fidelity. In terms of sparsity, PGE provides the best explanations on the expense of fidelity. However, the relative importance of fidelity and sparsity can be adjusted with a weight parameter of motifGA.
(PDF)

S4 Fig. Subgraph explanations from GNNExplainer. Explanations for the first eighteen graphs from the BA2 dataset.
(PDF)

S5 Fig. Subgraph explanations from PGExplainer. Explanations for the first eighteen graphs from the BA2 dataset.
(PDF)

S6 Fig. Subgraph explanations from SubgraphX. Explanations for the first eighteen graphs from the BA2 dataset.
(PDF)

S7 Fig. Subgraph explanations from motifGA. Explanations for the first eighteen graphs from the BA2 dataset by the explainer presented in this work.
(PDF)

S8 Fig. Subgraph explanations from Dummy Explainer. Explanations for the first eighteen graphs from the BA2 dataset by a random explainer.
(PDF)

Author contributions

Conceptualization: Bettina Soós, Gonzalo Nápoles, Çiçek Güven.

Formal analysis: Bettina Soós.

Investigation: Bettina Soós, Gonzalo Nápoles, Pieter Spronck, Çiçek Güven.

Methodology: Bettina Soós, Gonzalo Nápoles, Çiçek Güven.

Project administration: Pieter Spronck.

Resources: Gonzalo Nápoles, Pieter Spronck, Çiçek Güven.

Software: Bettina Soós.

Supervision: Gonzalo Nápoles, Pieter Spronck, Çiçek Güven.

Visualization: Bettina Soós.

Writing – original draft: Bettina Soós.

Writing – review & editing: Gonzalo Nápoles, Çiçek Güven.

References

1. Liang Y, Li S, Yan C, Li M, Jiang C. Explaining the black-box model: a survey of local interpretation methods for deep neural networks. Neurocomputing. 2021;419:168–82.
2. Saleem R, Yuan B, Kurugollu F, Anjum A, Liu L. Explaining deep neural networks: A survey on the global interpretation methods. Neurocomputing. 2022;513:165–80.
3. Watson DS, Gultchin L, Taly A, Floridi L. Local explanations via necessity and sufficiency: unifying theory and practice. In: Proceedings of Machine Learning Research. 2021. p. 1382–92.

4. Xia F, Sun K, Yu S, Aziz A, Wan L, Pan S. Graph learning: a survey. IEEE Transactions on Artificial Intelligence. 2021;2(2):109–27.
5. Yuan H, Yu H, Gui S, Ji S. Explainability in graph neural networks: a taxonomic survey. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2022;45(5):5782–99.
6. Nandan M, Mitra S, De D. GraphXAI: a survey of graph neural networks (GNNs) for explainable AI (XAI). Neural Comput Appl. 2025;:1–52.
7. Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U. Network motifs: simple building blocks of complex networks. Science. 2002;298(5594):824–7. https://doi.org/10.1126/science.298.5594.824 PMID: 12399590
8. Amara K, Ying R, Zhang Z, Han Z, Shan Y, Brandes U. Graphframex: Towards systematic evaluation of explainability methods for graph neural networks. PMLR LoG. 2022;:44–1.
9. Agarwal C, Queen O, Lakkaraju H, Zitnik M. Evaluating explainability for graph neural networks. Sci Data. 2023;10(1):144. https://doi.org/10.1038/s41597-023-01974-x PMID: 36934095
10. Pope PE, Kolouri S, Rostami M, Martin CE, Hoffmann H. Explainability methods for graph convolutional neural networks. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2019. p. 10772–81.
11. Yeh CK, Hsieh CY, Suggala A, Inouye DI, Ravikumar PK. On the (in) fidelity and sensitivity of explanations. Adv Neural Inf Process Syst. 2019;32.
12. Zheng X, Shirani F, Wang T, Cheng W, Chen Z, Chen H. Towards robust fidelity for evaluating explainability of graph neural networks. arXiv preprint 2023. https://arxiv.org/abs/2310.01820
13. Espejo R, Mestre G, Postigo F, Lumbreras S, Ramos A, Huang T, et al. Exploiting graphlet decomposition to explain the structure of complex networks: the GHuST framework. Sci Rep. 2020;10(1):12884. https://doi.org/10.1038/s41598-020-69795-1 PMID: 32732972
14. Soós B, Güven Ç, Nápoles G, Spronck P. Determining motif explanations for learning on graphs. In: Complex Netw Appl XIII. 2024. p. 251–4.
15. Garey MR, Johnson DS. Computers and intractability: a guide to the theory of NP-completeness. W H Freeman; 1990.
16. Jensen JH. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. Chem Sci. 2019;10(12):3567–72. https://doi.org/10.1039/c8sc05372c PMID: 30996948
17. Saqib M, Fung BC, Charland P, Walenstein A. GAGE: Genetic algorithm-based graph explainer for malware analysis. In: ICDE. 2024. p. 2258–70.
18. Deb K, Pratap A, Agarwal S, Meyarivan TAMT. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation. 2002;6(2):182–97.
19. Grau I, Nápoles G. Sparseness-Optimized Feature Importance. In: xAI. 2024. p. 393–415.
20. Ying Z, Bourgeois D, You J, Zitnik M, Leskovec J. Gnnexplainer: Generating explanations for graph neural networks. Adv Neural Inf Process Syst. 2019;32.
21. Luo D, Cheng W, Xu D, Yu W, Zong B, Chen H. Parameterized explainer for graph neural network. Adv Neural Inf Process Syst. 2020;33:19620–31.
22. Wang X, Wu Y, Zhang A, Feng F, He X, Chua TS. Reinforced causal explainer for graph neural networks. IEEE PAMI. 2022;45(2):2297–309.
23. Zhang Z, Liu Q, Wang H, Lu C, Lee C. Protgnn: Towards self-explaining graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2022. p. 9127–35.
24. Wu F, Li S, Jin X, Jiang Y, Radev D, Niu Z. Rethinking explaining graph neural networks via non-parametric subgraph matching. In: PMLR ICML. 2023. p. 37511–23.
25. Yuan H, Yu H, Wang J, Li K, Ji S. On explainability of graph neural networks via subgraph explorations. In: Proceedings of Machine Learning Research. 2021. p. 12241–52.
26. Zhao JL, Zhang XJ, Ding X, Zhang X, Zhang HF. Enhancing graph structure learning via motif-driven hypergraph construction. IEEE Transactions on Network and Service Management. 2025.
27. He X, Wang Y, Du H, Feldman MW. A memetic algorithm for finding multiple subgraphs that optimally cover an input network. PLoS One. 2023;18(1):e0280506. https://doi.org/10.1371/journal.pone.0280506 PMID: 36662749
28. Nicolaou CA, Brown N. Multi-objective optimization methods in drug design. Drug Discov Today Technol. 2013;10(3):e427-35. https://doi.org/10.1016/j.ddtec.2013.02.001 PMID: 24050140
29. Artzy-Randrup Y, Fleishman SJ, Ben-Tal N, Stone L. Comment on “Network motifs: simple building blocks of complex networks” and “Superfamilies of evolved and designed networks”. Science. 2004;305(5687):1107; author reply 1107. https://doi.org/10.1126/science.1099334 PMID: 15326338

