Bridging Equational Properties and Patterns on Graphs: an AI-Based Approach

Oguzhan Keskin [*] [1] **Alisia Lupidi** [*] [1] **Francesco Giannini** [2]
Stefano Fioravanti [2] **Lucie Charlotte Magister** [1] **Pietro Barbiero** [1] **Pietro Liò** [1]

Abstract

AI-assisted solutions have recently proven successful when applied to Mathematics and have opened new possibilities for exploring unsolved problems that have eluded traditional approaches for years or even centuries. Following this direction, this paper presents an innovative approach aiming at establishing correlations between equational properties of algebraic structures that can be represented through graphs and specific subportions of their topological representation. The methodology incorporates the utilization of graph neural architectures to validate theorems or conjectures, complemented by Explainability (XAI) metrics that lend support to these statements. In particular, we examine the distributive and modular properties of algebraic lattices, whose characterization is well-known in universal algebra, hence using these properties as an experimental test bench. The findings of this study demonstrate the effectiveness of the proposed approach in identifying and retrieving established subpatterns that characterize the equational properties under investigation. Moreover, the approach exhibits the capability to generate novel and noteworthy candidates as theorem suggesters, thereby offering valuable prospects for further exploration by mathematicians.

1. Introduction

From the early '60s, several fields of Mathematics have benefited enormously from technology-based approaches. While for now computational techniques and data-generation tasks have been the most prominent applications of Computer Science to Mathematics, recent advancements in AI technologies have opened up new avenues for researchers to tackle complex mathematical intractable problems with classical approaches. Notably, these AI techniques have facilitated the resolution of previously unsolved problems (Lample & Charton, 2019) and the formulation of fresh conjectures (Davies et al., 2021). This research line represents an untapped frontier within the broader scope of AI, holding great potential for profound impacts across various mathematical fields.

Universal Algebra (UA) is recognized as a fundamental discipline in contemporary mathematics, serving as a cornerstone of mathematical studies. Despite its significance, the intricate nature of abstract algebraic structures has presented challenges, impeding scientific advancement and dissuading many scholars from delving into this field. UA investigates algebraic structures from an abstract standpoint, and it is intriguing to note that several theorems in this field equivalently characterize algebraic properties using equations or graphs (Jipsen & Rose, 1992). Exploring universal algebra properties through graph representations could leverage powerful AI architecture such as Graph Neural Networks (GNN, (Scarselli et al., 2008)), excelling in analyzing graph-structured data. However, the limited transparency and brittle nature of GNN explainability impedes the human comprehension of their decision-making processes (Rudin, 2019), hindering mathematicians from employing these techniques to empirically validate existing conjectures or formulate new ones.

On the other hand, our approach uses a graph-explaining model to identify relevant sub-portions of the input graphs responsible for a certain prediction. In UA this explanation may correspond to a graph-pattern in an algebraic structure, whose omission may characterize an algebraic variety having a certain equational property. In the particular context of our work, isolating the subgraphs that have a direct correlation to the equational properties is pivotal in simplifying the characterization of large and/or infinite algebras - which, at that point, can be done just by looking at its structure.

[*]Equal contribution [1]University of Cambridge, Cambridge, United Kingdom [2]University of Siena, Siena, Italy. Correspondence to: Alisia Lupidi <aml201@cam.ac.uk>, Oguzhan Keskin <ok313@cam.ac.uk>.

Proceedings of the 2^{nd} Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML) at the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. 2023. Copyright 2023 by the author(s).

The main contributions of this paper are as follows:

- We propose the first AI-based approach to identify graph-patterns responsible for well-known equational properties in universal algebra.
- We present an explainable methodology for point out theorem suggestions on graph-structured data.
- We release three hand-crafted datasets of lattices labelled as satisfying/violating the distributive and modular properties. The experiments show the effectiveness of our approach on these datasets identifying known graph-patterns in UA characterizing these properties. As an extra step, we explore the quantitative metrics of GCExplainer, namely completeness and purity scores.

The paper is organized as follows. In Section 2 we discuss related studies exploring the use of AI techniques for mathematical problems and explainable AI graph neural architectures. Section 3 provides an introduction to the fundamental concepts of universal algebra, which are essential for the comprehension of the problem statement presented in this paper, and the graph neural architectures that we combine in Section 4 to define the pipeline of our approach and the experimental setup. Finally, Section 5 discusses the main findings of this paper, and Section 6 draws some conclusion and future research directions.

2. Related Work

2.1. AI for Mathematics

Several studies have been proposed with the aim of harnessing AI-based techniques to address specific mathematical questions or tasks: like the Erdős discrepancy problem, solved in 2015 after 60 years thanks to deep learning (Eberhard, 2016). Lample and Charton (Lample & Charton, 2019) demonstrated the power of neural networks at symbolic integration and solving differential equations, while Peifer et al. (Peifer et al., 2020) used reinforcement learning to speed up the calculations of polynomial equations. Wagner (Wagner, 2021) leveraged deep reinforcement learning to find constructions and counter-examples to several open conjectures in combinatorics and graph theory. A long traditional line of research focused on the study of automatic logic reasoning and theorem proving (Loveland, 1986; Fitting, 2012; Loveland, 2016) has more recently strongly deflected toward the use of embedding representations and graph neural networks (Wang et al., 2017; Bansal et al., 2019; Paliwal et al., 2020). Davies et al. (Davies et al., 2021) have proposed a novel machine learning framework to discover patterns and relations between mathematical objects. Moreover, AI can be applied to gain novel insights on innovative conjectures that can drive research advancements in several fields of mathematics. More similar to our approach, He (He, 2022) has further provided a comparative study about how AI is used for detecting the existence of mathematical structures in fields of mathematics spanning from geometry to representation theory. To the authors knowledge, our work is one of the first attempts to apply AI models in open research directions of universal algebra. Moreover, our approach differs from the mentioned methods, as we rely on the combination of common GNNs architectures for graph classification with the addition of a module highlighting the relevant sub-portion of its input as a graphical explanation for the prediction.

2.2. Graph Explanation

Explainable Artificial Intelligence (XAI) is the branch of AI concerned with developing methods that justify, in a human-understandable fashion, decisions made by a model. Most AI algorithms are considered as a black-box - i.e. there is no knowledge of the internal workings of the system, which is not desirable under a risk-prevention and interpretability point of view (Arrieta et al., 2019). Graph Neural Networks (GNNs), as deep learning (DL) methods, possess complex prediction processes that are not easily interpretable (van der Velden et al., 2022). The challenges lie not only in the DL nature of GNNs, but also in the representation of data itself. Graphs are less intuitive compared to images and texts, and the topology information on which GNNs are based requires specialized features for which traditional explainability methods are inadequate. There are two main approaches to achieve explainability in GNNs (Holzinger et al., 2022): instance-level methods and model-level methods. Instance-level methods aim to establish connections between the model's behavior and important input features for prediction. On the other hand, model-level methods provide broader insights and a higher-level understanding of the structure (graph type) of the model itself, rather than focusing solely on the data. Among these approaches, GNNExplainer (Ying et al., 2019), a perturbation-based explainer, stands out as it can provide explanations for predictions made by any GNN-based model across various Graph Machine Learning (GML) tasks. However, such explanations are specific to individual instances and can be challenging to reason about in the context of the entire class. To address this limitation, we consider a concept-based explanation method, Graph Concept Explainer (GCExplainer) (Magister et al., 2021), aiming at offering explanations in the form of concepts, which are small, higher-level units of information readily accessible to humans.

