PH.D. COMPREHENSIVE EXAM

TOWARDS A COMPREHENSIVE UNDERSTANDING OF TECHNIQUES FOR GRAPH-BASED NETWORK REPRESENTATION

by

Justin Carpenter

Synthesis Article

Advisor: Dr. Edoardo Serra

Committee Members: Dr. Spezzano Francesca, Dr. Jain Amit, Dr. Babinkostova Liljana

Boise State University

November 2024

ABSTRACT

Graph representations are crucial in domains such as social networks and bioinformatics, where detailed graph analysis is needed. Recent research trends have shifted away from random walk and matrix factorization approaches, focusing instead on Graph Neural Networks (GNNs) and Graph Representation Learning. However, GNNs face limitations like overfitting and difficulties in capturing the complete graph structure. Traditional graph representation learning methods also struggle with capturing complex structures due to limited neighborhood depth.

Node representation learning (NRL) aims to generate numerical vectors (embeddings) for the nodes of a graph, capturing meaningful structural information. In structural NRL, embeddings are designed to reflect the structural roles of nodes within the graph, making them particularly effective for tasks like node classification, where nodes belonging to the same class often share similar structural features, even if they are not directly connected. This is different from proximity-based methods, where embeddings are influenced by the spatial closeness of nodes within the graph.

The state-of-the-art in graph representation learning now increasingly relies on higher-order Weisfeiler-Lehman (WL) tests. This evolution is motivated by the need for more precise and detailed graph representations, which are critical for advanced applications. While higher-order WL tests, such as 3-dimensional variants, provide enhanced representations, they still face challenges related to computational ineffi-

ciency and high complexity, especially for medium to large graphs.

TABLE OF CONTENTS

ABSTRACT iv

1 INTRODUCTION 1

1.1 Research Scope and Problems Addressed 2

1.2 Proposed Solutions and Approaches 2

1.3 Literature Review and Methodology 3

1.4 Principal Findings and Contributions 3

2 METHODOLOGY ANALYSIS 5

2.0.1 Graph Representations 5

2.0.2 Structural Representation Approach 7

2.0.3 Inductive Representational Approaches 9

3 GOING BEYOND 1WL 11

3.1 Initial Steps Towards Higher-Order GNNs 11

3.2 Advancements that go Beyond 1WL 13

3.2.1 Observed Areas in Need of Improvement 17

4 ADDRESSING COMPLEXITY IN HIGHER-ORDER GRAPH REPRESENTATIONS 18

4.1 Advancements Beyond the 1WL Test 18

4.1.1 Higher-Order Weisfeiler-Lehman Approaches 19
4.1.2 Subgraph-Based Methods 21
4.2 Graph Partitioning Approaches and Structural Integrity 22
4.3 Architectural and Methodological Improvements 23
4.3.1 Feature Selection and Regularization 23
4.3.2 Inductive and Scalable Solutions 24
4.4 Integrating Structural and Architectural Advances 25
4.4.1 Balancing Expressiveness and Scalability 25
4.4.2 Hybrid Methods 25
4.5 Discussion 26
4.6 Conclusion and Future Directions 27

REFERENCES 28

CHAPTER 1:

INTRODUCTION

Graphs offer a rich source of information that surpasses what is achievable with standard vector representations. Graph representation learning automates the process of capturing complex structures by generating vector-like embeddings that preserve key features of nodes and relationships within the graph. The focus of this research is to understand the strengths and limitations of various techniques for graph-based network representation, specifically those related to Graph Neural Networks (GNN) and the Weisfeiler-Lehman (WL) tests.

Recent advancements in graph representation learning have increasingly relied on techniques such as higher-order WL tests and GNNs, which provide a more expressive way to capture complex graph structures. Gilmer *et al.* (2017); Hamilton *et al.* (2017); Kipf & Welling (2017) We will explore many well known works that have laid the groundwork for extending graph representation techniques through various approaches to WL tests, GNNs, and their theoretical properties. These works have focused on extending the expressiveness and applicability of GNNs in tasks such as distinguishing non-isomorphic graphs and understanding subgraph hierarchies. Maron *et al.* (2019); You *et al.* (2019); Yan *et al.* (2021)

1.1 Research Scope and Problems Addressed

The primary problem addressed in this paper revolves around improving the expressive power and scalability of GNNs for graph representation. Current state-of-the-art methods, such as Graph Isomorphism Networks (GIN) and higher-order WL tests, provide improved representational capabilities but continue to face significant challenges. The primary problems lie in the increased computational cost and tendency to overfit. Morris *et al.* (2019); Xu *et al.* (2019) We will also explore many current proposed solutions, such as graph partitioning approaches and why these solutions often fail to retain the structural integrity of the graph. We will further analyze precisely how the loss of structural integrity negatively impacts learning outcomes. Zhang *et al.* (2023); Stanton & Kliot (2012)

1.2 Proposed Solutions and Approaches

To address these problems, this paper explores a range of solutions proposed in the prior research:

- **Higher-Order WL Tests:** Increasing the expressiveness of GNNs by utilizing higher-order variants of the Weisfeiler-Lehman tests, enabling models to distinguish more complex graph structures. Morris *et al.* (2019); Feng *et al.* (2024)

- **Structural Representation Learning:** Leveraging approaches that focus on structural roles rather than proximity-based relationships to create more informative graph embeddings. Barceló *et al.* (2020); You *et al.* (2019)

- **Graph Partitioning and Subgraph-Based Approaches:** Introducing meth-

ods to efficiently partition large graphs while retaining critical structural information. Zhang *et al.* (2023, 2024) This helps in scaling the models without compromising the representation quality.