30. Bénichou A, Masson J-B, Vestergaard CL. Compression-based inference of network motif sets. PLoS Comput Biol. 2024;20(10):e1012460. https://doi.org/10.1371/journal.pcbi.1012460 PMID: 39388477
31. Ingram PJ, Stumpf MPH, Stark J. Network motifs: structure does not determine function. BMC Genomics. 2006;7:108. https://doi.org/10.1186/1471-2164-7-108 PMID: 16677373
32. Dobrin R, Beg QK, Barabási A-L, Oltvai ZN. Aggregation of topological motifs in the Escherichia coli transcriptional regulatory network. BMC Bioinformatics. 2004;5:10. https://doi.org/10.1186/1471-2105-5-10 PMID: 15018656
33. Mahlau Y, Berg L, Kayser L. [Re] On Explainability of Graph Neural Networks via Subgraph Explorations. MLRC2022. 2023.
34. Jüttner A, Madarasi P. VF2 —An improved subgraph isomorphism algorithm. Discrete Appl Math. 2018;242:69–81.
35. Matelsky JK, Reilly EP, Johnson EC, Stiso J, Bassett DS, Wester BA, et al. DotMotif: an open-source tool for connectome subgraph isomorphism search and graph queries. Sci Rep. 2021;11(1):13045. https://doi.org/10.1038/s41598-021-91025-5 PMID: 34158519
36. Gad AF. Pygad: An intuitive genetic algorithm python library. Multimed Tools Appl. 2004;83(20):58029–42.
37. Quadt T, Lindelauf R, Voskuijl M, Monsuur H, Čule B. Dealing with multiple optimization objectives for UAV path planning in hostile environments: a literature review. Drones. 2024;8(12).
38. Charkhgard H, Keshanian K, Esmaeilbeigi R, Charkhgard P. The magic of Nash social welfare in optimization: do not sum, just multiply!. ANZIAM J. 2022;64:119–34.
39. Nash J. Two-person cooperative games. Econometrica. 1953;21(1):128. https://doi.org/10.2307/1906951
40. Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: a next-generation hyperparameter optimization framework. ACM SIGKDD. 2019;25:2623–31.
41. Zhao L, Jin W, Akoglu L, Shah N. From stars to subgraphs: uplifting any GNN with local structure awareness. In: ICLR. 2022.
42. Zhang B, Feng G, Du Y, He D, Wang L. A complete expressiveness hierarchy for subgraph g n n s via subgraph weisfeiler-lehman tests. PMLR ICML. 2023;:41019–77.
43. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. ICLR. 2017.
44. Yan J, Yin XC, Lin W, Deng C, Zha H, Yang X. ACM ICMR. 2016;:167–74.
45. Nguyen HL, Vu DT, Jung JJ. Knowledge graph fusion for smart systems: a survey. Inf Fusion. 2020;61:56–70.
46. Chen Z, Chen L, Villar S, Bruna J. Can graph neural networks count substructures?. Adv Neural Inf Process Syst. 2020;33:10383–95.