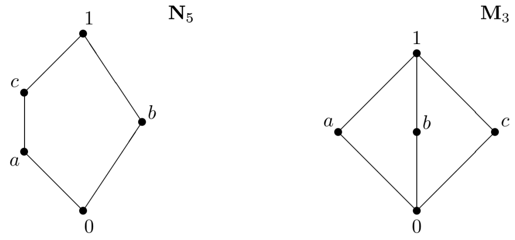


Figure 1: Examples of Hasse diagrams of the lattices $\mathbf{N}_5$ and $\mathbf{M}_3$. $\mathbf{N}_5$ is an example of a non-modular non-distributive lattice, while $\mathbf{M}_3$ is a modular non-distributive lattice.

3. Background

3.1. Universal Algebra

Universal Algebra (UA) considers the investigation of algebras, which can be defined as mathematical structures composed of a set of elements along with operations defined on those elements. The operations may include binary operations (such as addition or multiplication), unary operations (such as negation or inversion), nullary operations (like constants), and so forth.

Definition 1. An *algebra* $\mathbf{A}$ is a pair (A, F) where A is a non-empty set called *universe* and F is a set of finitary operations on A.

Apart from the operations on A, an algebra is further defined by axioms, that in the particular case of universal algebras are in the form of identities. The objective of universal algebra is to identify and explore the common algebraic properties that are shared among diverse mathematical systems, often expressed as sets of equations. In particular, *varieties* are classes of algebras which are models of equational theories. By explicitly defining these equations, varieties facilitate the classification and analysis of a wide class of algebras according to their shared properties. For a comprehensive understanding of foundational concepts in UA, we refer the reader to Burris & Sankappanavar (1981a).

3.1.1. Lattices

A noteworthy variety of algebras are Lattices, which holds significant relevance due to their association with logical structures and their remarkable algebraic properties.

Definition 2. A partially ordered set L, i.e. a set L equipped with a reflixive, antisymmetric, and transitive relation $\leq_L$, forms a *lattice* if and only if for every $a, b \in L$ both *supremum* and *infimum* of $\{a, b\}$ exist in L with $a \vee b$ being the supremum and $a \wedge b$ the infimum.

As well known in the literature (Birkhoff, 1940), a lattice $\mathbf{L}$ can be equivalently defined as an algebraic structure composed by a non-empty set L and two binary operations $\vee$ and $\wedge$ satisfying the axioms of *commutativity*, *associativity*, *idempotency*, and *absorption* (see Appendix 2). Graphical representations of lattices can be achieved using *Hasse diagrams*, which are undirected graphs representing the order of a given lattice, see Figure 1 for two examples.

3.1.2. Lattice Varieties and Lattices' Omission

The field of lattice varieties emerged as a branch of inquiry stemming from the investigation of general varieties. Foundational contributions by Birkhoff (1935), Dedekind (1900), and Jonnson (1967) show examples of results describing lattice varieties characterized by the omission of a set of lattices.

Definition 3. Let $\mathcal{V}$ be a variety of lattices. Then $\mathcal{V}$ *omits* a lattice $\mathbf{L}$ if $\mathbf{L}$ is not a sublattice of any lattice in $\mathcal{V}$.

In the field of universal algebra, a parallel line of research aims to characterize lattices based on equational properties. Notably, properties such as distributivity and modularity are of particular interest in this regard.

Definition 4. Let $\mathbf{L}$ be a lattice. Then $\mathbf{L}$ is *modular* (*distributive*) if it satisfies the following:

$$(x \wedge y) \vee (y \wedge z) \approx ((x \wedge y) \vee z) \wedge y \qquad \text{(modularity)}$$
$$x \wedge (y \vee z) \approx (x \wedge y) \vee (x \wedge z) \qquad \text{(distributivity)}$$

Obviously, a distributive lattice is also modular (Burris & Sankappanavar, 1981a). On the other side, Figure 1 illustrates the non-modularity of $\mathbf{N}_5$ by considering the substitution $x = a$, $y = c$, $z = b$, while the same substitution demonstrates the non-distributivity of $\mathbf{M}_3$. These observations highlight that the classes of distributive and modular lattices constitute distinct varieties, exemplifying a classical instance of characterizing lattice varieties through lattice omissions.

Theorem 1 ((Dedekind, 1900))**.** *Let $\mathcal{V}$ be a lattice variety. Then $\mathcal{V}$ is modular variety if and only if $\mathcal{V}$ omits $\mathbf{N}_5$.*

Theorem 2 ((Birkhoff, 1935))**.** *Let $\mathcal{V}$ be a lattice variety. Then $\mathcal{V}$ is distributive variety if and only if $\mathcal{V}$ omits $\mathbf{N}_5$ and $\mathbf{M}_3$.*

Based on these foundational findings, the exploration of lattice omissions and structural characterizations of class of lattice has emerged as a flourishing and extensively investigated field. Notably, significant advancements have been made in the form of general characterizations of lattices that, when omitted, yield lattice varieties (Whitman, 1941). Additionally, extensive research has focused on characterizing congruence lattices associated with algebraic varieties (Aglianò et al., 2022; Kearnes & Kiss, 2013; Nation, 1974; Whitman, 1941). For a more in-depth exploration of this subject, we refer the reader to Jipsen & Rose (1992).

3.2. Graph Neural Methods

3.2.1. Graph Neural Networks

In the experimental setting we evaluate our approach making use of the following GNNs architectures.

SimpleGNN. `SimpleGNN` is a simple cascade of Graph Convolutional Network (GCN) layers introduced by Kipf and Welling (Kipf & Welling, 2016). For all nodes $v_i \in G$ in a graph G, A GCN layer is formulated as

$$h_{v_i}^{(\ell+1)} = \sigma\left(\sum_j \frac{1}{c_{ij}} h_{v_j}^{(\ell)} W^{(\ell)}\right)$$

where v_j are the neighboring nodes of v_i, c_{ij} is the normalizing constant for the edge (v_i, v_j), $W^{(\ell)}$ is the weight matrix of the ℓ-th layer of the neural network and σ is the non-linear activation function. c_{ij} corresponds to the elements of the symmetrically normalized adjacency matrix $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, where A is the adjacency matrix and D is the degree matrix of G. However, multiplication with A makes use of the features of only the neighboring nodes, and not that of the node itself. This is addressed by Kipf and Welling (Kipf & Welling, 2016) by adding self-loops to the adjacency matrix. As a result, c_{ij} are obtained from $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, where $\tilde{A} = A + I$ and $\tilde{D}$ is the corresponding degree matrix.