- **Feature Selection and Regularization:** Applying feature selection and novel regularization techniques to control overfitting and manage computational complexity. Xu *et al.* (2020); Hamilton *et al.* (2017)

1.3 Literature Review and Methodology

The current body of literature emphasizes the limitations of standard GNNs in their ability to represent complex structural information. While GNN models such as GIN still receive promising results in overcoming these limitations. Maron *et al.* (2019); You *et al.* (2019) However, still are computationally inefficient and unable to scale to large graphs. In recent works, there has been an increasing number of research into feature selection mechanisms, graph partitioning, and a combination of shallow architectures to enhance and build upon the existing methodologies and reduce the computational cost. Liu *et al.* (2023); Corso *et al.* (2020); Feng *et al.* (2024); Bader *et al.* (2013)

1.4 Principal Findings and Contributions

The principal findings of this research indicate that a combination of higher-order WL-based methods and efficient feature selection can significantly enhance the expressiveness and scalability of graph representation models. We find that while higher-order GNNs provide greater discriminative power, they require careful tuning and novel regularization to avoid overfitting and ensure scalability for large graphs. Still, the most current attempt in reduction is still below that, which is needed to run large

real world graphs. The use of graph partitioning strategies allows for better scalability, however it increases the risk of losing structural information. Bader *et al.* (2013); Stanton & Kliot (2012); Patwary *et al.* (2019) The contributions of this paper are thus threefold: Fully exploring the usages of different GNN models, the usage of different WL implementations and the impacts of higher-order, and the different proposed methods which are added on top of prior structural sound implementations in an attempt to overcome computational complexity. Feng *et al.* (2024); Zhang *et al.* (2024, 2023); Liu *et al.* (2023)

The rest of the paper is structured as follows: **Chapter 2:** discusses the current methodologies in greater detail, including GNNs with an increased expressivity over 1WL either through message passing, graph partitioning or a higher order WL. **Chapter 3:** discuss the current research into solving the computational efficiency either on a proposed methodology or on past methodologies that suffer from the high computational cost. **Chapter 4:** presents the limitations of current methodologies and proposed solutions. Followed by a discussion on the future solutions and the need to continue towards a better implementation or solution.

CHAPTER 2:

METHODOLOGY ANALYSIS

This section is an overview of what graph representations are and the different areas of research being done to capture them. The While focusing on enhancing the expressiveness of graph representation techniques and addressing scalability challenges.

2.0.1 Graph Representations

There is an ever growing interest in developing effective and efficient unsupervised and supervised graph representation techniques for node or edge classification and prediction. In order for Machine Learning (ML) applications to use a graph's features, representation learning methods are needed to render the data for use.

Graph Neural Networks (GNNs) have revolutionized how we process graph-structured data by leveraging node feature aggregation to learn rich representations for each node. However, GNNs' expressiveness is inherently tied to the ability to model higher-order relationships and the depth of neighborhood aggregation. For an undirected graphs represented as $G =< V, E, X >$ where X denotes the features for each node, their expressive power is often gauged by their ability to approximate a graph isomorphism test. To argue that a GNN is as powerful as a certain isomorphism test algorithm m, it must be proven that for any two non-isomorphic graphs G_1 and G_2, as determined by the test m, their node representations are distinct. Conversely, it must

also be shown that for any two graphs deemed possibly isomorphic by the heuristic test m, their node representations align.

There have been many different representation learning techniques applied in different areas of study that achieved great success. Some of which are DeepWalk Perozzi *et al.* (2014) which continued with the well known NLP Skip-Gram model, World2Vec Grohe (2020); Mikolov *et al.* (2013) which generated word representations by taking advantage of word sequences (sentences) and using them to optimize a neighborhood. DeepWalk generalized the Skip-Gram model from sequences of words to graphs. DeepWalk uses a procedure similar to a Depth-First traversal of the graph in combination with the neighborhood aggregation which results in a connectivity-based representation learning method where nodes sharing similar neighbors in a direct (first-order proximity) or indirect(higher-order proximity) fashion are located closer in the resulting latent space. Following DeepWalk's idea to use Depth-First traversal, LINE Tang *et al.* (2015) proposed using a Breadth-First traversal where nodes sharing the same edge (first-order proximity) are located closer in the latent space. The problem arises when the graph is not fully connected which is a requirement for Breadth-First. Performance of each are similar compared to Node2vec Grover & Leskovec (2016) which uses a random walk procedure that interpolates between both the Depth-First and Breadth-First topology. This removes the connectivity requirement while maintaining better performance.

All these approaches tend to preserve connectivity information among nodes, however they lack in preserving structural properties which are crucial in node and edge classification or prediction. The other primary problem in each of these methods is capturing deeper structural information at scale, or even handle large graphs at all.

2.0.2 Structural Representation Approach

The WL Test is a very effective way to identify true structural similarities further than the nearest neighbors. For two static graphs G_1 and G_2 with their respective node features X^1 and X^2, the K-WL approach is delineated in a defined algorithmic process. This is achieved by initially assigning a color (color refers to a discrete node label) to each vertex of a graph, then applying a color refinement algorithm through iterations. This color refinement algorithm is a means to aggregate the neighborhood of the graph. In the 1WL, a vertex color is then refined by combining the colors of its neighboring vertices until the convergence criterion is satisfied.Morris *et al.* (2020); Shervashidze *et al.* (2011) Given the potential to gather more information