GIN. A Graph Isomorphism Network (GIN) (Xu et al., 2019) is a GNN implementation that generalizes the Weisfeiler-Lehman test and hence achieves maximum discriminative power among GNNs. Our `GIN` model consists of 8 `GINLayers` with the following message-passing equation:

$$h_v^{(k)} = \phi(h_v^{(k-1)}, f(h_u^{(k-1)} : u \in \mathcal{N}(v)))$$

where ϕ is an MLP and $h_v^{(k)}$ is the feature vector of node v at the k-th layer.

3.2.2. Graph Concept Explainer

The Graph Concept Explainer (GCExplainer) (Magister et al., 2021) is the first concept-based explainer for GNNs, which discovers and extracts concept in a unsupervised, post-hoc manner. GCExplainer leverages the observation that GNNs cluster nodes with similar features and neighborhoods in the activation space. Each of these clusters is analogous to a concept, which can be extracted by performing k-Means clustering on the node embeddings and assigning nodes membership to these clusters. The concepts can then be represented by visualising the n-hop neighborhood of the five node samples closest to each cluster centroid. Here, n is bounded by the number of message passing layers to align with the reasoning of the GNN.

The concepts extracted by this process can be quantitatively evaluated by concept *purity* and *completeness* metrics. Purity score is calculated by computing graph edit distance, which measures the number of operations such as node insertion/deletion that must be performed to transform a graph g_1 to another graph g_2. This is formulated as follows by Abu-Aisheh et al. (Abu-Aisheh et al., 2015):

$$GED(g_1, g_2) = \min_{e_1,\dots,e_k \in \gamma(g_1,g_2)} \sum_{i=1}^{k} c(e_i)$$

where c is the cost function for an edit operation e_i, $\gamma(g_1, g_2)$ is the set of edits that transforms g_1 into g_2, and k is the number of clusters. If the purity score is 0, then the concept is said to be pure.

Completeness score, as proposed by Yeh et al. (2020), is equivalent to the accuracy of a classifier such as decision tree, which takes concept c relating to a data point as input and predicts an output label y.

4. Methodology and Experimental Setup

4.1. Model Pipeline

The aim of this paper is to identify correspondences between equational properties of lattices and subportions of their graph representation. In particular, we focus our study on distributivity and modularity (cfr. Definition 4), using GNNs with a graph explainer. The overall pipeline describing our methodology is represented in Figure 2. Our approach involves two main steps: (i) A classification-step, utilizing GNN architectures to perform a classification task. The objective is to learn from a given dataset whether a lattice is modular or distributive. (ii) The node embeddings obtained from the GNN are forwarded to the concept-based graph explainer, namely `GCExplainer`. This explainer leverages the embeddings to confirm the labeling of the lattices and provide interpretability measures. Specifically, `GCExplainer` generates similarity clusters based on the original data, aggregating the embeddings by their centroids using k-means. The classes serve to group together similar topological substructures, resulting in a categorization that highlights the shared characteristics among them (as described in Section 3.2.2). This approach allows to gain insights into the reasons behind the classification results and obtain a better understanding of the underlying factors contributing to distributivity and modularity of the lattices. The utilization of `GCExplainer`, being a human-in-the-loop method, enables mathematicians to validate the accuracy of the predictions by comparing the concepts labels with those obtained through the GNN, corroborating mathematical statements via the explanations.

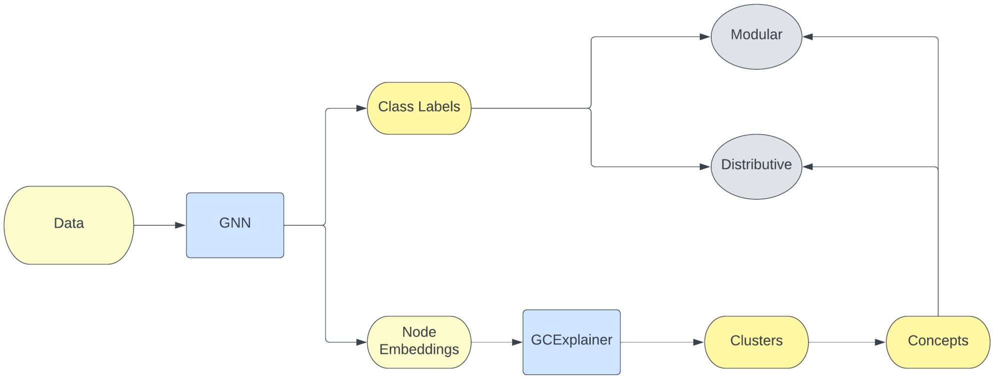


Figure 2: Pipeline visualization. Given a dataset of labelled lattices, we use a GNN to process the data in order to: (i) obtain class labels to check whether or not a lattice is modular/distributive, (ii) get node embeddings to be passed to GCExplainer, hence verifying that the GNN is capable of identifying relevant substructures for the classified properties. We have two procedures: the first is to use a GNN to directly obtain class labels to check whether or not a lattice is modular/distributive, and the next part is to pass node embeddings inferred from the GNN to GCExplainer in order to verify that the GNN indeed identifies relevant substructures for the classified.

4.2. Datasets

To experimentally evaluate our approach, both on the prediction task of algebraic properties and on its capacity of recovering known results of universal algebra about lattices' omission, we realized a suitable collection of lattices. As noted above, lattices are special graph structures, hence to realize a dataset of lattices, we simply need to consider common graphs up to a certain amount of nodes as candidates, and then discard the ones not satisfying Definition 2. Then, each lattice is labelled as distributive/modular depending on its satisfaction/violation of equations in Definition 4. In our experimental analysis, we employed three distinct datasets, namely `Sample_50`, `Sample_8` and `Sample_8_balanced`. These datasets consist of lattices with their respective graph representations in the form of Hasse diagrams. The datasets include automatically calculated labels regarding the modularity and distributivity properties of the lattices.

- `Sample_50` (1500 graphs with maximum cardinality of 50) for `SimpleGNN`;
- `Sample_8` (30000 graphs and maximum cardinality of 8) for `GIN`;
- `Sample_8_balanced` (248 graphs for each class, maximum cardinality of 8) for `GIN`;

Datasets Statistics. Splits between training, validation, and test was set at 60% - 20% - 20%. Both datasets exhibit a significant class imbalance concerning the satisfaction of the algebraic properties. In the considered datasets, 17% and 20% of all lattices in `Sample_50` are distributive and modular, respectively; in `Sample_8`, 16% and 19% are distributive and modular, respectively. Such an unbalance can be detrimental to the performance of GNNs, so we experiment with two different approaches to tackle this issue. We simply oversample distributive and modular lattices (positive labels) while training `SimpleGNN` on `Sample_50`. For a complete overview of all density concentration of classes in all the splits, please refer to Figure 3. To handle `Sample_8`'s highly unbalanced composition (Figure 3a), we realized an additional dataset `Sample_8_balanced` (Figure 3b) and use it with the same splits as above.

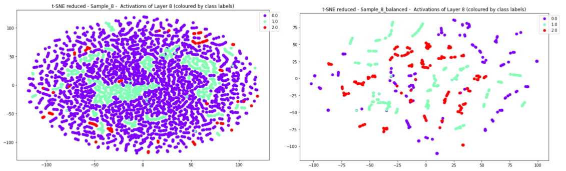
4.3. Experimented Models

SimpleGNN. In our experiments, we use 8 GCN layers with ReLU activation functions in between and Sigmoid function after the GCN cascade for classification. We use an embedding size of 128 for all GCN layers, learning rate of 0.001, and the model is trained for 1000 epochs. `SimpleGNN` is trained and evaluated on `Sample_50` dataset, and modular and distributive lattices are oversampled by a factor of 4 to overcome the issue of unbalance. Modularity and distributivity are handled separately, so that the model performs graph-level binary classification.

GIN. This model is trained for 30 epochs and, contrary to `SimpleGNN`, it performs multi-class classification. The classification space consists of the following 4 classes, which represent all possible combinations of (non)modularity and/or (non)distributivity:

Class 0	non-modular and non-distributive
Class 1	modular and distributive
Class 2	modular but non-distributive
Class 3	non-modular but distributive

Table 1: Classification accuracy for distributive and modular properties on validation and test splits



(a) Unbalanced dataset (b) Balanced dataset

Figure 3: Visualization of the node embedding spaces obtained by GIN with dimensionality reduction technique t-SNE. (a): Unbalanced case, class 0 is "not modular and not distributive", class 1 is "modular and distributive", and class 2 is "modular but not distributive". Class 0 is clearly dominating the sample space. (b): In the balanced version all classes are represented uniformly.

5. Results

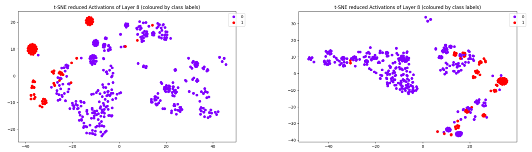
5.1. Experiments on SimpleGNN

5.1.1. GNN classification performance

	Validation	Test
Distributive	97.9 ± 0.99	97.3 ± 0.64
Modular	93.3 ± 2.27	92.3 ± 2.50

Table 2: Classification accuracy for distributive and modular properties on validation and test splits

We report the binary classification accuracy of `SimpleGNN` on distributivity and modularity properties in Table 4. Even this simple architecture can provide strong performance on classifying distributive lattices, and a slightly worse but still competitive performance on modular lattices. This indicates that by using GNNs we are able to correctly check the satisfaction of an equational property for a (potentially very large) lattice with a simple prediction step of the network with a cubic complexity with respect to the lattice dimension. However, the main strength of our approach relies on its capability of pointing out graph patterns considered relevant from the GNN to classify the predicted properties. For instance, the next section shows that GNN has also been able to identify the existence/omission of $\mathbf{N}_5$ and $\mathbf{M}_3$ lattices. We explore this possibility in the following sections with the help of GCExplainer.



(a) Distributivity dataset (b) Modularity dataset

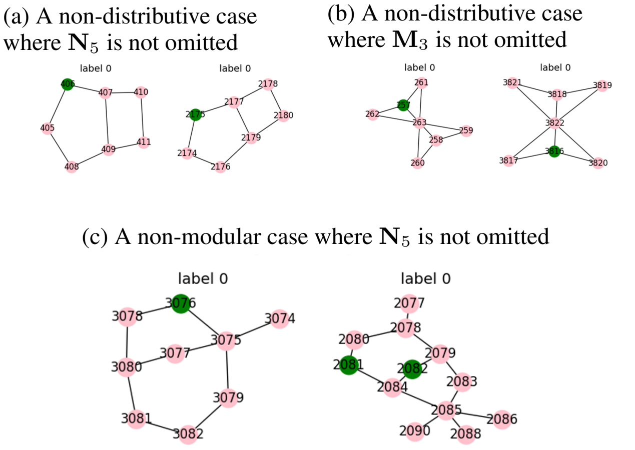
Figure 4: Visualization of the combined (validation + test) node embedding spaces obtained by SimpleGNN wth dimensionality reduction technique t-SNE. The plots visualize the scale of unbalance in the datasets.

5.1.2. GCExplainer qualitative results

Figure 4 visualises the node embeddings' distribution extracted by `SimpleGNN` and reduced to two dimensions by t-SNE algorithm. Each point in the scatter plot represents a graph in the respective dataset split, and the colors correspond to their true labels. We expect node embeddings to be separately clustered for different classes, which is mostly the case as seen in the plots. This provides a qualitative support to the discriminative power of the GNN.

An important insight on how the GNN determines whether or not a lattice is distributive, non-distributive, modular or non-modular is shown in Figure 5. These visualizations are obtained by GCExplainer, which groups the node embeddings to a user-controlled number of clusters, and plots k-hop neighborhoods of nodes in a given cluster. In this experiment, we divide the embedding space into 12 clusters and calculate 3-hop neighborhood of nodes. We expect to see substructures $\mathbf{N}_5$ and $\mathbf{M}_3$ when the label is 0, indicating that a lattice is non-distributive or non-modular. In fact, we observe $\mathbf{N}_5$ substructure in Figure 5a, and $\mathbf{M}_3$ substructure in Figure 5b. This matches Theorem 2, which states that a lattice that omits both $\mathbf{N}_5$ and $\mathbf{M}_3$ is distributive. In these examples, they do not omit either $\mathbf{N}_5$ or $\mathbf{M}_3$, hence they are non-distributive and correctly labeled as 0. Similarly, the lattices in Figure 5c do not omit $\mathbf{N}_5$ so they are non-modular by Theorem 1.

Figure 8 in Appendix B visualizes the decision trees obtained from the embedding spaces of validation splits of distributive and modular datasets. Since we have only one feature per node, only `x[0]` is relevant for making decisions.


(a) A non-distributive case where $\mathbf{N}_5$ is not omitted

(b) A non-distributive case where $\mathbf{M}_3$ is not omitted

(c) A non-modular case where $\mathbf{N}_5$ is not omitted

Figure 5: Example visualizations of 3-hop neighborhoods containing relevant substructures extracted by GCExplainer. The examples are taken from the clusters of validation sets.