Algorithm 1 1-Dimensional Weisfeiler-Lehman (WL) Algorithm

```
function WLAlgorithm(G)
    Initialize node labels L for each node in G
    Initialize iteration counter t to 0
    repeat
        Initialize empty multiset M
        for each node v in G do
            N(v) ← sorted list of neighbors of v in G
            h(v) ← hash of L[v] concatenated with the sorted labels of N(v)
            Add h(v) to M
        end for
        Increment t by 1
        for each node v in G do
            Compute the new label L'[v] using the multiset M and the current label L[v]
        end for
    until no label changes or maximum number of iterations reached
    return Final node labels L
end function
```

from graphs than standard vectorial representations, many graph analysis methods

on many real-world applications have been developed. Many structural approaches aim to emulate 1WL or similar approaches. These representation learning methods have a similar goal in creating vector like representations of the information within a graph. The approach is to utilize attention layers such as attention layers, which prove to retain good structural information.Liu *et al.* (2023) This approach has been built on to achieve higher structural information in order to predict future connections such as A3t-gcn,Bai *et al.* (2021) which use a combination of soft attention and hard attention mechanisms with an encoder/decoder for traffic flow prediction. The attention mechanism is used to capture temporal information correlations of the temporal changes graph, and the global temporal information and spatial correlation are captured to predict future traffic forecasts. In other approaches, to achieve usable information extracted from graphs researchers have looked towards node proximity. This requires graphs to be traversed, with approaches such as Depth-first Sampling (DFS) or Breadth-first Sampling (BFS) in a random walk approach. Works like DeepWalk Perozzi *et al.* (2014) use a DFS similarly LINE Tang *et al.* (2015) uses a BFS approach. Both tend to be computationally expensive and tend to lack certain structural information. A proposed random walk approach that utilizes both DFS and BFS, is Node2vec, which the authors argue provides representations that closely correspond with the structural role similarity. However, the issue with proximity is a large amount of information of the graph's overall structure is lost. Grover & Leskovec (2016)

Graph isomorphism networks (GINs) are a type of graph neural network that, as the title suggests, is designed to be powerful enough to distinguish between different graph structures. However, it is crucial to note that GINs must reach an optimal

parameter set during training to achieve the structural expressive power equivalent to WL. This condition is particularly relevant when the degree of nodes is bounded and the support feature set is finite, suggesting a unique set of parameters exists that satisfies the theorem for any graph pair. The subsequent section presents initial results illustrating the GIN's performance limitations in structural tasks. The model is rooted in the WL test of graph isomorphism, a heuristic method for determining whether two graphs are isomorphic, i.e. identical up to a relabeling of the vertices. The authors of the paper propose an architecture that generalizes the WL test to use neural networks. The GIN model operates by performing several "update" and "aggregate" steps, each of which involves updating node features based on the features of neighboring nodes, and then aggregating these features in some way. The model uses a learnable parameter ε to prevent over-smoothing and preserve the uniqueness of node representations. The authors demonstrate that the model is both theoretically and empirically powerful, capable of distinguishing between almost all non-isomorphic graphs and achieving state-of-the-art performance on several benchmark datasets.

2.0.3 Inductive Representational Approaches

Additional inductive representational approaches focus on node embeddings as a function of time and represented as static and evolving topological structure. The authors use self-attention and stacking layers based on time to create node embeddings as functions of time. This approach is able to handle node classification and link prediction. Continuous representational learning approach on temporal interaction graphs uses an encoder/decoder for the features with a continuous inference block and with a graph attention layer. The work focuses on link prediction and node recommendation and classification on the Wikipedia, Reddit, MOOC, and CollegeMsg

datasets. Using a Standard Vector Model (SVD) on dynamic networks has shown to be extremely efficient but not very effective for this task. The problem that SVD has is when to restart the SVD, which the authors claim to solve in order to reconstruct the dynamic network or in other terms link prediction. This work provide an alternative approach to detect evolving dynamic networks using the triadic closure process to semi-supervised algorithm for link prediction, vertex prediction, link reconstruction, and changed link prediction on evolving graphs. In a more simple approach, the authors attempt to create usable embeddings on temporal networks by sequencing neighborhood formation through sequential events using a Hawkes process. When applying an attention mechanism followed by the Euclidean distance function and optimization using softmax and negative sampling to create the final embeddings. The authors evaluated the embeddings using node classification and link prediction on DBLP, Yelp, and Tmall datasets.Xu *et al.* (2020); Yan *et al.* (2021); Zuo *et al.* (2018); Zhang *et al.* (2018)

CHAPTER 3:

GOING BEYOND 1WL

3.1 Initial Steps Towards Higher-Order GNNs

The WL test has been extensively utilized in various studies focused on structural node representation learning in graphs. Below, we highlight several proposed methodologies presented which could achieve higher expressive power, particularly in distinguishing non-isomorphic graphs that the 1WL test fails to differentiate. Though most achieved small improvements, the cost of structural integrity and computational cost, the methodologies still have a need for improvements. Some of the strong approaches utilize GNN with an additional layer such as Graph Convolutional Networks (GCNs) which preforms recursive operations on graph data in the convolutional layer to learn node embeddings, Graph Attention Networks (GATs) employ attention mechanisms to weight the importance of neighboring nodes, and Recurrent Neural Networks (RNNs) with gated units.Sutskever *et al.* (2014); Xu *et al.* (2018); Liu *et al.* (2023); Defferrard *et al.* (2016); Berg *et al.* (2017); Xu *et al.* (2018)

The feature selection mechanism allowed for reducing the dimensionality of graph features while retaining the most informative aspects, leading to enhanced model performance. Regularization techniques, such as soft-selection, helped prevent overfitting, particularly in the deeper GNN variants.Hamilton *et al.* (2017); Veličković

et al. (2018)

The proposed idea to create a completed model which focused on extracting features and properly implemented multiple layers on top of a Neural Network, was one major leap towards higher-order implementations. It was a novel approach to create Select Assemble-Normalize Architecture called PATCHY-SAN. The authors focused more on being able to efficiently handle discrete and continuous node vertex and edge attributes while still having a versatile solution fit for small, limited featured graphs or large, rich in features graphs. PATCHY-SAN utilizes ideas which we now know are crucial to achieving higher expressive power and still handle large scale graphs. The PATCHY-SAN architecture takes advantage of a lightweight WL kernel and is able to operate on substructures of larger graphs. It may not fully utilize subgraphs nor does it utilize higher dimension WL but it does allow the architecture to be implemented on top of a GNN to graph-structured data. This limited the past limitations in higher-order GNNs and achieve higher expressively than past attempts.Niepert *et al.* (2016) **The PATCHY-SAN CNN (PSCN) operates in three steps:**

1. **Normalization:** A normalization procedure that selects a fixed-size neighborhood around each node. This is done by defining a node sequence for each graph based on a breadth-first search (BFS) strategy. The BFS starts from a node with the highest centrality score and proceeds by adding nodes to the sequence based on their distance from the starting node.

2. **Extraction and expansion:** The model extracts local patches around each node in the sequence. Each patch is a fixed-size neighborhood around a node. The patches are then expanded into a matrix representation that can be processed by a standard Convolutional algorithm to prioritize important structures.

3. **Convolution:** The model then applies a 1D convolution over the matrix representation of each patch. The convolutional layer is followed by a dynamic pooling layer that aggregates the information from all patches to form a graph-level representation.

The authors demonstrate the effectiveness of their approach in several graph classification tasks, showing that their method outperforms other graph kernel methods. The PSCN model is particularly effective for graphs in which node labels provide useful information for the classification task.

3.2 Advancements that go Beyond 1WL

The implementation of Position-aware Graph Neural Networks (PGNN) provides an in-depth analysis of GNNs and their limitations, as well as the potential improvements that can be achieved through Random Node Initialization (RNI). The paper proves that GNNs with RNI are universal, a first such result for GNNs that do not rely on computationally demanding higher-order properties. The universality result holds even with partially randomized initial node features and preserves the invariance properties of GNNs in expectation. The empirical findings of the paper support the superior performance of GNNs with RNI over standard GNNs. In fact, the performance of GNNs with RNI is often comparable to or better than that of higher-order GNNs, while keeping the much lower memory requirements of standard GNNs. However, this improvement typically comes at the cost of slower model convergence.You *et al.* (2019)

GSN is a topologically-aware message passing scheme based on substructure encoding. The authors show that GSN is strictly more expressive than the WL test and provide sufficient conditions for universality. Importantly, GSN does not attempt to

adhere to the WL hierarchy, allowing it to retain attractive properties of standard GNNs such as locality and linear network complexity, while being able to disambiguate even hard instances of graph isomorphism. The authors perform an extensive experimental evaluation on graph classification and regression tasks and obtain state-of-the-art results in diverse real-world settings including molecular graphs and social networks. In the GSN model, each node's contribution is transformed differently depending on its structural relationship with the central node. This relationship is expressed by counting the appearance of certain substructures. Since substructure counts are vertex invariants, they are invariant to vertex permutations, the resulting GNN will be invariant to isomorphism. The authors also discuss the concept of structural roles in graphs and how they can be used to enhance the expressive power of GNNs. They propose to explicitly encode structural roles as part of message passing, in order to capture richer topological properties.

The paper "Natural Graph Networks" introduces a novel concept to the field of GNNs, known as Natural Graph Networks (NGNs). The key idea behind this paper is to process a graph in a way that is not dependent on how the graph is described. This is traditionally achieved by making the network equivalent to node permutations. However, the authors argue that instead of using equivalence, the more general concept of "naturally" is sufficient for a graph network to be well defined. The authors define two types of natural graph networks: global and local. Local natural graph networks are seen as conventional message-passing GNNs, but offer more flexibility. The paper also presents a practical instance of a natural network on graphs which uses an equivalent message passing network. This instance was shown to yield good performance on several benchmarks.de Haan *et al.* (2020)

The model Message Passing Simplicial Networks (SIN) is designed to capture topological information in data by extending the concept of message-passing neural networks to simplicial complexes, which are generalizations of graphs that can capture higher-order interactions. SIN is based on the WL test 1, a graph isomorphism test that updates node labels based on their local neighborhood. The authors extended this concept to simplicial complexes, creating a new algorithm called the Simplicial Weisfeiler-Lehman (SWL) algorithm. This algorithm updates the labels of simplices (the higher-dimensional equivalent of nodes) based on their local neighborhood in the simplicial complex. The SIN model uses the SWL algorithm to propagate information across the simplicial complex. It does this by applying a series of transformations to the simplex labels, which are then aggregated to form the final output of the model. Transformations and aggregations are learned from the data, allowing the model to adapt to the specific topological structures present. The authors also discuss the computational complexity of their model. They note that the complexity is dependent on the number of nodes, edges, and feature dimensions, as well as the number of propagation layers and negative samples used. They also suggest potential ways to reduce the computational load, such as using Strassen's Algorithm to compute the power of the adjacency matrix more efficiently. Bodnar *et al.* (2021)

Building upon the Feature Selection Graph Neural Network (FSGNN) model and introducing further architectural improvements to enhance the performance of GNNs for node classification tasks. The researchers emphasizes the importance of simplicity in the design of GNNs and argues that complex models are not always necessary for achieving high performance. The authors propose several architectural improvements that can be applied to existing GNN models to enhance their performance without

significantly increasing their complexity. Similar to the FSGNN model, the authors employ a feature selection mechanism that assigns weights to different features based on their importance. This allows the model to focus on the most informative features and ignore irrelevant or noisy ones. The authors demonstrate that a shallow (2-layered) architecture can achieve competitive performance compared to deeper models. By carefully designing the layers and incorporating the feature selection mechanism, the model can learn meaningful representations without the need for many hidden layers. The paper introduces several regularization techniques that help prevent overfitting. These include dropout, L2 regularization, and a novel technique called "soft-selection" that constrains the weights assigned to different features. The over-smoothing problem acknowledged where the representations of different nodes become too similar as information is propagated through the graph. By using a shallow architecture and carefully controlling the propagation of information, the model avoids over-smoothing and retains distinct representations for different nodes. The proposed improvements are designed to be computationally efficient, allowing the model to be trained and deployed on large graphs without significant computational overhead. In the context of DGCNs, diffusion-convolution combines the ideas of feature learning, parameter tying, and invariance that are characteristic of convolutional neural networks. The core operation of a DGCN is a mapping from the nodes and their features to the results of a diffusion process that starts at that node. Unlike standard GNNs, DGCN parameters are tied according to search depth rather than their position on a grid. The diffusion-convolutional representation is invariant with respect to the index of the node rather than the position. Atwood & Towsley (2016)

3.2.1 Observed Areas in Need of Improvement

Despite significant advancements in GNN architectures, there are several observed limitations and challenges that still need addressing:

- **Scalability to Large Graphs:** High-order WL and subgraph-based approaches, while effective, are computationally expensive when applied to large graphs. The complexity introduced by managing k-tuples or multiple subgraphs often results in prohibitive memory and time requirements.