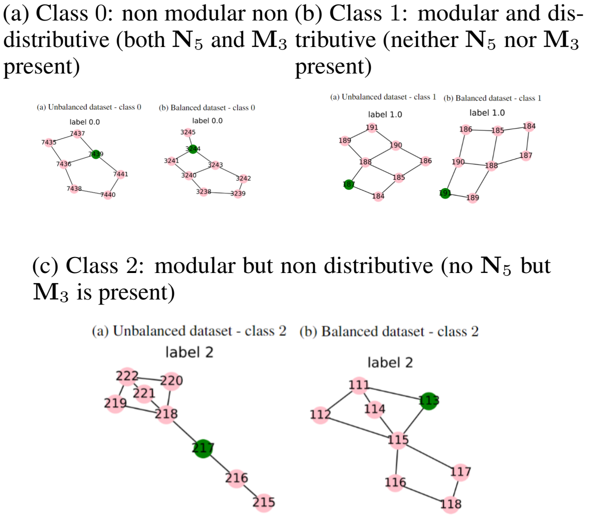
5.1.3. GCExplainer quantitative results

	Purity (↓)	Completeness (↑)
Distributive	2.57 ± 0.89	75.0 ± 1.82
Modular	1.44 ± 0.66	72.1 ± 1.56

Table 3: Purity and completeness scores of concepts extracted from the combined validation and test embeddings for both properties. (↓): lower is better. (↑): higher is better.

Table 3 presents the quantitative metrics of GCExplainer applied on combined validation and test embedding spaces obtained from SimpleGNN. We obtain around 72-75% for completeness scores, which suggests that the concept spaces obtained by `SimpleGNN` are consistent with the ground truth labels. We note that even though these results are considerably lower than the actual classification results shown in Table 4, we do not employ any oversampling strategy to balance the concept spaces that are fed into the decision tree. This shows that the GNN produces disentangled representations such that the resulting concept spaces can be accurately mapped to the ground truth labels even in the presence of heavily unbalanced datasets.

Purity scores are quite low considering that `Sample_50` has lattices with very high number of nodes, indicating that the concepts belonging to a class are quite similar to each other. We note, however, that the graph edit distance is extremely computationally expensive, so we only include graphs with a maximum number of 10 nodes in the calculation. Otherwise, computation of the purity score becomes practically intractable. This constraint limits the exploration of purity in datasets with large number of nodes.


(a) Class 0: non modular non distributive (both $\mathbf{N}_5$ and $\mathbf{M}_3$ present)

(b) Class 1: modular and distributive (neither $\mathbf{N}_5$ nor $\mathbf{M}_3$ present)

(c) Class 2: modular but non distributive (no $\mathbf{N}_5$ but $\mathbf{M}_3$ is present)

Figure 6: Examples of 3-hop neighborhoods containing relevant substructures extracted by GCExplainer with GIN.

Overall, the combination of low purity and high completeness scores provides evidence that even vanilla GNN can be significantly expressive to identify interesting universal algebra properties. The visualizations of the highlighted subpatterns that are extracted by the node embeddings further support the reliability of GNNs for their potential applications in mathematics.

5.2. Experiments on GIN

5.2.1. GNN classification performance

The experimental analysis with `GIN` has been carried out on three different datasets: the unbalanced dataset `Sample_8`, the unbalanced dataset `Sample_8_balanced`, and a dataset `One_class_at_time` consisting of representatives of only one class (three runs in total, one per class). For all the three datasets we report the following validation and test accuracies:

Datasets	Validation	Test
`Sample_8`	98.96 ± 0.53	98.5 ± 0.49
`Sample_8_balanced`	99.0 ± 0.61	96.5 ± 0.32
`One_class_at_time`	98.46 ± 0.66	98.53 ± 0.26

Table 4: Classification accuracy in test and validation splits for all datasets described in 5.2.1

5.2.2. GCExplainer qualitative results

The test embeddings (Figure 3) retrieved from `GCExplainer` were calculated with a clusteriza-

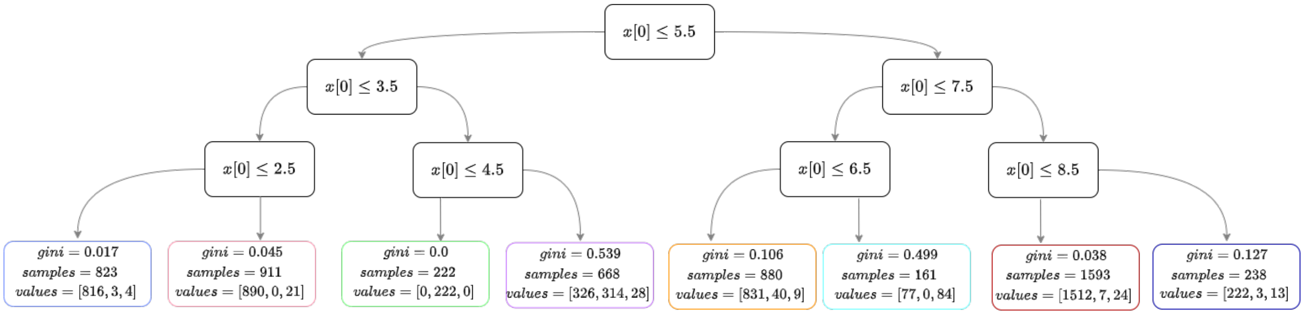

$x[0] \leq 5.5$
$x[0] \leq 3.5$
$x[0] \leq 7.5$
$x[0] \leq 2.5$
$x[0] \leq 4.5$
$x[0] \leq 6.5$
$x[0] \leq 8.5$
$gini = 0.017$ $samples = 823$ $values = [816, 3, 4]$
$gini = 0.045$ $samples = 911$ $values = [890, 0, 21]$
$gini = 0.0$ $samples = 222$ $values = [0, 222, 0]$
$gini = 0.539$ $samples = 668$ $values = [326, 314, 28]$
$gini = 0.106$ $samples = 880$ $values = [831, 40, 9]$
$gini = 0.499$ $samples = 161$ $values = [77, 0, 84]$
$gini = 0.038$ $samples = 1593$ $values = [1512, 7, 24]$
$gini = 0.127$ $samples = 238$ $values = [222, 3, 13]$

Figure 7: Decision tree.

tion hyperparameter $k = 10$ with a k-means hop of 3. Since the clusterisation regime is an upper bounded function (Magister et al., 2021) - i.e. it does not improve infinitely, instead it plateaus after a certain threshold - by using binary search is easy to find the ideal interval/value for it - in this case 10. As depicted in Figures 6a, 6b, and 6c, the explainer shows that in both `Sample_8` and `Sample_8_balanced` experiments, the concepts retrieved are consistent with the definitions of our classes and the classification done by the GNN.