- **Over-Smoothing in Deep GNNs:** As the depth of GNNs increases, the node representations tend to converge, resulting in the loss of unique features across nodes. This phenomenon, known as over-smoothing, limits the depth to which GNNs can effectively be trained without compromising expressiveness.

- **Loss of Structural Information in Graph Partitioning:** Techniques such as graph partitioning and hierarchical aggregation aim to address scalability but often lead to a loss of global structural information. This trade-off between scalability and structural integrity is a critical challenge, particularly for applications where retaining the full scope of graph relationships is essential.

- **Computational Complexity:** The adoption of methods like k-WL, subgraph aggregation, and equivariant approaches can result in increased computational load, which is not always feasible for real-time or large-scale applications. There is a need for more efficient models that can maintain a balance between high expressiveness and computational efficiency.

CHAPTER 4:

ADDRESSING COMPLEXITY IN HIGHER-ORDER GRAPH REPRESENTATIONS

As graph representation learning evolves, there is an increasing demand for models that can capture complex structural information to enhance their expressive power. However, this pursuit often introduces significant challenges, particularly related to computational efficiency and scalability. Higher-order methods, while theoretically promising, can bring about complexities that hinder practical applicability, especially with large-scale graphs. In this chapter, we explore various solutions proposed in recent research to mitigate these complexities. We focus on approaches that maintain or enhance structural integrity while addressing computational feasibility.

4.1 Advancements Beyond the 1WL Test

The WL Test has been foundational in graph isomorphism testing and still has significant influence over the development of GNNs. Traditional GNNs based on the 1WL test, such as GINs, even with large research into features or additional layers, many still have limitations in achieving expressiveness higher than 1WL; particularly in distinguishing non-isomorphic graphs that the 1WL test fails to differentiate. This

limitation has spurred efforts to develop higher-order GNNs that extend beyond the capabilities of the 1WL test.Xu *et al.* (2019)

4.1.1 Higher-Order Weisfeiler-Lehman Approaches

Higher-order WL tests 1, such as the 3-dimensional variant, were employed to improve the discriminative power of GNNs in capturing complex graph structures. These tests enhance the ability of GNNs to distinguish non-isomorphic graphs by considering higher-order neighborhoods and interactions.Morris *et al.* (2020); Feng *et al.* (2024)

Previous approaches to retain as much structural information in graphs tend to be either computationally expensive or lacks structural information in their representation. Many attempts to simulate the Weisfeiler-Lehman isomorphism test in 2 dimensions Weisfeiler-Lehman(2WL)1 tend to follow the pattern. SIR-GN emulates the 1-dimensional WL test1 to retain the structure of graph networks. Identifying the Isomorphism between sub-graphs has shown to retain significantly more structural information than typical connectivity based methods. The ability to iterate over all the neighbors of every node to find near isomorphic, is far superior than looking at 1-2 hops out of a node. Damke *et al.* (2020)

Qian *et al.* (2022); Feng *et al.* (2024)

Recent work which introduced the N-WL hierarchy, generalizes the WL test by considering neighborhood relations up to a user-defined level N. This approach balances expressiveness with computational feasibility, offering better performance than the 1WL test while avoiding the exponential complexity associated with large k in k-WL tests. Similarly, the Neighborhood2-WL (N2-WL) approach, strikes a balance between expressiveness and computational demands. N2-WL is theoretically as powerful as the 3-WL test but requires significantly fewer computational resources,

Algorithm 2 2-Dimensional Weisfeiler-Lehman (WL) Algorithm

function WLAlgorithm(G)
    Initialize node labels $L^{(0)}$ for each node in G
    Initialize iteration counter t to 0
    repeat
        Initialize empty multiset $M^{(t)}$
        for each node v in G **do**
            $N(v) \leftarrow$ sorted list of neighbors of v in G
            $h(v) \leftarrow$ hash of $L^{(t)}[v]$ concatenated with the sorted labels of $N(v)$ and their pairwise labels
            Add $h(v)$ to $M^{(t)}$
        end for
        Increment t by 1
        for each node v in G **do**
            Compute the new label $L^{(t)}[v]$ using the multiset $M^{(t)}$ and the current label $L^{(t-1)}[v]$
        end for
    until no label changes or maximum number of iterations reached
    return Final node labels $L^{(t)}$
end function

making it more practical for large graphs.Wang *et al.* (2023); Feng *et al.* (2024)

Despite these advancements, higher-order WL methods still face scalability issues due to the exponential growth in complexity with increasing k. This has led researchers to explore more efficient algorithms and approximation methods to harness the benefits of higher-order tests without incurring prohibitive computational costs.

4.1.2 Subgraph-Based Methods

Subgraph GNNs (SGNNs) have emerged as a promising direction to enhance expressiveness while managing complexity. By decomposing graphs into smaller subgraphs, SGNNs can capture local structural information more effectively. Frasca *et al.* (2022) The expressiveness of SGNNs is influenced by their subgraph generation policies and aggregation methods, with studies suggesting that certain classes of SGNNs have expressive power bounded by the 3-WL test. Frasca *et al.* (2022)

Recent advancements include the Subgraph Weisfeiler-Lehman (SWL) framework, which extends the classic WL test to subgraphs. This allows for capturing both global and local structural features. Zhang *et al.* (2024) The SWL approach has demonstrated state-of-the-art results in graph classification tasks, particularly in domains where subgraph patterns are critical, such as chemistry.

Equivariant Subgraph Aggregation Networks (ESAN) further enhance expressiveness by considering symmetries at both the node and subgraph levels. Bevilacqua *et al.* (2022) By maintaining equivariance during aggregation, ESAN effectively captures structural nuances inherent in many real-world graph datasets. However, ensuring equivariance introduces additional computational complexity, and scaling these methods to very large graphs remains an area of active research.

While subgraph-based methods offer improved expressiveness, they introduce their

own computational overheads, particularly in generating and aggregating subgraphs efficiently. Practical applications often involve partitioning large graphs into smaller subgraphs and using parallel processing. However, these approaches can sometimes lead to the loss of global structural information, highlighting the need for further optimization.

4.2 Graph Partitioning Approaches and Structural Integrity

A step further into subgraphs we have Graph partitioning techniques, which are often employed to reduce the computational complexity of processing large-scale graphs. Particularly when dealing with higher-order methods that are computationally intensive. By dividing a large graph into smaller, more manageable subgraphs, it becomes feasible to process each partition independently, thereby reducing memory requirements and computational time. The benefit of graph partitioning in comparison to only using existing subgraphs is the fact that a well written partition algorithm can be run on any size or shape of a graph. However, this approach can come at the significant cost of losing important structural information, which may adversely affect the performance of graph neural networks.

Partitioning can disrupt the inherent structural properties of the original graph, such as connectivity patterns and global topology, which are crucial for tasks that rely on understanding the overall graph structure. The loss of connected edges can lead to incomplete information about node relationships, causing the model to miss critical interactions between nodes in different partitions. As a result, while partitioning makes computation more tractable, partitioning methods may adversely affect the

message-passing mechanisms central to GNNs, as nodes in different partitions may not effectively share information.

Therefore, while graph partitioning approaches are effective in reducing the complexity of higher-order methods, on any size graph, they must be applied carefully. Identifying a loss in message passing would be difficult to validate outside of a testing environment. Therefore, the benefits of computational efficiency with the need to preserve structural information still remains a significant challenge. Techniques that minimize edge cuts or incorporate strategies to retain critical structural features across partitions are areas of ongoing research aimed at mitigating these issues.

4.3 Architectural and Methodological Improvements

Beyond structural methods, advancements in GNN architectures and methodologies contribute significantly to addressing complexity issues. These improvements often focus on reducing computational demands while maintaining or enhancing model performance.

4.3.1 Feature Selection and Regularization

Effective feature selection mechanisms reduce the dimensionality of graph features by focusing on the most informative aspects and ignoring irrelevant or noisy data. Velickovic *et al.* (2018) This not only enhances model performance but also helps prevent overfitting, a common issue with complex models.

Building upon the Feature Selection Graph Neural Network (FSGNN) model, researchers have introduced architectural improvements emphasizing simplicity and efficiency. Liu *et al.* (2023) By employing feature selection mechanisms that assign

weights to different features based on their importance, models can focus on informative features while ignoring less relevant ones. This proposed shallow architecture, has been shown to achieve competitive performance compared to deeper models when combined with effective feature selection and regularization techniques. Regularization methods—including dropout, L_2 regularization, and soft-selection techniques—help prevent overfitting and manage complexity by constraining the model's capacity. This approach allows the model to reduce complexity without sacrificing performance. Xu *et al.* (2018); Barceló *et al.* (2020); Bai *et al.* (2021)

4.3.2 Inductive and Scalable Solutions

Inductive learning frameworks like GraphSAGE, provide scalable solutions by generating node embeddings for unseen nodes or entirely new subgraphs without retraining the entire model. By leveraging node feature aggregation at multiple hops, GraphSAGE effectively incorporates neighborhood information while maintaining computational efficiency. This makes it particularly adept at scaling to large graphs, which is crucial for applications with dynamically evolving data, such as social or biological networks. Hamilton *et al.* (2017)

Neural Message Passing for Quantum Chemistry (MPNN) represents a more specialized application of the message-passing paradigm, focusing on molecular property prediction. Gilmer *et al.* (2017) MPNNs pass messages within the local neighborhood of each node and aggregate these messages to predict molecular properties. Despite being bound by the expressivity limits of the 1WL test, MPNNs have made considerable strides in applying GNNs to domains where node attributes have physical meanings, like molecular structures. This highlights the potential for hybrid solutions that mix structural learning with domain-specific features.

4.4 Integrating Structural and Architectural Advances

Combining higher-order structural approaches with architectural improvements offers a comprehensive strategy to address complexity challenges. This integration aims to balance expressiveness and computational feasibility, tailoring models to fit the problem at hand.

4.4.1 Balancing Expressiveness and Scalability

Flexible frameworks like the (k, t)-FWL+ aim to retain high expressiveness while adjusting model complexity according to specific requirements. By varying parameters such as k (the order of the WL test) and t (the number of iterations), these models can be fine-tuned to balance the trade-off between expressiveness and computational efficiency. This adaptability is particularly crucial for tasks where real-time or large-scale processing is necessary, such as in social network analysis or recommendation systems.Feng *et al.* (2024)

The use of subgraph aggregation, as seen with SGNNs and ESAN, combined with higher-order interactions, creates opportunities for leveraging the best aspects of both global expressiveness and local precision. Each of these methods contributes valuable tools to the ongoing quest for expressive, efficient, and scalable graph representation techniques.

4.4.2 Hybrid Methods

Hybrid approaches that integrate feature selection with higher-order and subgraph techniques can achieve a balance between expressiveness and efficiency. For instance, incorporating feature selection mechanisms into subgraph-based models reduces com-

putational overhead while preserving the ability to capture complex structural patterns.