5.2.3. GCExplainer quantitative results

As an extra quality assurance, we computed two additional measures: the completeness and purity calculations. In both cases we have that the number of leaves in the decision trees are equal to the number of concepts as we were expecting (`num_leaves` $= 10 = k$) and by starting at the head and going down by tracking the hyperparameters in the box, it is possible to map the GNN output to its concept. Table 5 reports the completeness and purity scores for both the unbalanced and balanced datasets on the test embeddings. The unbalanced dataset exhibits lower purity due to the large number of data entries. In contrast, the balanced dataset displays higher purity as all classes are uniformly represented, resulting in a reduced variance within the dataset.

Datasets	Purity (↓)	Completeness (↑)
Unbalanced dataset	3.04 ± 0.67	96.6 ± 2.03
Balanced dataset	3.59 ± 0.45	56.3 ± 1.77

Table 5: Completeness and purity scores of test embeddings for both datasets. (↓): lower is better. (↑): higher is better.

We also provide an explainable decision-making process that could be backtracked in order to ascertain the course that lead to the decision. In particular, this is done through decision trees obtained from the embedding spaces of validation splits of the datasets. In Figure 7, each leaf corresponds to a possible concept representing the data. Starting from the initial feature and depending on its value, at each step the explainer makes a prediction based on the likelihood of the value up until it gets assigned to a concept - i.e. up until it reaches the leaves. A more extensive example can be found in the Appendix B.

Limitations

This paper primarily investigates established theorems in the field of universal algebra, specifically concerning the equivalence between algebraic and topological definitions of distributive and modular lattices. We believe that our approach represents an initial step towards systematically studying various equational properties of lattice varieties, and their potential connection with graph-pattern identification. Moreover, it is important to note that our methodology is specifically designed for finite lattices, and therefore may not encompass all relevant aspects related to infinite algebraic structures. However, the insights and understanding gained from our explanations based on finite lattices can provide valuable contributions to specific problems, although there may be limitations in terms of generalization.

6. Conclusion and Future Work

This paper introduces an innovative approach that leverages established AI techniques to investigate equational and topological theorems in the domain of universal algebra. It aims to identify graph-patterns associated with well-known equational properties in universal algebra and presents a transparent methodology to validate theorems using graph-structured data. Following this methodology, we were able to empirically corroborate the theorems on distributive and modular lattices proposed by Dedekind (Dedekind, 1900) and Birkhoff (Birkhoff, 1935), respectively, by recovering pertinent lattices. We provide two datasets of labelled lattices, specifically categorized as either satisfying or violating the distributive and modular properties. Through our experiments, we demonstrate the effectiveness of our approach on these datasets by successfully identifying well-known graph patterns that characterize these properties in the field of universal algebra. Additionally, we investigate the quantitative metrics of GCExplainer, particularly the completeness and purity scores, to gain further insights into the interpretability and performance of our methodology. Our approach can be readily extended to investigate various structural properties of lattices, including the characterization of congruence lattices of algebraic varieties (Aglianò et al., 2022; Kearnes & Kiss, 2013; Nation, 1974; Whitman, 1941). Given that universal algebra serves as a foundational branch of modern mathematics, any contribution to this field carries significant implications for various mathematical disciplines.

Acknowledgements

This work was supported by TAILOR and by HumanE-AI-Net projects, which are funded by EU Horizon 2020 research and innovation programme under GA No 952215 and No 952026, respectively.

References

Abu-Aisheh, Z., Raveaux, R., Ramel, J.-Y., and Martineau, P. An exact graph edit distance algorithm for solving pattern recognition problems. In *4th International Conference on Pattern Recognition Applications and Methods 2015*, 2015.

Aglianò, P., Bartali, S., and Fioravanti, S. On Freese's technique. *arXiv:2302.11452*, 2022.

Arrieta, A. B., Díaz-Rodríguez, N., Ser, J. D., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, 2019.

Bansal, K., Loos, S., Rabe, M., Szegedy, C., and Wilcox, S. Holist: An environment for machine learning of higher order logic theorem proving. In *International Conference on Machine Learning*, pp. 454–463. PMLR, 2019.

Birkhoff, G. On the structure of abstract algebras. In *Mathematical proceedings of the Cambridge philosophical society*, volume 31, pp. 433–454. Cambridge University Press, 1935.

Birkhoff, G. *Lattice theory*, volume 25. American Mathematical Soc., 1940.

Burris, S. and Sankappanavar, H. P. *A Course in Universal Algebra*. Springer, 1981a.

Burris, S. and Sankappanavar, H. P. *A Course in Universal Algebra*. Springer, 1981b. URL http://www.math.uwaterloo.ca/~snburris/htdocs/ualg.html.

Davies, A., Veličković, P., Buesing, L., Blackwell, S., Zheng, D., Tomašev, N., Tanburn, R., Battaglia, P., Blundell, C., Juhász, A., Lackenby, M., Williamson, G., Hassabis, D., and Kohli, P. Advancing mathematics by guiding human intuition with AI. *Nature*, 600(7887):70–74, December 2021. doi: 10.1038/s41586-021-04086-x. URL https://doi.org/10.1038/s41586-021-04086-x.

Day, A. A characterization of modularity for congruence lattices of algebras*. *Canadian Mathematical Bulletin*, 12(2):167–173, 1969. doi: 10.4153/CMB-1969-016-6.

Dedekind, R. Über die von drei Moduln erzeugte Dualgruppe. *Math. Ann.*, 1900.

Eberhard, S. Product mixing in the alternating group. *Discrete Analysis*, 2016.

Fitting, M. *First-order logic and automated theorem proving*. Springer Science & Business Media, 2012.

He, Y.-H. Machine-learning mathematical structures. *International Journal of Data Science in the Mathematical Sciences*, pp. 1–25, 2022.

Holzinger, A., Saranti, A., Molnar, C., Biecek, P., and Samek, W. *Explainable AI Methods - A Brief Overview*, pp. 13–38. Springer International Publishing, Cham, 2022. ISBN 978-3-031-04083-2. doi: 10.1007/978-3-031-04083-2_2. URL https://doi.org/10.1007/978-3-031-04083-2_2.

Hyland, M. and Power, J. The category theoretic understanding of universal algebra: Lawvere theories and monads. *Electronic Notes in Theoretical Computer Science*, 172:437–458, 2007. ISSN 1571-0661. doi: https://doi.org/10.1016/j.entcs.2007.02.019. URL https://www.sciencedirect.com/science/article/pii/S1571066107000874. Computation, Meaning, and Logic: Articles dedicated to Gordon Plotkin.

Jipsen, P. and Rose, H. *Varieties of Lattices*. Springer Berlin, 1992.

Jonnson, B. Algebras whose congruence lattices are distributive. *MATHEMATICA SCANDINAVICA*, 21:110–121, Dec. 1967. doi: 10.7146/math.scand.a-10850. URL https://www.mscand.dk/article/view/10850.

Kearnes, K. and Kiss, E. The shape of congruence lattices. *Memoirs of the American Math. Soc.*, 2013.

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

Lample, G. and Charton, F. Deep learning for symbolic mathematics. *arXiv preprint arXiv:1912.01412*, 2019.

Loveland, D. W. Automated theorem proving: mapping logic into ai. In *Proceedings of the ACM SIGART international symposium on Methodologies for intelligent systems*, pp. 214–229, 1986.

Loveland, D. W. *Automated theorem proving: A logical basis*. Elsevier, 2016.

Magister, L. C., Kazhdan, D., Singh, V., and Liò, P. Gcexplainer: Human-in-the-loop concept-based explanations for graph neural networks. *arXiv preprint arXiv:2107.11889*, 2021.

Nation, J. B. Varieties whose congruences satisfy certain lattice identities. *Algebra Universalis*, 1974.

Paliwal, A., Loos, S., Rabe, M., Bansal, K., and Szegedy, C. Graph representations for higher-order logic and theorem proving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 2967–2974, 2020.

Peifer, D., Stillman, M., and Halpern-Leistner, D. Learning selection strategies in buchberger's algorithm. In *International Conference on Machine Learning*, pp. 7575–7585. PMLR, 2020.

Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

van der Velden, B. H., Kuijf, H. J., Gilhuijs, K. G., and Viergever, M. A. Explainable artificial intelligence (xai) in deep learning-based medical image analysis. *Medical Image Analysis*, 79:102470, 2022. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2022.102470. URL https://www.sciencedirect.com/science/article/pii/S1361841522001177.

Wagner, A. Z. Constructions in combinatorics via neural networks. *arXiv preprint arXiv:2104.14516*, 2021.

Wang, M., Tang, Y., Wang, J., and Deng, J. Premise selection for theorem proving by deep graph embedding. *Advances in neural information processing systems*, 30, 2017.

Whitman, P. M. Free lattices. *Annals of Mathematics*, pp. 325–330, 1941.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks?, 2019.

Yeh, C.-K., Kim, B., Arik, S., Li, C.-L., Pfister, T., and Ravikumar, P. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33:20554–20565, 2020.

Ying, R., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. Gnnexplainer: Generating explanations for graph neural networks, 2019.

A. Algebra definitions

A.1. Formal definitions for Universal Algebra

Universal algebra is the field of mathematics that studies algebraic structures, which are defined as a set A along with its own collection of operations. An n-ary operation on A is a function that takes n elements of A and returns a single element from the set. More formally (Burris & Sankappanavar, 1981b; Jonnson, 1967; Day, 1969):

Definition 5. For A non-empty set and n nonnegative integer we define $A^0 = \{\emptyset\}$ and, for $n > 0$, A^n is the set of n-tuples of elements from A. An n-ary operation (or function) on A is any function f from A^n to A; n is the arity (or rank) of f. An operation f on A is called an n-ary operation if its arity is n.

Definition 6. An *algebra* $\mathcal{A}$ is a pair $\langle A, F\rangle$ where A is a non-empty set called universe and F is a set of finitary operations on A.

Apart from the operations on A, an algebra is further defined by axioms, that in the particular case of universal algebras are often of the form of identities. The collection of algebraic structures defined by equational laws are called varieties. (Hyland & Power, 2007)

Definition 7. A nonempty class K of algebras of type $\mathcal{F}$ is called a *variety* if it is closed under subalgebras, homomorphic images, and direct products.

Definition 8. A *lattice* $\mathbf{L}$ is an algebraic structure composed by a non-empty set L and two binary operations $\vee$ and $\wedge$ satisfying the following axioms and their duals obtained exchanging $\vee$ and $\wedge$:

$$\begin{aligned} &x \vee y \approx y \vee x && \text{(commutativity)}\\ &x \vee (y \vee z) \approx (x \vee y) && \text{(associativity)}\\ &x \vee x \approx x && \text{(idempotency)}\\ &x \approx x \vee (x \wedge y) && \text{(absorption)} \end{aligned}$$

Theorem 3 ((Birkhoff, 1940))**.** *Definition 2 and Definition 8 are equivalent.*

Congruence lattices of algebraic structures are partially ordered sets such that every pair of elements has unique supremum and infimum determined by the underlying algebra. This object is important relatively to algebraic structures' properties, many of which can be described by omission or admission of certain subpatterns in a graph.

Definition 9. Congruence Lattice

For every algebra $\mathcal{A}$ on the set A, the identity relation on A, and $A \times A$ are trivial congruences. An algebra with no other congruences is called simple. Let $\mathrm{Con}(\mathcal{A})$ be the set of congruences on the algebra $\mathcal{A}$. Because congruences are closed under intersection, we can define a meet operation: $\wedge : \mathrm{Con}(\mathcal{A}) \times \mathrm{Con}(\mathcal{A}) \to \mathrm{Con}(\mathcal{A})$ by simply taking the intersection of the congruences $E_1 \wedge E_2 = E_1 \cap E_2$. Congruences are not closed under union, however we can define the closure operator of any binary relation E, with respect to a fixed algebra $\mathcal{A}$, such that it is a congruence, in the following way: $\langle E\rangle_{\mathcal{A}} = \bigcap\{F \in \mathrm{Con}(\mathcal{A}) \mid E \subseteq F\}$. Note that the closure of a binary relation is a congruence and thus depends on the operations in $\mathcal{A}$, not just on the carrier set. Now define $\vee : \mathrm{Con}(\mathcal{A}) \times \mathrm{Con}(\mathcal{A}) \to \mathrm{Con}(\mathcal{A})$ as $E_1 \vee E_2 = \langle E_1 \cup E_2\rangle_{\mathcal{A}}$. For every algebra $\mathcal{A}$, $(\mathrm{Con}(\mathcal{A}), \wedge, \vee)$ with the two operations defined above forms a lattice, called the congruence lattice of $\mathcal{A}$.

Definition 10. Let $\mathbf{A}$ and $\mathbf{B}$ be two algebras of the same type. Then $\mathbf{B}$ is a *subalgebra* of $\mathbf{A}$ if $B \subseteq A$ and every fundamental operation of $\mathbf{B}$ is the restriction of the corresponding operation of $\mathbf{A}$, i.e., for each function symbol f, $f^{\mathbf{B}}$ is $f^{\mathbf{A}}$ restricted to $\mathbf{B}$.

Definition 11. Suppose $\mathbf{A}$ and $\mathbf{B}$ are two algebras of the same type $\mathcal{F}$. A mapping $\alpha : A \to B$ is called a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ if

$$\alpha f^{\mathbf{A}}(a_1, \ldots, a_n) = f^{\mathbf{B}}(\alpha a_1, \ldots, \alpha a_n)$$

for each n-ary f in $\mathcal{F}$ and each sequence $a_1, \ldots, a_n$ from $\mathbf{A}$. If, in addition, the mapping α is onto then $\mathbf{B}$ is said to be a homomorphic image of $\mathbf{A}$.

Definition 12. Let $\mathbf{A}_1$ and $\mathbf{A}_2$ be two algebras of the same type $\mathcal{F}$. We define the *direct product* $\mathbf{A}_1 \times \mathbf{A}_2$ to be the algebra whose universe is the set $A_1 \times A_2$, and such that for $f \in \mathcal{F}$ and $a_i \in A_1$, $a_i' \in A_2$, $1 \leq i \leq n$,

$$f^{\mathbf{A}_1 \times \mathbf{A}_2}(\langle a_1, a_1' \rangle, \ldots, \langle a_n, a_n' \rangle) = \langle f^{\mathbf{A}_1}(a_1, \ldots, a_n), f^{\mathbf{A}_2}(a_1', \ldots, a_n') \rangle$$

B. Decision Trees by GCExplainer

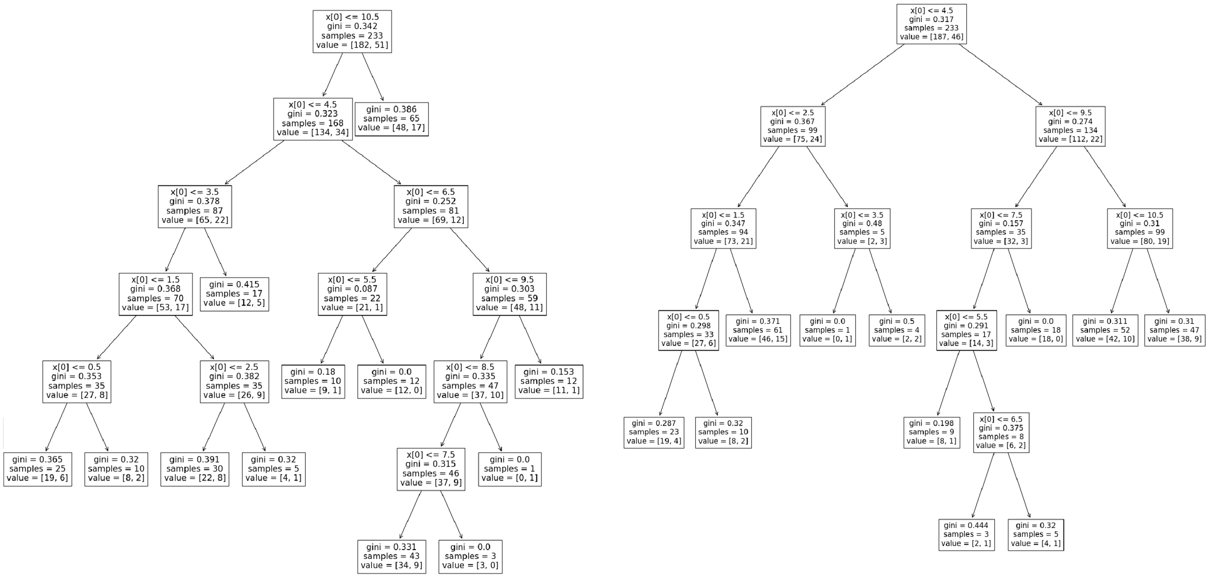

x[0] <= 10.5 gini = 0.342 samples = 233 value = [182, 51]
x[0] <= 4.5 gini = 0.323 samples = 168 value = [134, 34]
gini = 0.386 samples = 65 value = [48, 17]
x[0] <= 3.5 gini = 0.378 samples = 87 value = [65, 22]
x[0] <= 6.5 gini = 0.252 samples = 81 value = [69, 12]
x[0] <= 1.5 gini = 0.368 samples = 70 value = [53, 17]
gini = 0.415 samples = 17 value = [12, 5]
x[0] <= 5.5 gini = 0.087 samples = 22 value = [21, 1]
x[0] <= 9.5 gini = 0.303 samples = 59 value = [48, 11]
x[0] <= 0.5 gini = 0.353 samples = 35 value = [27, 8]
x[0] <= 2.5 gini = 0.382 samples = 35 value = [26, 9]
gini = 0.18 samples = 10 value = [9, 1]
gini = 0.0 samples = 12 value = [12, 0]
x[0] <= 8.5 gini = 0.335 samples = 47 value = [37, 10]
gini = 0.153 samples = 12 value = [11, 1]
gini = 0.365 samples = 25 value = [19, 6]
gini = 0.32 samples = 10 value = [8, 2]
gini = 0.391 samples = 30 value = [22, 8]
gini = 0.32 samples = 5 value = [4, 1]
x[0] <= 7.5 gini = 0.315 samples = 46 value = [37, 9]
gini = 0.0 samples = 1 value = [0, 1]
gini = 0.331 samples = 43 value = [34, 9]
gini = 0.0 samples = 3 value = [3, 0]
x[0] <= 4.5 gini = 0.317 samples = 233 value = [187, 46]
x[0] <= 2.5 gini = 0.367 samples = 99 value = [75, 24]
x[0] <= 9.5 gini = 0.274 samples = 134 value = [112, 22]
x[0] <= 1.5 gini = 0.347 samples = 94 value = [73, 21]
x[0] <= 3.5 gini = 0.48 samples = 5 value = [2, 3]
x[0] <= 7.5 gini = 0.157 samples = 35 value = [32, 3]
x[0] <= 10.5 gini = 0.31 samples = 99 value = [80, 19]
x[0] <= 0.5 gini = 0.298 samples = 33 value = [27, 6]
gini = 0.371 samples = 61 value = [46, 15]
gini = 0.0 samples = 1 value = [0, 1]
gini = 0.5 samples = 4 value = [2, 2]
x[0] <= 5.5 gini = 0.291 samples = 17 value = [14, 3]
gini = 0.0 samples = 18 value = [18, 0]
gini = 0.311 samples = 52 value = [42, 10]
gini = 0.31 samples = 47 value = [38, 9]
gini = 0.287 samples = 23 value = [19, 4]
gini = 0.32 samples = 10 value = [8, 2]
gini = 0.198 samples = 9 value = [8, 1]
x[0] <= 6.5 gini = 0.375 samples = 8 value = [6, 2]
gini = 0.444 samples = 3 value = [2, 1]
gini = 0.32 samples = 5 value = [4, 1]

Figure 8: Decision trees of distributive (left) and modularity (right) datasets extracted by SimpleGNN + GCExplainer.