The PATCHY-SAN architecture is an example of a model that integrates multiple strategies to enhance performance. It employs a normalization procedure to select fixed-size neighborhoods, extracts and expands local patches into a matrix representation, and then applies convolutional layers. This approach allows the model to handle both discrete and continuous node and edge attributes efficiently, achieving higher expressive power without the computational burdens typically associated with higher-order GNNs.Niepert *et al.* (2016)

4.5 Discussion

The exploration of these methodologies highlights significant progress in enhancing the expressiveness and scalability of GNNs. Higher-order WL methods and subgraph-based approaches provide powerful tools for capturing complex graph structures. However, challenges in computational efficiency and scalability persist, particularly when applying these methods to large, real-world graphs.

One key finding is that while higher-order GNNs offer superior performance, their computational complexity remains a significant barrier. Graph partitioning and feature selection are effective strategies for scaling these models, but limitations exist regarding their adaptability to different graph structures. Many papers that implement a higher-order WL tests in combination with graph partitioning and feature selection have led to improved classification accuracy and scalability, carving a clear path for future research. Xu *et al.* (2019); Morris *et al.* (2020); Xu *et al.* (2018); Zhang *et al.* (2023)

Architectural improvements, such as shallow networks and regularization tech-

niques, demonstrate that complexity can be managed without sacrificing performance. Inductive learning methods like GraphSAGE provide scalable solutions, but there is still a need to enhance their expressiveness to match higher-order models.

4.6 Conclusion and Future Directions

Addressing the complexities associated with higher-order graph representation learning requires a multifaceted approach. While advancements in higher-order WL tests and subgraph methods enhance expressiveness, they often come at the cost of increased computational demands. Architectural improvements and feature selection mechanisms offer practical solutions to manage complexity.

Future research directions include:

- **Developing Scalable Higher-Order Methods:** Exploring efficient algorithms and approximations that retain the benefits of higher-order tests without prohibitive computational costs.
- **Optimizing Subgraph Techniques:** Enhancing subgraph generation and aggregation processes to improve scalability and efficiency, potentially through parallel processing or sampling strategies.
- **Integrating Domain Knowledge:** Incorporating domain-specific information to enhance model performance while potentially reducing complexity.
- **Hybrid Approaches:** Combining structural and architectural advancements to create models that balance expressiveness with computational feasibility.

In conclusion, advancing graph representation learning hinges on overcoming the challenges posed by complexity. By integrating higher-order structural insights with

architectural and methodological innovations, it is possible to develop models that are both expressive and scalable. The insights and methodologies discussed in this chapter contribute to a clearer understanding of the current landscape and highlight promising paths forward in the field of graph representation learning.

REFERENCES

Atwood, James, & Towsley, Don. 2016. Diffusion-convolutional neural networks. *Advances in neural information processing systems*, **29**.

Bader, David A, Meyerhenke, Henning, Sanders, Peter, & Wagner, Dorothea. 2013. *Graph partitioning and graph clustering*. Vol. 588. American Mathematical Society Providence, RI.

Bai, Jiandong, Zhu, Jiawei, Song, Yujiao, Zhao, Ling, Hou, Zhixiang, Du, Ronghua, & Li, Haifeng. 2021. A3t-gcn: Attention temporal graph convolutional network for traffic forecasting. *ISPRS International Journal of Geo-Information*, **10**(7), 485. Publisher: Multidisciplinary Digital Publishing Institute.

Barceló, P., Kostylev, E. V., Monet, M., Pérez, J., Reutter, J. L., & Silva, J. Pablo. 2020. The logical expressiveness of graph neural networks. *In: International Conference on Learning Representations*.

Berg, Rianne van den, Kipf, Thomas N, & Welling, Max. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*.

Bevilacqua, Beatrice, Frasca, Fabrizio, Lim, Derek, Srinivasan, Balasubramaniam, Cai, Chen, Balamurugan, Gopinath, Bronstein, Michael M., & Maron, Haggai. 2022 (Mar.). *Equivariant Subgraph Aggregation Networks*. arXiv:2110.02910 [cs].

Bodnar, Cristian, Frasca, Fabrizio, Wang, Yuguang, Otter, Nina, Montufar, Guido F, Lio, Pietro, & Bronstein, Michael. 2021. Weisfeiler and lehman go topological: Message passing simplicial networks. *Pages 1026–1037 of: International Conference on Machine Learning*. PMLR.

Corso, G., Cavalleri, L., Beaini, D., Liò, P., & Velickovic, P. 2020. Principal neighbourhood aggregation for graph nets. *CoRR*, **abs/2004.05718**.

Damke, Clemens, Melnikov, Vitalik, & Hüllermeier, Eyke. 2020. A novel higher-order Weisfeiler-Lehman graph convolution. *Pages 49–64 of: Asian Conference on Machine Learning*. PMLR.

de Haan, Pim, Cohen, Taco S, & Welling, Max. 2020. Natural Graph Networks. *Pages 3636–3646 of: Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc.

Defferrard, M., Bresson, X., & Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Pages 3844–3852 of: Advances in Neural Information Processing Systems*.

Feng, Jiarui, Kong, Lecheng, Liu, Hao, Tao, Dacheng, Li, Fuhai, Zhang, Muhan, & Chen, Yixin. 2024. Extending the design space of graph neural networks by rethinking folklore Weisfeiler-Lehman. *Advances in Neural Information Processing Systems*, **36**.

Frasca, Fabrizio, Bevilacqua, Beatrice, Bronstein, Michael, & Maron, Haggai. 2022. Understanding and extending subgraph gnns by rethinking their symmetries. *Advances in Neural Information Processing Systems*, **35**, 31376–31390.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. 2017. Neural message passing for quantum chemistry. *In: International Conference on Machine Learning.*

Grohe, M. 2020. *Word2vec, Node2vec, Graph2vec, X2vec: Towards a theory of vector embeddings of structured data.* Published: CoRR, abs/2003.12590.

Grover, Aditya, & Leskovec, Jure. 2016. node2vec: Scalable feature learning for networks. *Pages 855–864 of: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining.*

Hamilton, W. L., Ying, R., & Leskovec, J. 2017. Inductive representation learning on large graphs. *Pages 1025–1035 of: Advances in Neural Information Processing Systems.*

Kipf, Thomas N., & Welling, Max. 2017. Semi-supervised classification with graph convolutional networks. *In: International Conference on Learning Representation.*

Liu, Chenyi, Aggarwal, Vaneet, Lan, Tian, Geng, Nan, Yang, Yuan, & Xu, Mingwei. 2023. Machine Learning for Robust Network Design: A New Perspective. *IEEE Communications Magazine*, **61**(10), 86–92.

Maron, Haggai, Ben-Hamu, Heli, Serviansky, Hadar, & Lipman, Yaron. 2019. Provably Powerful Graph Networks. *In: Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, & Dean, Jeff. 2013. Distributed Representations of Words and Phrases and their Compositionality. *In:*

Advances in Neural Information Processing Systems, vol. 26. Curran Associates, Inc.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, Jan Eric, Rattan, G., & Grohe, M. 2019. Weisfeiler and Leman go neural: Higher-order graph neural networks. *Pages 4602–4609 of: AAAI Conference on Artificial Intelligence*.

Morris, Christopher, Rattan, Gaurav, & Mutzel, Petra. 2020. Weisfeiler and Leman go sparse: Towards scalable higher-order graph embeddings. *Advances in Neural Information Processing Systems*, **33**, 21824–21840.

Niepert, M., Ahmed, M., & Kutzkov, K. 2016. Learning convolutional neural networks for graphs. *Pages 2014–2023 of: International Conference on Machine Learning*.

Patwary, Md Anwarul kaium, Garg, Saurabh, & Kang, Byeong. 2019 (Feb.). *Window-based Streaming Graph Partitioning Algorithm*. arXiv:1902.01543.

Perozzi, Bryan, Al-Rfou, Rami, & Skiena, Steven. 2014. Deepwalk: Online learning of social representations. *Pages 701–710 of: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Qian, Chendi, Rattan, Gaurav, Geerts, Floris, Niepert, Mathias, & Morris, Christopher. 2022. Ordered subgraph aggregation networks. *Advances in Neural Information Processing Systems*, **35**, 21030–21045.

Shervashidze, N., Schweitzer, P., Leeuwen, E. J. van, Mehlhorn, K., & Borgwardt, K. M. 2011. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, **12**, 2539–2561.

Stanton, Isabelle, & Kliot, Gabriel. 2012. Streaming graph partitioning for large distributed graphs. *Pages 1222–1230 of: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining.*

Sutskever, Ilya, Vinyals, Oriol, & Le, Quoc V. 2014. Sequence to Sequence Learning with Neural Networks. *Pages 3104–3112 of: Advances in Neural Information Processing Systems*, vol. 27.

Tang, Jian, Qu, Meng, Wang, Mingzhe, Zhang, Ming, Yan, Jun, & Mei, Qiaozhu. 2015. Line: Large-scale information network embedding. *Pages 1067–1077 of: Proceedings of the 24th international conference on world wide web.*

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. 2018. Graph attention networks. *In: International Conference on Learning Representations.*

Veličković, Petar, Fedus, William, Hamilton, William L, Liò, Pietro, Bengio, Yoshua, & Hjelm, R Devon. 2018. Deep graph infomax. *arXiv preprint arXiv:1809.10341.*

Wang, Qing, Chen, Dillon Ze, Wijesinghe, Asiri, Li, Shouheng, & Farhan, Muhammad. 2023.

\

mathscr {N} $-WL: ANewHierarchyofExpressivityforGraphNeuralNetworks.In: TheEleve$

Xu, Da, Ruan, Chuanwei, Korpeoglu, Evren, Kumar, Sushant, & Achan, Kannan. 2020. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962.*

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K., & Jegelka, S. 2018. Representation learning on graphs with jumping knowledge networks. *Pages 5453–5462 of: International Conference on Machine Learning*.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. 2019. How powerful are graph neural networks? *In: International Conference on Learning Representations*.

Yan, Xu, Fan, Xiaoliang, Yang, Peizhen, Wu, Zonghan, Pan, Shirui, Chen, Longbiao, Zang, Yu, & Wang, Cheng. 2021. ConTIG: Continuous Representation Learning on Temporal Interaction Graphs. *arXiv preprint arXiv:2110.06088*.

You, Jiaxuan, Ying, Rex, & Leskovec, Jure. 2019. Position-aware Graph Neural Networks. *Pages 7134–7143 of: Proceedings of the 36th International Conference on Machine Learning*. PMLR. ISSN: 2640-3498.

Zhang, Bohang, Feng, Guhao, Du, Yiheng, He, Di, & Wang, Liwei. 2023. A complete expressiveness hierarchy for subgraph gnns via subgraph weisfeiler-lehman tests. *Pages 41019–41077 of: International Conference on Machine Learning*. PMLR.

Zhang, Bohang, Gai, Jingchu, Du, Yiheng, Ye, Qiwei, He, Di, & Wang, Liwei. 2024. Beyond weisfeiler-lehman: A quantitative framework for GNN expressiveness. *arXiv preprint arXiv:2401.08514*.

Zhang, Ziwei, Cui, Peng, Pei, Jian, Wang, Xiao, & Zhu, Wenwu. 2018. Timers: Error-bounded svd restart on dynamic networks. *In: Thirty-second AAAI conference on artificial intelligence*.

Zuo, Yuan, Liu, Guannan, Lin, Hao, Guo, Jia, Hu, Xiaoqian, & Wu, Junjie. 2018. Embedding temporal network via neighborhood formation. *Pages 2857–2866 of: Pro-*

ceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining.