PAPER NAME

M_Tech_Dissertation_Suranjan_Dey.pdf

AUTHOR

Suranjan Dey

WORD COUNT

11261 Words

CHARACTER COUNT

60750 Characters

PAGE COUNT

54 Pages

FILE SIZE

2.3MB

SUBMISSION DATE

Jun 1, 2025 1:33 PM GMT+5:30

REPORT DATE

Jun 1, 2025 1:34 PM GMT+5:30

15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 11% Internet database
- 10% Publications database
- Crossref database
- Crossref Posted Content database
- 0% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Cited material

0% words excluded by Custom Sections

Enhancing Expressive Power Of Graph Neural Networks Using Geometric Transformations

Suranjan Dey

Enhancing Expressive Power Of Graph Neural Networks Using Geometric Transformations

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology
in
Computer Science

by

Suranjan Dey
[Roll No: CS2330]

under the guidance of

Dr. Swagatam Das
Associate Professor
Electronics and Communication Sciences Unit

Indian Statistical Institute
Kolkata-700108, India

June 2025

To my parents and my guide

"Intelligence is the ability to adapt to change."

- Stephen Hawking

CERTIFICATE

This is to certify that the dissertation entitled **"Enhancing Expressive Power of Graph Neural Networks Using Geometric Transformations"** submitted by Suranjan Dey to the Indian Statistical Institute, Kolkata, in partial fulfillment of the requirements for the award of the degree of Master of Technology in Computer Science, is a bonafide record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

Dr. Swagatam Das
Associate Professor,
Electronics and Communication Sciences Unit,
Indian Statistical Institute,
Kolkata-700108, INDIA.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. Swagatam Das, for his invaluable guidance, continuous support, and encouragement throughout the course of this research. I am also thankful to the faculty members and staff of Indian Statistical Institute, Kolkata, for providing a stimulating academic environment and the necessary resources. I extend my sincere thanks to my peers and lab mates for the fruitful discussions and for creating a collaborative atmosphere that made the research journey more enjoyable. Special thanks to my family members for their unwavering support and patience. Their belief in me has been a constant source of strength. Finally, I acknowledge the use of open-source tools and frameworks that significantly aided my experiments and documentation.

Suranjan Dey
Indian Statistical Institute
Kolkata - 700108 , India.

Abstract

Graph Neural Networks (GNNs) are highly effective in many real-world tasks, such as molecular property prediction, modeling protein structures, analyzing user-item relationships, and making link predictions. What sets them apart is their ability to learn meaningful representations by capturing not just the features of individual nodes, but also the overall structure of the graph they belong to. This expressive strength allows GNNs to model complex relationships more accurately. In this work, we take a step further by introducing geometric transformations aimed at improving how GNNs handle spatial information. In particular, we focus on angular aggregation methods that maintain rotational consistency, helping the model deliver more stable and reliable predictions even when the input orientation changes.

Keywords: *Expressive Power, Equivariance, Weisfeiler-Lehman test, Graph Isomorphism, Graph Representation Learning.*

Contents

Certificate iv

Acknowledgments v

Abstract 1

List of Figures 4

List of Tables 5

1 Introduction 7
- 1.1 Problem Statement . . . 7
- 1.2 Brief overview on GNN . . . 7
- 1.3 Thesis Overview . . . 9

2 Preliminaries 10
- 2.1 Basics of Graph Isomorphism . . . 10
- 2.2 Equivariance: Concepts and application in GNNs . . . 12
- 2.3 Equivariance and Invariance in Physical and Molecular Systems . . . 13
  - 2.3.1 Physical Dynamics simulation . . . 13
  - 2.3.2 Molecules . . . 14

3 Expressive Power of GNNs 15
- 3.1 Necessity of GNNs expressive power . . . 15
- 3.2 Expressive Power: Definition and Representation . . . 16
- 3.3 Design Choices and Their Impact on GNN Expressive Power . . . 17

4 Related Work And Motivation 19
- 4.1 Related Work . . . 19
- 4.2 Motivation . . . 20
- 4.3 Our Contribution . . . 20

5 Proposed GNN Model Architecture and Its Properties 22
5.1 Overview of the Proposed Model – Version 1 22
5.2 Analysis of the Proposed Model - Version 1 24
5.2.1 Local Rotation Equivariance 24
5.2.2 Permutation Invariance Over Neighbourhood 25
5.2.3 Scale Invariance and Equivariance 25
5.3 Overview of the Proposed Model – Version 2 25
5.4 Mathematical Formulation of The Version 2 Architecture 26
5.5 Analysis of the Proposed Model - Version 2 28

6 Experimental Results and Dataset Overview 30
6.1 Citation Networks Dataset 30
6.1.1 Dataset Details 30
6.1.2 Experimental Setup 31
6.1.3 Results and Visualizations 32
6.2 Co-Purchase Networks Dataset 34
6.2.1 Dataset Details 34
6.2.2 Experimental Setup 34
6.2.3 Results and Visualizations 35
6.3 Molecular Datasets 38
6.3.1 Dataset Details 38
6.3.2 Experimental Setup 40
6.3.3 Quantitative Results 41

7 Conclusion and Future Work 44
7.1 Conclusion 44
7.2 Scope for Future Work 45

Bibliography 46

List of Figures

1.1 Illustration of the message passing mechanism in GNNs 8

2.1 The aggregation and the update process of WL test 12
2.2 Some of the non-isomorphic graphs where WL test failed 13

3.1 Illustration of GNNs Expressive Power 17
3.2 Graphs that mean and max aggregators fail to distinguish 18

6.1 TSNE Plots of Pubmed Dataset 32
6.2 TSNE Plots of Amazon Photo Dataset 35
6.3 TSNE Plots of Amazon Computer Dataset 36

List of Tables

1 List of Notations Used in Thesis 6

2.1 List of datasets used in the evaluation of equivariant GNNs 14

6.1 Details of the Planetoid dataset 31
6.2 Hyperparameter settings for Experiments on Planetoid Dataset . . . 31
6.3 Comparison of Baseline Results on Citation Network Datasets 33
6.4 Details of the Amazon Co-Purchase Dataset 34
6.5 Comparison of Baseline Results on Co-Purchase Network Datasets . . 37
6.6 Open Graph Benchmark Dataset Details 39
6.7 QM9 Dataset Statistics 39
6.8 Hyperparameter Settings for Experiments on OGB Datasets 40
6.9 Hyperparameter settings for Experiments on QM9 Datasets 40
6.10 Hyperparameter settings for experiments on the ZINC dataset 41
6.11 Performance comparison on OGB datasets 41
6.12 QM9 Molecular Property Prediction Results 42
6.13 Performance comparison on Zinc datasets 43

Table 1: List of Notations Used in Thesis

V	Set of vertices
E	Set of edges
h	Embedding vector
$\mathbb{R}$	Set of real numbers
$\mathcal{N}$	Set of neighbouring nodes of a particular vertex
ϕ	Functions that represent Multi-layer perceptrons and GNNs
A	Adjacency Matrix
X	Feature Matrix
C	Color label of a vertex used in WL test of isomorphism
Q	Set of orthogonal matrices
π	Set of permutations
$\mathcal{X}$	Original space of graph nodes
$\mathcal{Y}$	Embedding space
1	Feature matrix with all ones
Θ	Angular Distance Matrix

Chapter 1

Introduction

1.1 Problem Statement

Learning from graph-structured data—such as molecular networks, social networks, and financial systems—requires effective representations that capture the underlying structural information. Graph Neural Networks (GNNs) have emerged as a powerful framework for representation learning on graphs. These models operate through iterative message-passing and aggregation mechanisms, transforming node feature vectors by incorporating information from their multi-hop neighborhoods.

GNNs with high expressive power and representation learning ability can effectively capture subtle structural differences across diverse graph instances.

Several popular GNN variants have been proposed, each differing in their aggregation and pooling strategies, including Graph Convolutional Networks (GCN) [2], Graph Attention Networks (GAT) [3], and GraphSAGE [4].

The primary objective of this work is to enhance the expressive power of GNNs through a novel aggregation mechanism. The proposed GNN model aims to be highly expressive and demonstrate strong performance across a range of domains.

Additionally, we emphasize the importance of equivariance to structural transformations. In many real-world scenarios—such as molecular datasets—graphs undergo transformations, and ensuring equivariance allows the model to maintain consistent predictions. This not only improves reliability but also enhances the generalization capability of the model across structurally varying inputs.

1.2 Brief overview on GNN

Graphs are considered a highly useful data structure due to their ability to model a wide variety of real-world datasets. Well-known examples include social networks, citation networks, and molecular structures, all these datasets can be represented as graphs. These structures consist of nodes as entities, and the edges represent the

relationships between pairs of entities.

Traditional machine learning models like RNNs and CNNs are designed for data with regular structures, such as sequences or grids, and often struggle to learn from the irregular and complex topology of graphs.

Graph Neural Networks (GNNs) are permutation-equivariant neural networks that operate directly on graph-structured data. GNNs use a message-passing mechanism to compute and update node embeddings. In each iteration, features from single or multi-hop neighboring nodes are aggregated to compute updated node representations. These representations are then used for downstream tasks like classification or regression. Unlike CNNs or RNNs, which typically operate with fixed-size neighborhoods, GNNs are capable of aggregating information from variable-sized neighborhoods.

Given a graph $G = (V, E)$ with nodes $v_i \in V$ and edges $e_{ij} \in E$, we define a graph convolutional layer, following the notation in [1], as:

$$m_{ij} = \phi_e(h_i^l, h_j^l, a_{ij}). \tag{1.1}$$

$$m_i = \sum_{j \in \mathcal{N}(i)} m_{ij}. \tag{1.2}$$

.

$$h_i^{l+1} = \phi_h(h_i^l, m_i). \tag{1.3}$$

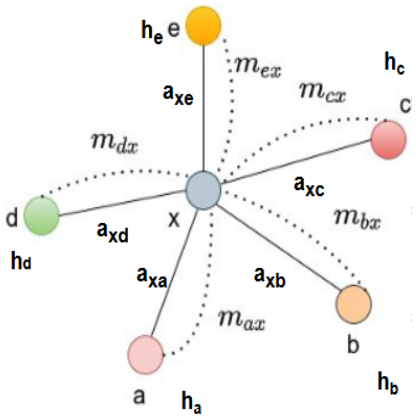


Figure 1.1: Illustration of the message passing mechanism in GNNs

1.3 Thesis Overview

The rest of the sections of this thesis are organized as follows.

Chapter 2: In this section, we will introduce some of the preliminary concepts on graph isomorphism and equivariance properties of graphs.

Chapter 3: This chapter provides a unifying perspective on the GNNs expressive power, highlighting its necessity, representation, and its strength.

Chapter 4: We will list down and analyze the design in some of the existing works that were used to improve the expressive power of GNNs.

Chapter 5: We will discuss about our novel approach and will provide the mathematical formulation of the entire architecture of the model.

Chapter 6: This chapter presents the experiments conducted on both small and large-scale real-world datasets, along with the corresponding results. Additionally, we provide a brief overview of the datasets utilized in our study to offer context for the experimental analysis.

Chapter 7: At the end, we will summarize our work and discuss the future scope of improvements.

Chapter 2

Preliminaries

2.1 Basics of Graph Isomorphism

Two graphs G = (A, X) and $G' = (A', X')$ where A and A' are the corresponding adjacency matrices and X and X' are the respective feature matrices, are considered to be isomorphic if there exists a bijective mapping π between the two graphs:

$$\pi : V[G] \rightarrow V[G']. \tag{2.1}$$

which satisfy $A_{uv} = A'_{\pi(u)\pi(v)}$ and $X_v = X'_{\pi(v)}$, while graph isomorphism reflects the topological equivalence of 2 graphs, detecting this equivalence is challenging and is considered an NP-hard problem. The WL test, known as the **Weisfeiler-Lehman test** [5],[6], provides us a fast and effective way to solve the problem using the color refinement algorithm. The algorithm uses a coloring technique to assign labels to every node in each iteration.

The iterative process has been illustrated in Figure 2.1 [7]. Initially, all the nodes in the graph were assigned the same color label. In every iteration, the labels of the center node and the neighbors are combined to get a set of multiple colors, which are further mapped to unique color labels using a hash function. This new, unique color becomes the new label of the central node. The process is repeated until there is no change in the distribution of the colors in the graph. A color histogram is used in every iteration to see the distribution of colors across all the nodes in the graph. This process can be mathematically formulated as:

$$C_v^{(l)} = Hash(C_v^{(l-1)}, \{C_u^{l-1} | u \epsilon N(v)\}). \tag{2.2}$$

To compare the structural similarity of two graphs, a systematic relabeling process is employed. The procedure iteratively refines node labels to capture their neighborhood context with increasing depth, as described below.

Algorithm 1 One iteration of the 1-WL Graph Isomorphism Test

Multiset-label determination

For l=0,set $M^l(v) = c^0(v) = c(v)$.

For $l > 0$, assign a multiset -label $M^l(v)$ to each node v in G and G' which consists of the multiset $\{c^{l-1}(u) \mid u \in N(v)\}$

Sorting each multiset

Sort the elements in $M^l(v)$ in ascending order and concatenate them into a string $s^l(v)$.

Add $c^{l-1}(v)$ as a prefix to s^l(v) and call the resulting string $s^l(v)$.

Label compression

Sort all of the strings $s^l(v)$ for all v from G and G' in ascending order.

Map each string $s^l(v)$ to a new compressed label, using a function f: $\sum_* \rightarrow \sum$ such that $Hash(s^l(v)) = Hash(s^l(w))$ iff $s^l(v) = s^l(w)$

Relabeling

set $c^l(v) = Hash(s^l(v))$ for all nodes in G and G'.

While the identical coloring results can indicate, but they do not guarantee that the graphs are isomorphic. The WL test has limitations when it comes to distinguishing members of certain regular non-isomorphic graphs. Figure 2.2 shows some of the non-isomorphic graphs that are mistakenly identified as isomorphic by the algorithm.

There exists a study on the expressive power of GNN by Xu et al.[8] where they have characterized the maximum representational capacity of general class of GNNs to be up to 1-WL test, that is GNNs are as powerful as the 1-WL test in their ability to distinguish graphs. Now, a maximally powerful GNN can distinguish different graph structures by mapping them to different representations in the embedding space. Since this property of mapping the two different graphs to separate embeddings involves solving the graph isomorphism problem, the expressive power of this class of GNNs is theoretically upper-bounded by the discriminative capability of the Weisfeiler-Lehman (WL) test for graph isomorphism.

In our context, it is important to note that any GNN model that maps two graphs to different embeddings must at least be as powerful as the WL test. In other words, if a GNN is able to produce distinct embeddings for two graphs, then the WL test must also be capable of distinguishing them. This places a theoretical upper bound on the expressive power of message-passing GNNs, aligning their behavior closely with that of the WL test.

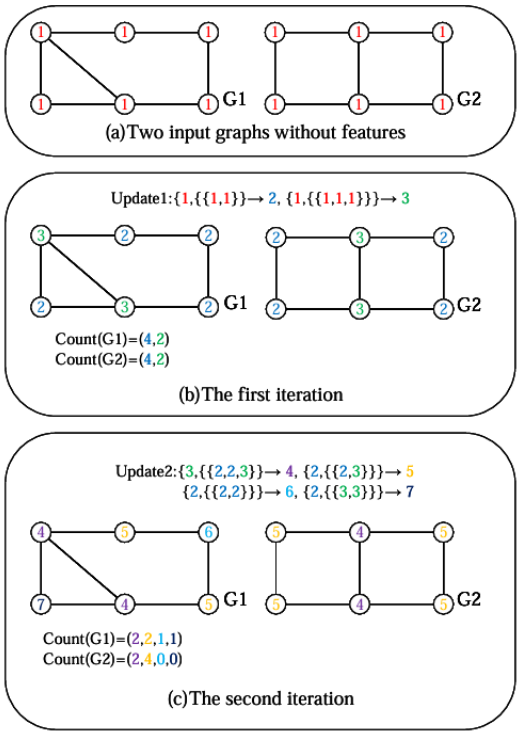


Figure 2.1: The aggregation and the update process of WL test

2.2 Equivariance: Concepts and application in GNNs

Let $g \epsilon G$ be an abstract group and T_g be a set of transformations on X, denoted as $T_g : X \rightarrow X$. A function $\phi : X \rightarrow Y$, is considered to be equivariant to g if there exists a similar transformation in the output space $S_g : Y \rightarrow Y$, such that:

$$\phi(T_g(x)) = S_g(\phi(x)). \tag{2.3}$$

Some of the most common equivariance that we encounter while working with the GNN models are as follows:

- **Translation Equivariance.** suppose x is an input set of M point clouds i.e., X=$(x_1, x_2, ...x_M) \epsilon R^{M*n}$, and ϕ be a non-linear funcion. Let T_g be a translation on the input set by g that is $T_g = x + g$. Then ϕ is said to exhibit translation equivariance iff $\phi(x) + g = \phi(x + g)$.

$$g + g = \phi(x + g). \tag{2.4}$$

- **Rotation (and Reflection) Equivariance.** For any orthogonal matrix $Q \epsilon R^{n*n}$, let Qx denote $(Qx_1, Qx_2, ..., Qx_M)$. Then rotating the input using Q will result in equivalent rotation in the output. So $Qy = \phi(Qx)$.

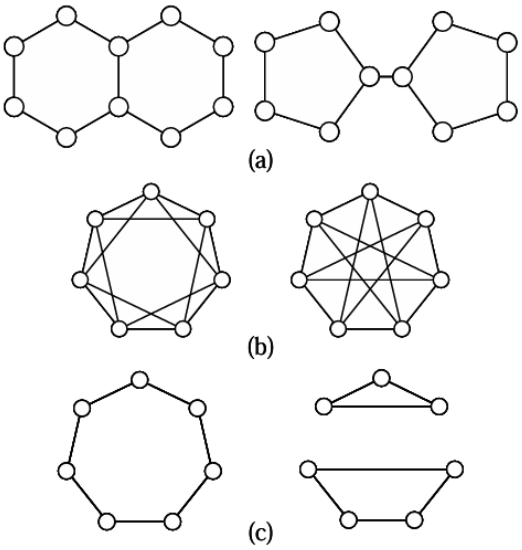


Figure 2.2: Some of the non-isomorphic graphs where WL test failed

- **Permutation Equivariance.** Permutation equivariance in general states that the output order of the function should correspond to the input order. The function ϕ is said to be permutation equivariant if for all possible permutations denoted by π it satisfies $\phi(\pi(x)) = \pi(y)$.

  While designing GNN models, we generally focus on generating permutation equivariant functions for graphs by applying a local permutation invariant aggregation function on the neighboring features of each node.

Functions that satisfy properties 1 & 2, i.e., (translation, rotation, and reflection equivariance), are said to exhibit E(3) equivariance.

Equivariant GNNs have a wide range of applications on various real-world data that exhibits geometry like physical systems, point clouds, and molecular datasets. An overview regarding these datasets can be found in Table 2.1.[9]

2.3 Equivariance and Invariance in Physical and Molecular Systems

2.3.1 Physical Dynamics simulation

Physical systems involve objects like the charged particles whose interactions follow the physical laws. A N-body simulation consists of multiple charged bodies driven by the Coulomb force. The task in this case is to predict the dynamics of the particles

Table 2.1: List of datasets used in the evaluation of equivariant GNNs

Dataset	Application	Task	Property
N-body	Physical Simulation	Position & Velocity Prediction	Equivariant
Constrained N-body	Physical Simulation	Position Prediction	Equivariant
Motion Capture	Physical Simulation	Position Prediction	Equivariant
QM9	Small Molecule	Chemical Property Prediction	Invariant
MD17	Small Molecule	Energy & Force Prediction	Invariant
ISO17	Small Molecule	Energy & Force Prediction	Invariant
OC20	Molecule (Catalyst)	Relaxed Energy Prediction	Invariant
GEOM	Small Molecule	Structure Generation	Equivariant

given the initial coordinates, including the position, velocities, and charges. As the dynamics of the particles translate, rotate, and reflect together along with the entire system, this makes the task of predicting the dynamics E(3) equivariant.

2.3.2 Molecules

Molecular data is a key area where equivariant models find significant application. In such datasets, molecules are represented as graphs where atoms serve as nodes, often characterized by attributes like atomic number. The connections between atoms—whether they represent true chemical bonds or are defined by spatial proximity within a specific cutoff distance—form the edges of these graphs. These molecular graphs exhibit a range of chemical properties, such as energy levels or reactivity, which are typically predicted using regression techniques.

A notable characteristic of these properties is their invariance to spatial transformations—shifting, rotating, or reflecting the entire molecule doesn't alter its intrinsic chemical behavior. Because of this, models that are invariant under Euclidean group transformations (E(3)) are naturally well-matched to the task. They can learn more effectively by aligning with the symmetries of the domain, often resulting in better generalization and reduced data requirements.

Chapter 3

Expressive Power of GNNs

3.1 Necessity of GNNs expressive power

All machine learning problems are more or less based on learning an approximation f_θ of a function f that maps a dataset from an input space X to an output space Y. Here *theta* is the parameter involved in the abstraction that we try to optimize. Now, since the function f is typically unknown to us a priori, we would expect f_θ to approximate a wide range of functions so that it can cover f* as well. An estimate of this range of functions being approximated by f_θ is known to be its expressive power. The expressive power of GNNs refers to their ability to distinguish between two different graph structures, to be able to identify structurally different nodes and edges. Also, it denotes the ability of the model to capture the complex relationships existing in the graph data.

The expressive power of GNN is not limited only to its feature embedding ability, GNNs that cannot maintain the graph topology often show inferior performance as compared to MLP. The topology representation ability of GNNs is often considered to be the key to their superior performance. Deep graph neural networks having low expressive power often suffer from the problem of over-smoothing, as a result, the node representations become indistinguishable from one another across the layers. Also, poor expressive power restricts the model's ability to capture the long-range dependencies that are considered to be critical in domains like protein folding and fraud detection.
As mentioned in the earlier sections, one of the most common applications of GNNs in the field of chemistry involves the property prediction of molecules. Such tasks involve identifying similar graph structures, detecting subgraphs, and mapping isomorphic graphs to similar embeddings in the latent space, while ensuring that non-isomorphic pairs are mapped to distinct embeddings. In order to achieve this level of discrimination in graph structures, GNN models should have a high degree of expressive power.

Efforts towards the improvement of the expressive power have led to models with enhanced architecture, like the use of higher-order message passing scheme (K-GNNs) [10], leveraging the use of subgraph-based models, and positional encoding in the models. Equivariant neural networks in physical systems are also crucial for achieving high expressive power, as they inherently respect the underlying symmetries of the system (e.g., translation, rotation, reflection), allowing the model to generalize more effectively while reducing the need for redundant learning.

Without sufficient expressive power, GNNs can become indistinguishable function approximators, which can further limit their applications in structure-sensitive tasks.

3.2 Expressive Power: Definition and Representation

Any graphical data will involve information related to node features and graph topology. Since representation learning is a crucial element of GNN models, their feature embedding ability is unquestionable. As a result, the expressive power of these models is more reflected by their topology representation ability. Any GNN model can be considered to be an approximation of a nonlinear function $\mathbf{f}$ that maps data from the input domain $\mathcal{X}$ to the embedding space $\mathcal{Y}$. We get the node embedding h_v using $\mathbf{f}$ as $h_v = f(A, x_v)$, where A represents the graph adjacency matrix and x_v is the input node feature. For any two nodes v1 and v2 with different topological positions, if $x_1 \neq x_2$, then due to the high feature embedding ability we will get $\mathbf{f}(A, x_1) \neq \mathbf{f}(A, x_2)$, and if $x_1 = x_2$, still we get $\mathbf{f}(A, x_1) \neq \mathbf{f}(A, x_2)$, by the topology representation ability of the GNNs.

Let $\mathcal{F}$ represent the domain space of the feature embedding ability of the GNNs. That is, if $\mathbf{X}$ is random (input graph with some random node feature) then $\mathcal{F} = \{f(A, X)|X \text{ is random}\}$.
Similarly, let $X = 1$ (input graph without any node features), then the value domain $\mathcal{F}' = \{f(A, X)|X = 1\}$, signifies the topology representation ability. Then given X is random the size of the intersection of both the domains gives us the expressive power of GNNs, i.e., $\mathcal{F} \cap \mathcal{F}'$ represents the expressive power of $\mathbf{f}$. This is illustrated in Figure 3.1 [11]

Some of the existing works on expressive power characterize it in terms of approximation, separation, and subgraph counting ability. Approximation ability looks at GNN's ability to approximate functions on graphs, while the separation ability focuses on GNN's ability to identify isomorphic graphs. The other perspective, which is the subgraph counting ability, is based on subgraph topology, like community identification in real-world datasets (molecular functional groups, etc), hence it holds practical

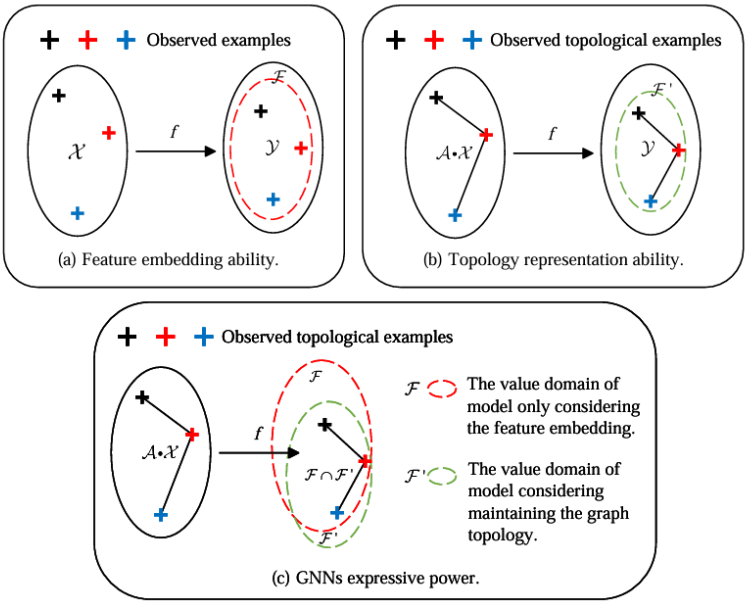


Figure 3.1: Illustration of GNNs Expressive Power

application value, unlike the other two perspectives, which have limitations when it comes to examining the real-world problems.

3.3 Design Choices and Their Impact on GNN Expressive Power

The neighbourhood aggregation process in GNNs is strikingly similar to the neighbour label aggregation in the WL test [8]. While all the message-passing GNNs share the same framework, subtle architecture differences can contribute to their ability to model complex graph structures, as a result, different GNNs with the same basic framework can show differences in their expressive powers.

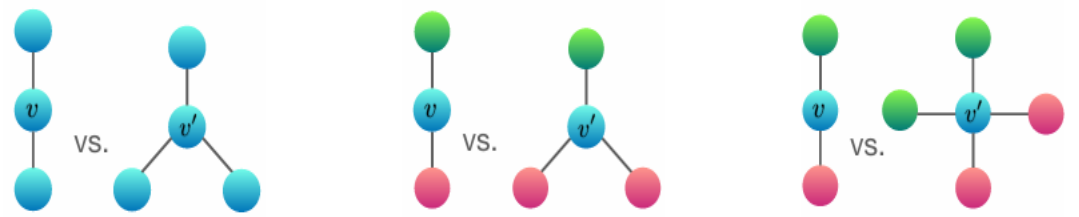
Some of the advanced GNNs go beyond the 1-WL test limit by using higher-order interactions like the K-GNNs [10]. Such models are more powerful in terms of expressive power and can even distinguish between structures that are indistinguishable by all 1-WL equivalent GNN models. But to possess such strength, this model requires more computing power and is less scalable compared to their 1-WL-equivalent counterparts.

The Aggregation function used in the model is one of the critical factors in determining the expressive strength. Xu et al.[8] showed that the mean and max aggregator functions lose information about the exact structure of the multiset of neighbours.

They used the sum aggregator function to design their Graph Isomorphism Network Model (GIN) that can preserve the full multiset structure and is also provably more powerful than the other 2 models.

Another factor is the post-aggregation transformation, which is typically implemented using MLPs. Stronger GNNs have deeper and more expressive MLPs that allow them to capture higher-order dependencies. Shallow layers and linear transformation, on the other hand, can restrict the model's representational capacity even if the aggregation is strong.

The strength of GNNs' expressive power is affected by the model's design choice, the aggregation and transformation functions are the crucial elements that shape its ability to encode the structural differences. Designing a GNN model with strong expressive capabilities is essential when working with complex graphs where minor structural differences can affect the learning objective.



(a) Mean and Max both fail (b) Max fails (c) Mean and Max both fail

Figure 3.2: Graphs that mean and max aggregators fail to distinguish

The above Figure from [8] shows the shortcomings of the max and mean aggregation functions. The node colors denote the distinct node features. Our assumption is that GNNs aggregate the neighbour features first before combining with the central node labelled as v and v'. The mean will fail to distinguish when the distribution of features is same across the neighbours. Max, on the other hand, neither captures the exact structure nor the distribution, it only identifies the representative element from every set of nodes sharing similar features.

Chapter 4

Related Work And Motivation

4.1 Related Work

There have been several works in the past towards the enhancement of the expressive power of GNNs. We will mention some of the important and notable ones in the underlying section.

The theoretical foundation for measuring the expressive power was laid by the Paper of Xu et al.[8]. This paper established the relation between the GNNs and the 1-Weisfeiler-Lehman (1-WL) isomorphism test, where they showed that common GNNs are only as powerful as the 1-WL test. Further, they proposed **Graph Isomorphism Network (GIN)** and demonstrated that it can achieve a maximum discriminative power of 1-WL test by using an injective aggregation function. This work has laid the theoretical foundation for improving the expressive capacity of the GNNs. To overcome the limitation of 1-WL, Morris et al. [12] introduced a generalized architecture, called **k-dimensional GNN** or (k-GNNs), which takes into account higher-order graph structures at multiple scales. This provided a non-isomorphic graph distinguishing ability beyond the capability of the traditional GNNs.

The **"E(n) Equivariant Graph Neural Networks"** paper by Garcia Satorras et al.[13] proposed a framework that is equivariant to rotations, reflections, translations, and permutations. This architecture was used by them to model physical systems (e.g., molecules, particles), and their design involved a message passing framework that is inherently equivariant and can easily scale the equivariance to more than 3-dimensional spaces. This approach improves the expressive power of GNNs in learning spatially aware representations.
Another interesting work towards advancement beyond 1-WL expressiveness was done by Yang et al.[14]. This paper tried to overcome the limitations of conventional GNNs, that is, their inability to perform injective aggregations. To overcome this bottleneck, they introduced **ExpandingConv**, an aggregation framework that uses learnable ag-

gregation coefficient matrices and applies non-linear transformations before aggregation. Their approach has shown good results for large and densely connected graphs. This framework can capture richer structural information beyond immediate neighbors.

4.2 Motivation

Traditional GNNs that are based on message passing mechanism focus on the topological structure of the data. Most of the methods have neglected the geometric and spatial context that is present in the real-world graph datasets.

In applications involving molecular graphs, node positions can define meaningful relationships. While some of the GNNs have used positional encoding but there is a gap in capturing the directional and angular dependencies among the nodes in the graph. Capturing the angular relationship among the nodes can further improve the expressiveness of the GNNs. Some of the datasets, like Citation and product Co-purchase networks, have non-uniform topological orientations. While traditional GNNs treat the neighbours symmetrically depending on the edges between the nodes, none of the models focus on feature alignment.

To address these limitations, we will introduce a novel GNN architecture that will use the angular aggregation mechanism. The design will consider the angular distance between the node features and the spatial orientation of nodes in the feature space. The learned representations will be used to update both the feature vectors and node positions. This feedback mechanism between the feature and geometry will lead to the computation of richer representations and consequently will improve the expressive power of the model. The proposed approach will encode both feature similarity and the geometric structure of the graph.

4.3 Our Contribution

Our contributions are summarized as follows:

- We have proposed two variants of our novel GNN model that are scale-invariant and adapt to rotations in feature space through angular transformation.
- Our first model uses angular distance-based aggregation and a message passing mechanism, and multi-layered rotation-based feature transformations.
- We have optimized the run-time using sparse tensors to process large graph datasets, and the model also supports multiple pooling mechanisms that are

controlled through parameters to enhance its adaptability to multiple downstream tasks like (regression, classification, etc)

- We have run experiments using this model on multiple medium and large-scale citation networks and co-purchase network datasets.
- We have tried to enhance the performance of this model by proposing a second variant that uses the spatial relationships between nodes. This version iteratively updates node coordinates using both feature-based and spatial angular distances, and establishes a feedback loop between the evolving node features and their spatial positions.
- We have run experiments on multiple molecular datasets like QM9, Open Graph Benchmark Datasets (molhiv), and Zinc datasets. The results have been summarized in the upcoming sections.

Chapter 5

Proposed GNN Model Architecture and Its Properties

In this section, we will provide a structural overview of our novel GNN model and will look into the key properties of the proposed model. This shall include mathematical formulations of both versions of the model, with necessary explanations.

5.1 Overview of the Proposed Model – Version 1

Here is the mathematical formulation of our version 1 model architecture:

- Given the initial node feature X is an $n * d_{in}$ feature matrix, we first project the feature vectors to the hidden dimension space

$$H^{(0)} = \text{ReLU}(XW_0 + b_0), \tag{5.1}$$

where W_0 is a real matrix of size $d_{\text{in}} \times d_{\text{hid}}$, and b_0 is a real vector of dimension d_{hid}. Layer Normalization was used for faster convergence, followed by Dropout:

$$H^{(0)} = \text{Dropout}(\text{LayerNorm}(H^{(0)})). \tag{5.2}$$

- For each layer $l = 1$ to L, let h_i be the feature vector for node i, which is a real vector of dimension d_{hid}. We first normalize it:

$$\hat{h}_i = \frac{h_i^l}{||h_i^l||_2}, \text{ for all i=1.....n} \tag{5.3}$$

- We define the **Angular Distance Matrix**, where Θ is a real matrix of size $n \times n$.

$$\Theta_{ij} =< \hat{h}_i, \hat{h}_j > . \tag{5.4}$$

- We define the symmetrically normalized adjacency matrix $\tilde{A}$ as a real matrix of size $n \times n$, given by:
$$\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}. \tag{5.5}$$

- We aggregate the angular distances from the neighborhood for each node as follows:
$$\alpha_i = [\tilde{A}\Theta^T]_{ii}. \tag{5.6}$$

  The aggregated angle for each node is:
$$\theta_{agg,i} = \frac{1}{deg(i)} \sum_{j \epsilon \mathcal{N}(i)} \theta_{ij}. \tag{5.7}$$

- The aggregated angle is used to rotate the feature vector in the latent space, thereby updating the representation of the vector, which is useful for the downstream tasks:
$$H^{l+1} = \mathcal{R}_a H^{(l)}. \tag{5.8}$$

- After L angular rotations, we pass the updated feature matrix through multilayer perceptrons (MLPs) involving non-linear activation, and normalization for the downstream tasks
$$Z = \mathcal{F}_{MLP}(H^L). \tag{5.9}$$

The use of a symmetrically normalized adjacency matrix $\tilde{A}$ during the aggregation process facilitates stable and unbiased feature propagation. This method was used by Kipf et al.[2] in their paper **"Semi-Supervised Classification with Graph Convolutional Networks"**[2]. Simply using the adjacency matrix would lead to the dominance of nodes with a high degree during the aggregation process. The normalization scales the contribution inversely with the node degrees.

$$Message : \frac{1}{\sqrt{d_i d_j}} A_{ij} \theta_j, \text{ where j} \epsilon \mathcal{N}(i) \tag{5.10}$$

5.2 Analysis of the Proposed Model - Version 1

Let us now examine a few key characteristics that emerge from the behavior of this model. These properties provide insight into its functioning.

5.2.1 Local Rotation Equivariance

The model exhibits local rotation equivariance on the basis of the node wise rotation matrix, which is different across all the nodes. All feature vectors are rotated consistently with their local angular context, which preserves the geometric alignment in the latent space.

Let $\mathcal{R}$ denote the rotation matrix, and we have a learned rotation angle θ_i from the neighbours. A feature vector h_i. Then we can define a node-wise function f_i as:

$$f_i(h_i) = \mathcal{R}(\theta_i). \tag{5.11}$$

Now, suppose we first rotate the input vector by some angle ϕ_i,i.e.,

$$\mathcal{T}(h_i) = \mathcal{R}(\phi_i)h_i. \tag{5.12}$$

Then we apply f_i:

$$f_i(\mathcal{T}(h_i)) = \mathcal{R}(\theta_i)\mathcal{R}(\phi_i)h_i. \tag{5.13}$$

Now suppose we apply T after rotation:

$$\mathcal{T}'(f_i(h_i)) = \mathcal{R}(\phi_i)\mathcal{R}(\theta_i)h_i. \tag{5.14}$$

Now, when we combine two rotations, the resulting transformation is also a rotation, and the angle of the resulting rotation is the sum of the individual angles:

$$\begin{aligned} \mathcal{R}(\theta_i)\mathcal{R}(\phi_i) &= \mathcal{R}(\theta_i + \phi_i) \\ &= \mathcal{R}(\phi_i + \theta_i) \\ &= \mathcal{R}(\phi_i)\mathcal{R}(\theta_i). \end{aligned} \tag{5.15}$$

So we get:

$$f_i(\mathcal{T}(h_i)) = \mathcal{T}'(f_i(h_i)). \tag{5.16}$$

5.2.2 Permutation Invariance Over Neighbourhood

The model is permutation invariant and is unaffected by node neighbourhood ordering. We use the symmetrically normalized adjacency matrix $\hat{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ to aggregate angles from neighbouring nodes. Since summation is order-independent, reordering the neighbours does not affect the output, ensuring invariance under neighborhood permutations.

$$\theta^{l+1}_{agg,i} = \sum_{j \epsilon \mathcal{N}(i)} \hat{A}_{ij}\theta^{l}_{ij}. \tag{5.17}$$

5.2.3 Scale Invariance and Equivariance

The angular distance computation mechanism in our model is scale invariant, since the distance is unaffected under scalar multiplications:

$$\theta_{ij} = Cos^{-1}\frac{ch^l_i \cdot ch^l_j}{\|ch^l_i\|\|ch^l_j\|} = Cos^{-1}\frac{h^l_i \cdot h^l_j}{\|h^l_i\|\|h^l_j\|}. \tag{5.18}$$

The rotation step satisfies scale equivariance:

$$\mathcal{R}(\theta_i)(c.h_i) = c.\mathcal{R}(\theta_i)(h_i). \tag{5.19}$$

So the proposed model shows some desirable geometric properties that make it suitable for learning over spatially structured data like graphs. Together, these properties contribute to a stable and expressive framework for encoding geometry-aware features in graph-based learning.

5.3 Overview of the Proposed Model – Version 2

The newly proposed version 2 model is an extension of the version 1 architecture that uses n-dimensional coordinates for every node in the process of representation learning.
Molecular datasets show various properties, like binding affinity, which involves the 3-D geometry of the molecules apart from the graph connectivity. Certain datasets, like the QM9, MoleculeNet, exhibit properties like energy, dipole moments that depend on the interatomic distance and angles.

As a result, the models that are able to capture the 3-D structural information are supposed to show good results on these datasets.

There are many existing models that leverage the spatial information from the coordinates very well and show robustness to transformations like rotation, translation, and also exhibit transformation invariance/equivariance. Such models have also

outperformed Vanilla GNNs on multiple occasions. Some of the well-known models in this category are **DimeNet**[15],**EGNN**[13], etc. Leveraging the information provided by the coordinates with the angles will add some physical intuition that will improve the physical inductive bias and generalization.

GNNs that use coordinates really shine when we are dealing with problems like energy calculations, force interactions, or simulating how molecules move. In these kinds of tasks, it's important that the model understands the geometry of the molecule and respects how it would behave in physical space, like staying consistent even if you rotate or shift it. By building this spatial understanding directly into the network, these models tend to be more accurate and are better at handling new molecular shapes they haven't seen before.

5.4 Mathematical Formulation of The Version 2 Architecture

After looking at the above-mentioned benefits of GNNs that leverage coordinate information, we decided to extend our version 1 model. Below, we have mentioned the mathematical formulation of the architecture of our version 2 model in detail.

- Each node feature $h_i^{(0)}$ is linearly projected ,and ReLU activation followed by LayerNormalization is applied :

$$h_i^{(1)} = LayerNorm(ReLU(W_{in}.h_i^0 + bin)), \tag{5.20}$$

  where W_{in} is a linear layer that maps the feature vectors to the hidden dimension.

- For each pair of nodes (i,j), we compute the angular distance:

$$\theta_{i,j}^x = \cos^{-1} \frac{x_i^l.x_j^l}{||x_i^l||||x_j^l||}, \tag{5.21}$$

$$\theta_{i,j}^h = \cos^{-1} \frac{h_i^l.h_j^l}{||h_i^l||||h_j^l||}, \tag{5.22}$$

  here h^l stands for the respective node features at layer l and x^l stands for n-dimensional coordinates of nodes.

- We have used an angle-based and spatial attention mechanism as a modulating factor to the direction information, which we are computing as $h_i - h_j$ and $x_i - x_j$

$$M_{ij} = \alpha^h_{ij}(h^l_i - h^l_j) + \alpha^x_{ij}(x^l_i - x^l_j). \tag{5.23}$$

The attention computation mechanism involves using a fixed or learnable Fourier basis:

$$\phi(\theta) = [cos(k\theta), sin(k\theta)], \text{ where k=1..K} \tag{5.24}$$

followed by passing it through an MLP layer:

$$\alpha^h_{ij} = MLP_f(\phi(\theta^h_{ij})). \tag{5.25}$$

,

$$\alpha^x_{ij} = MLP_x(\phi(\theta^x_{ij})). \tag{5.26}$$

- For each node i, we update its coordinate using the computed Message vector M_{ij}:

$$x^{l+1}_i = x^l_i + \sum_{j \epsilon \mathcal{N}(i)} (x^l_i - x^l_j).W_m(M_{ij}), \tag{5.27}$$

where W_m is a learnable linear layer applied to the message vector.

- The updated coordinates of the central and the neighbouring nodes are further used to calculate the aggregated angular distance from the neighbours

$$\theta^i_{agg} = \sum_{j \in N(i)} \cos^{-1} \frac{x^{l+1}_i \cdot x^{l+1}_j}{||x^{l+1}_i|| ||x^{l+1}_j||}. \tag{5.28}$$

- The aggregated angle computed in the previous step is used to construct the rotation matrix.

$$h^{l+1}_i = \mathcal{R}_{\theta agg} h^l_i. \tag{5.29}$$

- After L layers of aggregation and rotation, the feature vectors are passed through multiple MLPs with non-linear activation for node-level downstream tasks. We use an additional graph-level pooling layer f_{pool} in the middle of two MLPs for graph-level prediction tasks.

$$h^{out}_i = MLP_1(h^L_i). \tag{5.30}$$

$$H_G = f_{Pool}(h^{out}_i). \tag{5.31}$$

$$y_G = MLP_2(H_G). \tag{5.32}$$

5.5 Analysis of the Proposed Model - Version 2

As can be seen in the new model design above, it uses an inter-feedback mechanism between the coordinates and the feature update process. This model involves a cyclical interaction between the nodes and the features that happens at each of the L GNN layers. The computed angular distance is used to construct direction-sensitive message vectors (M). These message vectors are then used to update the coordinates, which means that features affect the coordinate updates. The updated coordinates are further used to compute the aggregated angles per node, which further leads to the rotation of the feature vectors using the aggregated angle (derived from the coordinates).
So, in this manner, the coordinates affect the feature update process. The following figure shows the interrelationship between the two factors. The feature influences the coordinates through messages, while the coordinates influence the features through the rotational transformation. Such rotation of features based on the spatial changes allows the model to learn directional and orientation-dependent interaction, which is hard to capture by just using the message passing scheme. Also, the intertwined update mechanism leads to a structure-aware representation that can show a better generalization to unseen or rotated inputs. Since molecular data constantly undergo such transformations, including such a process in the model should show us a better performance and should also improve the learning process in each iteration/layer.

The model uses a Fourier-based angular encoding mechanism [16] to compute the attention. In the tasks involving 3D/2D geometry, the use of raw angles does not lead to smooth representations due to their periodic nature. Neural networks often struggle due to this unless angles are properly encoded. Instead of using raw angles, mapping them to a higher-dimensional space using Fourier features makes the angular relationship smoother and more expressive.

Unlike the earlier version of the model, that can capture only up to the first harmonics.

$$\phi_1(\theta) = [cos(\theta), sin(\theta)]. \tag{5.33}$$

This new version of the model can capture multiple harmonics of up to K frequencies, which is sensitive to finer angular variations. Higher values of k make the model more sensitive to angular changes and small directional shifts.

The coordinate update process in this version of the model shows rotation and translation Equivariance. If Q is an $n \times n$ real orthogonal matrix and g is an n-dimensional translation vector, then applying these transformations on the coordinate update expression gives us:

$$
\begin{aligned}
Qx_i^{l+1} + g &= (Qx_i^l + g) + \sum_{j \in \mathcal{N}(i)} (Qx_i^l - Qx_j^l) W_m(M_{ij}) \\
&= Q \left(x_i^l + \sum_{j \in \mathcal{N}(i)} (x_i^l - x_j^l) W_m(M_{ij}) \right) + g \\
&= Qx_i^{l+1} + g. \qquad (5.34)
\end{aligned}
$$

Rotating and translating x^l will lead to similar transformations to x^{l+1}.

Chapter 6

Experimental Results and Dataset Overview

In order to check the performance of our proposed GNN models, we have tested the 2 versions on separate sets of small and large-scale graph datasets. The models with high expressive power are supposed to compute a better representation, which can help in downstream classification and regression tasks.

- The version 1 model has been run on small and large-scale graph classification tasks that involved multiple citation networks and co-purchase networks. The results, along with the baseline model comparison, have been shown in the underlying sections.
- We tested our version 2 model comprehensively on small and large-scale benchmark molecular datasets. Now these datasets span a wide range of graph learning tasks that involve property prediction, toxicity classification, and quantum chemistry regression, which offers us a way to test the generalization and performance of the model.

6.1 Citation Networks Dataset

6.1.1 Dataset Details

We have used the Planetoid dataset, which is a benchmark citation network dataset from Yang et al., Paper [17]. The datasets are named **"Cora", "CiteSeer"** and **"Pubmed"**. In this graph dataset, the nodes represent the documents, respectively, while the edges represent the citation links between the papers. Binary masks have been used to label the train, test, and validation splits in the dataset. The feature vector for each document is encoded as a sparse bag-of-words. Every document has a class label. Since each of the nodes, which are research papers, has

been categorized into different fields/domains, the goal involves the accurate prediction of the topic or the field using the node features and the graph structure.

We can refer to the following table to look into the necessary details of these datasets.

Table 6.1: Details of the Planetoid dataset

Dataset Name	**Nodes**	**Edges**	**Classes**	**Features**	**Label Rate**
Cora	2,708	10,556	7	1,433	0.052
Citeseer	3,327	9,104	6	3,703	0.036
Pubmed	19,717	88,648	3	500	0.003

In the above table, the Label Rate can be stated as the number of labeled nodes that are used for training divided by the total number of nodes in each dataset.

6.1.2 Experimental Setup

For performing the experiments, we have used the same setup that was mentioned in Yang et al. Paper [17]. The dataset split uses 20 randomly labeled nodes from every class. An additional validation set of 500 labeled examples was also taken based on the Kipf et al. paper [2]. We have used accuracy as a performance evaluation metric for classification after referencing the mentioned papers.

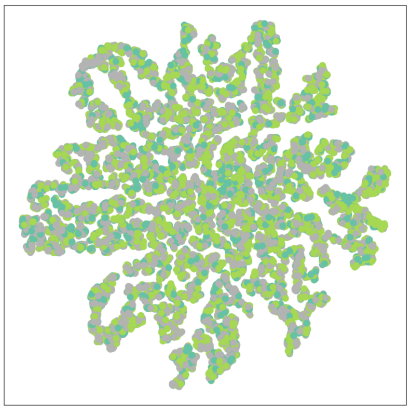
The following hyperparameter settings were used to perform experiments on all three datasets

Table 6.2: Hyperparameter settings for Experiments on Planetoid Dataset

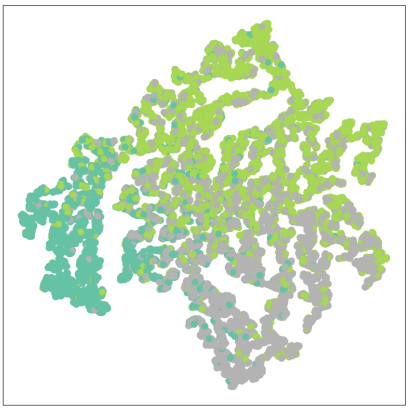
Epochs	**LR**	**Weight Decay**	**Hidden**	**Dropout**	**Layers**	**Coordinate Dim**
100	1e-3	1e-16	256	0.5	2	3

6.1.3 Results and Visualizations

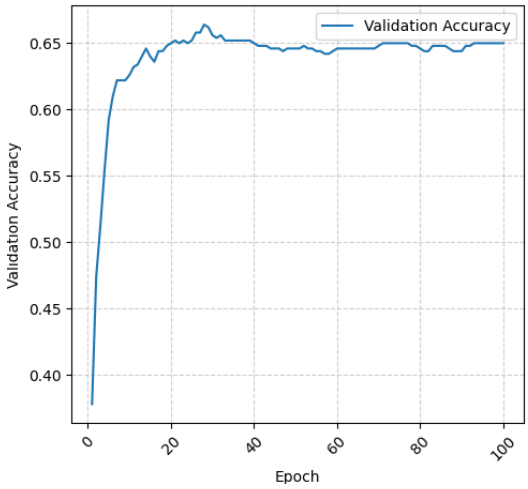
We have depicted the **TSNE** plots of the node of embeddings before and after training the model, along with the plots showing the training dynamics.



(a) Before Training



(b) After Training



(c) Validation Accuracy Plot

Figure 6.1: Dataset:Pubmed. Top-Left: TSNE plot of node embeddings before training the model. Top-Right: TSNE plot on the entire dataset after model training. Bottom: Validation Accuracy vs. Epochs on the Pubmed Dataset

The top right TSNE plot clearly shows the cluster formation represented by 3 different colours. It depicts the model's expressive power and better representation of computation. ability, as a result, it is able to make a clear distinction between the nodes in separate categories, compared to the top left plot, where the embeddings are indistinguishable from one another. The figure also depicts the rapid initial improvement in the validation accuracy with stagnation at epoch 30.

We have compared our results against some of the well-known baseline models **EGNN** [13] and **EXPC** [14]. Since these models were not evaluated on the same dataset used in our experiments, we re-implemented them and have conducted evaluations under consistent experimental settings for a fair and direct comparison with our model.
Our Experimental Results have been summarized below. Here, we report the mean and standard error of prediction accuracy on the test set split in percent over 10 runs with random initializations of model parameters

Table 6.3: Summary of results in terms of classification accuracy (in percentage).

Methods	**Cora**	**Citeseer**	**Pubmed**
EGNN [13]	46.74 ± 2.63	35.58 ± 7.18	**66.1± 3.77**
EXPC [14]	36.74± 7.00	26.26± 4.78	55.6± 13.04
Variant 1 Model	**48.14± 2.42**	**43.88± 1.98**	65.52± 1.36

Higher is better

These results demonstrate that Variant 1 outperforms the baseline models on the Cora and Citeseer datasets. Although EGNN slightly surpasses Variant 1 on Pubmed in terms of mean accuracy, the latter achieves significantly lower variance, indicating greater consistency.The improved accuracy and reduced variability of Variant 1 indicate its potential as a more reliable model for graph classification tasks across diverse datasets.

6.2 Co-Purchase Networks Dataset

6.2.1 Dataset Details

We have used two subsets of the Amazon Co-purchase graphs as part of this experimentation. The **Amazon Computer** and the **Amazon Photo** datasets from the **"Pitfalls of Graph Neural Network Evaluation"** [18] paper were released for research on community detection on large-scale graphs. These datasets are commonly used for evaluating Graph Neural Networks (GNNs) in semi-supervised and unsupervised learning tasks. In this dataset, the nodes represent purchase items/goods, and edges represent that 2 goods are frequently purchased together. Each product is described by a bag-of-words (BoW) feature vector derived from its product description. The feature vectors are sparse and high-dimensional. Every product has a label based on the product category to which it belongs, and the classification task involves the correct identification of the product category for each product.

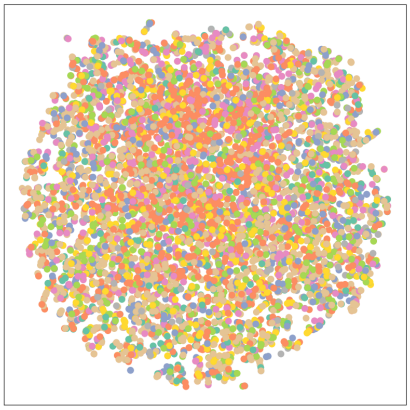
Table 6.4: Details of the Amazon Co-Purchase Dataset

Dataset	Nodes	Edges	Classes	Features
Computers	13,752	491,722	10	767
Photo	7,650	238,162	8	745

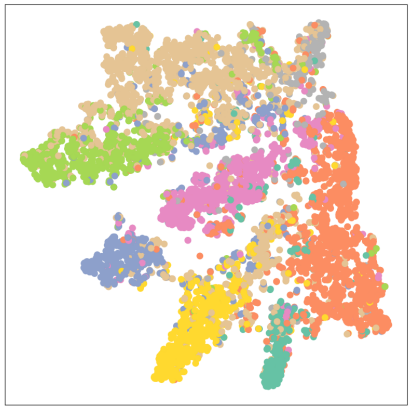
6.2.2 Experimental Setup

We have maintained the same hyperparameter settings as were used in the case of Citation Networks. Only difference exists in the dataset splits, here we have randomly sampled 20 labeled nodes per class as the training set, 30 nodes per class as the validation set, and the rest as the test set [17]. This approach ensures a consistent evaluation protocol.Such a setup is commonly adopted to simulate semi-supervised learning conditions, where labeled data is limited and the model must generalize effectively across unseen nodes.

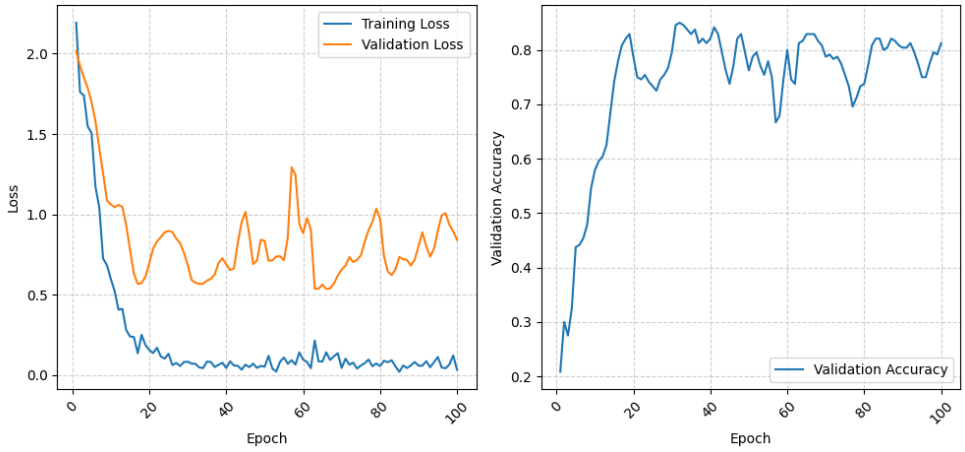
6.2.3 Results and Visualizations



(a) Before Training

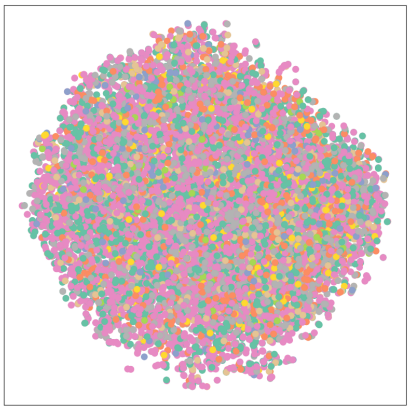


(b) After Training

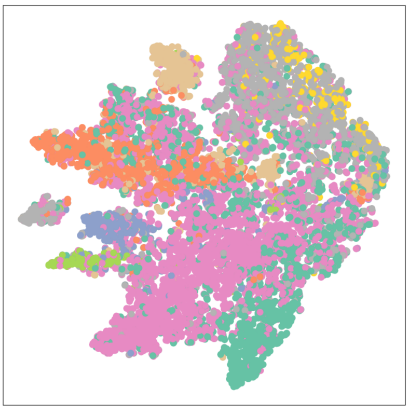


(c) Loss Curves & Validation Accuracy Plots

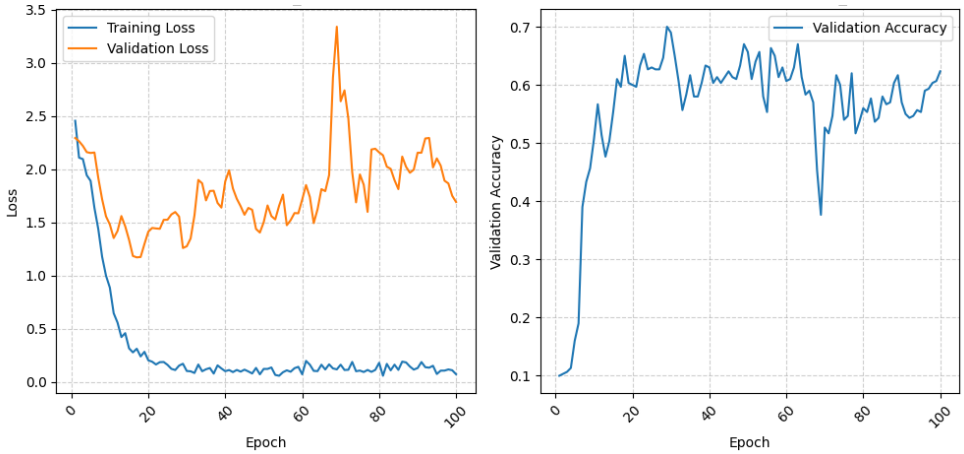
Figure 6.2: Dataset:Amazon Photo. Top-Left: TSNE plot of node embeddings before training the model. Top-Right: TSNE plot on the entire dataset after model training. Bottom: Loss curves and validation accuracy plot



(a) Before Training



(b) After Training



(c) Loss Curves & Validation Accuracy Plots

Figure 6.3: Dataset:Amazon Computer. Top-Left: TSNE plot of node embeddings before training the model. Top-Right: TSNE plot on the entire dataset after model training. Bottom: Loss curves and validation accuracy plot

The model performed better for both the large-scale co-purchase graph datasets. The clusters are very clearly visible for the Amazon Photo dataset. This demonstrates the model's high expressive power, which allowed it to clearly distinguish among the items belonging to separate categories. The top left visuals depict that the embeddings are intermixed and are indistinguishable from one another when no training epochs were executed. Once we followed the training procedure and trained the model, the model accurately predicted and distinguished the items belonging to separate categories.

Below, we have reported the mean and standard error of prediction accuracy on the test set split in percent over 10 runs with random initializations of model parameters

Table 6.5: Summary of results in terms of classification accuracy (in percent).

Methods	**Amazon Photo**	**Amazon Computers**
EGNN [13]	44.64± 2.776	39.512± 3.91
EXPC [14]	37.86± 9.43	32.66± 8.99
Variant 1 Model	**72.35± 1.74**	**52.82± 3.54**

Higher is better

The results show that our model successfully outperformed the baseline models by large margins. It even showed very little standard deviation as compared to the baselines.

To assess the performance of the Variant 1 Model on node classification tasks, we conducted experiments across five benchmark citation and co-purchase graph datasets—Cora, Citeseer, Pubmed, Amazon Photo, and Amazon Computers—and compared the results against EGNN and EXPC models.

The proposed model consistently achieved strong classification performance across all datasets. On Cora, it obtained an accuracy of 48.14 ± 2.42, outperforming both EGNN (46.74 ± 2.63) and EXPC (36.74 ± 7.00). A similar trend was observed in Citeseer, where Variant 1 reached 43.88 ± 1.98, surpassing EGNN by over 8 percentage points and EXPC by an even wider margin.

On the Pubmed dataset, which is known for its relatively clean citation structure, EGNN slightly edged out Variant 1 with a peak performance of 66.1 ± 3.77. Nonethe-

less, Variant 1 was close behind at 65.52 ± 1.36, demonstrating its competitiveness even on large-scale graphs with complex topologies.

Notably, the model excelled on the Amazon co-purchase datasets. For Amazon Photo, Variant 1 achieved a standout accuracy of 72.35 ± 1.74, significantly outperforming EGNN (44.64 ± 2.78) and EXPC (37.86 ± 9.43). In Amazon Computers, it again led the pack with 52.82 ± 3.54, clearly indicating its effectiveness in capturing the hierarchical and semantic structure of product co-purchase relationships.

Overall, these results demonstrate the robustness and generalizability of the Variant 1 Model across a diverse range of graph classification benchmarks, outperforming or matching state-of-the-art baselines in most scenarios.

An additional conclusion that can be drawn is that our version-1 model performed better on large-scale graphs like Pubmed, Amazon Computer & Photo, as compared to the small-scale ones like Cora and Citeseer.

6.3 Molecular Datasets

We have used the version 2 model to conduct extensive experiments on some of the benchmark molecular datasets. Some of the datasets, like the **QM9** dataset, are a standard in machine learning in the chemical property prediction task. In the following sections we will show the execution results, along with the visual plots, and also compare the performance with some baseline models.

6.3.1 Dataset Details

We have used 3 datasets to conduct our experiments the details regarding them are summarized below:

6.3.1.1 Open Graph Benchmark Dataset (OGB)

The Open Graph Benchmark (OGB) is a collection of standardized large-scale graph datasets that have been designed for the fair evaluation of graph machine learning models. It has been provided with consistent data splits and evaluation metrics. As part of this benchmark, I have conducted experiments on two of its graph-level datasets — **MolHIV** and **PPA**. The **OGBG-MolHIV** dataset focuses on molecular property prediction, where the task is to determine whether a molecule can inhibit HIV replication. Molecules are represented as graphs, with atoms as nodes and chemical bonds as edges. This binary classification task is particularly important in the context of drug discovery.The **OGBG-PPA (Protein-Protein Association)** dataset consists of biological networks where nodes represent proteins and edges denote functional associations. The task involves multi-class graph classification and is especially challenging due to the complexity and size of the graphs, making it a strong benchmark for evaluating model generalization on biological data.

Table 6.6: Summary of the OGB dataset used in our experiment, including the number of graphs, nodes, edges, and evaluation metric

Dataset Name	#Graphs	#Nodes per graph	#Edges per graph	Task Type	Metric
ogbg-molhiv	41,127	25.5	27.5	Binary classification	ROC-AUC
ogbg-ppa	158,100	243.4	2,266.1	Multi-class classification	Accuracy

6.3.1.2 Molecular data - QM9

The **QM9 dataset**[19] is a standard benchmark in molecular machine learning, containing around 134,000 small organic molecules made up of carbon, oxygen, nitrogen, and fluorine atoms. Each molecule comes with detailed information about its structure and quantum chemical properties, such as energy levels, dipole moments, and molecular orbital values. What makes QM9 particularly valuable is that it provides both the molecular graph (connectivity) and precise 3D coordinates, enabling models to learn from both the topology and geometry of molecules. This makes it a popular choice for developing and evaluating models that aim to predict molecular properties from structure.

#Graphs	#Nodes	#Edges	#Features	#Tasks
130,831	18.0	37.3	11	19

Table 6.7: Statistics of The QM9 Dataset

6.3.1.3 Zinc Dataset

The **ZINC** dataset[20] is a commonly used benchmark in molecular graph learning, originally derived from the ZINC database of commercially available chemical compounds. It contains a curated subset of molecules along with their penalized logP scores—a measure that combines properties like solubility and synthetic accessibility. Each molecule is represented as a graph, where atoms are nodes and chemical bonds are edges, making it ideal for testing graph neural network models. Due to its moderate size and meaningful regression target, the ZINC dataset has become a go-to

choice for evaluating the expressiveness of graph-based architectures, particularly in the context of molecule-level property prediction.

6.3.2 Experimental Setup

In this section, we present the experimental settings adopted for evaluating the proposed model on multiple molecular datasets from the Open Graph Benchmark suite. These datasets differ in scale and complexity, and help assess the robustness and generalizability of the model across distinct graph-based tasks. We now detail the experimental configuration used specifically for each dataset.

6.3.2.1 Experimental setup for OGB Dataset

We have used the same experimental setup as mentioned in [14]. The dataset split was chosen to be the default one consistent with [14].

Table 6.8: Hyperparameter Settings for Experiments on OGB Datasets

Hyperparameter	ogbg-molhiv	ogbg-ppa
Epochs	401	601
Learning Rate	0.0001	0.0005
Decay Rate	0.7	0.8
Hidden Dimension	64	256
Dropout	0.5	0.5
Layers	3	4
Batch Size	64	32
Pooling	MEAN	SUM

6.3.2.2 Experimental setup For QM9 Dataset

The hyperparameter settings have been mentioned in the following table, which is consistent with [13]. We have trained our model for each individual property up to 100 epochs, the other parameters, like the optimizer, weight decay, batch size, and use of cosine decay for learning rate, are the same as what is mentioned in [13], except for the number of layers, we used 5 layers instead of 7 as mentioned in the paper due to computational resource constraints. Additionally, all properties were normalized by subtracting the mean and dividing by the Mean Absolute Deviation.

Table 6.9: Hyperparameter settings for Experiments on QM9 Datasets

Epochs	LR	Weight Decay	Dropout	Layers	Batch Size	Pooling
1000	1e-3	1e-16	0.5	5	96	SUM

6.3.2.3 Experimental setup for ZINC Dataset

We have chosen the Zinc Sparse graph for our experiments due to computational constraints. The hyperparameter settings are based on [21]. Refer to the table below for our experimental settings.

Table 6.10: Hyperparameter settings for experiments on the ZINC dataset

Epochs	Batch Size	Init LR	LR Reduce Factor	Hidden Dim	Readout
1000	128	0.0007	0.5	64	MEAN

6.3.3 Quantitative Results

6.3.3.1 Results on the OGB Dataset

Below we have reported the performance of our model along with that we have listed the performance of some well-known baseline models. The evaluation metric can be found in the dataset summary table mentioned in the earlier sections.
We have stated the mean and standard error of the corresponding evaluation metric after running our model on the test set split in percent over 10 runs with random initializations of model parameters.

Table 6.11: Performance comparison on ogbg-molhiv and ogbg-ppa datasets

Methods	ogbg-molhiv	ogbg-ppa
GCN [2]	76.06± 0.97	68.39± 0.84
GIN [8]	75.58± 1.40	68.92± 1.0
ExpC-3,4,5 [14]	**77.89**	**80.11**
Variant 2 Model	70.89 ± 1.36	65.73 ± 1.22

Higher is better

On the **ogbg-molhiv** dataset, the Variant 2 Model achieved a score of 70.89 ± 1.36, which was notably lower than the best-performing method (ExpC-3,4,5) that reached 77.89. A similar trend was observed in the **ogbg-ppa** dataset, where the Variant 2 Model scored 65.73 ± 1.22, again falling short of the highest benchmark

performance of 80.11. These results suggest that while the model can learn meaningful representations, it struggles to match the predictive accuracy of more optimized baselines in large-scale bioactivity and protein-protein interaction tasks.

6.3.3.2 Results on the QM9 Dataset

The following table shows the Mean Absolute Error for the molecular property prediction task in the QM9 dataset. The results highlight the comparative performance of various models, providing insight into their accuracy in estimating quantum chemical properties. Lower MAE values indicate better model precision, which is crucial for downstream tasks in computational chemistry.

Table 6.12: Mean Absolute Error (MAE) for various molecular property prediction tasks on the QM9 benchmark

Task (Units)	**Schnet** [22]	**EGNN** [13]	**Variant 2 Model**
α (bohr^3)	0.235	**0.071**	3.9705
$\Delta\varepsilon$ (meV)	63	**48**	87
$\varepsilon_{\text{HOMO}}$ (meV)	41	**29**	43
$\varepsilon_{\text{LUMO}}$ (meV)	34	**25**	80
μ (D)	0.033	**0.029**	1.0684
C_ν (cal/mol·K)	0.033	**0.031**	2.4235
G (meV)	14	**12**	66
H (meV)	14	**12**	67
R^2 (bohr^3)	**0.073**	0.106	1.74
U (meV)	19	**12**	75
U_0 (meV)	14	**11**	64
ZPVE (meV)	1.70	**1.55**	6.28

Lower is better

For the **QM9** molecular property prediction benchmark, which involves multiple regression tasks, the Variant 2 Model generally produced higher mean absolute errors (MAEs) than its competitors. For example, in the prediction of molecular polarizability α, the model reported a value of 3.97 bohr^3 compared to the significantly lower errors from SchNet (0.235) and EGNN (0.071). This pattern of underperformance

was consistent across most properties, such as dipole moment μ, heat capacity, and internal energies (U, U_o), indicating that the Variant 2 Model may require additional tuning or architectural refinement to better capture the subtle quantum mechanical interactions represented in the dataset.

6.3.3.3 Results on the Zinc Dataset

The following table shows the performance of our model and some other baselines on the ZINC dataset. The Performance Measure used is MAE. Results are the average of the MAE values over 4 runs with 4 different seeds for random initialization of model parameters.

Table 6.13: Summary of Model Performance on Zinc Dataset With Baseline Results

Model	**TestPerf ± s.d.**
GCN [2]	0.367 ± 0.011
GAT [3]	0.384 ± 0.007
Graph Transformer [21]	**0.226 ± 0.014**
Variant 2 Model	0.4907 ± 0.177

Lower is better

The **ZINC dataset** results also reflect a similar gap, where the proposed model obtained a test performance of 0.4907 ± 0.177, substantially higher (and thus worse, since lower is better) than the Graph Transformer's 0.226 ± 0.014. This indicates that the Variant 2 Model currently lacks the expressivity or regularization strength required to generalize well in molecular graph generation tasks.

In summary, although the Variant 2 Model demonstrates consistent training and reasonable convergence, its overall predictive performance remains behind more mature models. These insights provide valuable direction for future improvements in model design and training strategies.Further investigation into architectural refinements, feature encoding mechanisms, and regularization techniques may help bridge this performance gap. Additionally, leveraging advanced optimization schedules or incorporating domain-specific inductive biases could enhance both convergence speed and generalization. Future work should also explore model robustness across diverse datasets to ensure broader applicability.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Across a diverse set of benchmarks, the performance of the proposed Variant models reveals a mixed yet insightful picture. Variant 1 Model demonstrates strong classification capabilities on citation network datasets like Cora and Citeseer, outperforming baseline models such as EGNN and EXPC in terms of accuracy. This trend is even more prominent on co-purchase datasets, where Variant 1 significantly outperforms both EGNN and EXPC, especially on Amazon Photo with a substantial margin.

However, the Variant 2 Model underperforms on molecular prediction tasks from the OGB benchmark. On both ogbg-molhiv and ogbg-ppa, it trails behind existing methods like GCN, GIN, and especially the ExpC variants, which set the highest scores. Similarly, on the regression-based ZINC dataset, where lower values indicate better performance, Variant 2 reports the highest test error, suggesting room for substantial improvement.

These results indicate that while Variant 1 exhibits strong generalization on node classification tasks, Variant 2 struggles with molecular property prediction and graph regression tasks. The observed variability underscores the importance of architecture-task alignment and the need for tailored inductive biases in model design.
The implementation details, including the complete codebase and result plots, are available on GitHub.The code is written in PyTorch along with relevant supporting libraries. The link is given below.

https://github.com/suranjandey2012/Dissertation-On-Graph-Neural-Networks-Expressive-Power-Enhancement-Using-Geometric-Transformation

7.2 Scope for Future Work

Given the divergent performance across tasks, future research should focus on adapting model architectures to the specific characteristics of each dataset type.

Better Molecular Feature Embedding For Variant 2, strategies such as incorporating domain-aware graph pooling, better molecular feature embeddings, or enhanced attention mechanisms may lead to improvements on biochemical datasets like OGB and ZINC.

Dynamic architecture adaptation Additionally, exploring dynamic architecture adaptation, where the model structure evolves based on the dataset properties, could offer a promising direction. Another valuable avenue would be to integrate training stability techniques, such as learning rate warm-ups or adversarial training, particularly for tasks where high variance was observed.

Finally, a comprehensive ablation study focusing on individual components of the proposed models could provide deeper insights into performance bottlenecks, informing more targeted enhancements in future iterations.

Bibliography

[1] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. *Neural message passing for quantum chemistry.* arXiv:1704.01212, 2017.

[2] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks.* arXiv preprint arXiv:1609.02907, 2016.

[3] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. *Graph Attention Networks.* arXiv preprint arXiv:1710.10903, 2017.

[4] William L. Hamilton, Rex Ying, and Jure Leskovec. *Inductive Representation Learning on Large Graphs.* arXiv preprint arXiv:1706.02216, 2017.

[5] Boris Weisfeiler and Andrei Leman. *The reduction of a graph to canonical form and the algebra which appears therein.* NTI,Series, 2(9):12–16, 1968.

[6] L´ aszl´ o Babai and Ludik Kucera. *Canonical labelling of graphs in linear average time.* In 20th Annual Symposium on Foundations of Computer Science (sfcs 1979), pages 39–46. IEEE, 1979.

[7] Ronald C Read and Derek G Corneil. *The graph isomorphism disease.* Journal of graph theory, 1(4):339–363, 1977.

[8] Xu, Keyulu and Hu, Weihua and Leskovec, Jure and Jegelka, Stefanie *How Powerful are Graph Neural Networks?* arXiv e-prints arXiv:1810.00826,2018.

[9] Han, Jiaqi and Rong, Yu and Xu, Tingyang and Huang, Wenbing *Geometrically Equivariant Graph Neural Networks: A Survey* arXiv e-prints arXiv:2202.07230

[10] Nikolentzos, Giannis and Dasoulas, George and Vazirgiannis, Michalis *k-hop Graph Neural Networks* arXiv:1907.06051

[11] Zhang, Bingxu and Fan, Changjun and Liu, Shixuan and Huang, Kuihua and Zhao, Xiang and Huang, Jincai and Liu, Zhong *The Expressive Power of Graph Neural Networks: A Survey* arXiv:2308.08235

[12] Morris, Christopher and Ritzert, Martin and Fey, Matthias and Hamilton, William L. and Lenssen, Jan Eric and Rattan, Gaurav and Grohe, Martin *Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks* arXiv:1810.02244, 2018

[13] Garcia Satorras, Victor and Hoogeboom, Emiel and Welling, Max *E(n) Equivariant Graph Neural Networks* arXiv:2102.09844, 2021

[14] Yang, Mingqi and Shen, Yanming and Qi, Heng and Yin, Baocai *Breaking the Expressive Bottlenecks of Graph Neural Networks* arXiv:2012.07219, 2020

[15] Gasteiger, Johannes and Groß, Janek and Günnemann, Stephan *Directional Message Passing for Molecular Graphs* arXiv:2003.03123, 2020

[16] Hsu, Tim and Pham, Tuan Anh and Keilbart, Nathan and Weitzner, Stephen and Chapman, James and Xiao, Penghao and Qiu, S. Roger and Chen, Xiao and Wood, Brandon C. *Efficient and interpretable graph network representation for angle-dependent properties applied to optical spectroscopy* arXiv:2109.11576, 2022

[17] Yang, Zhilin and Cohen, William W. and Salakhutdinov, Ruslan *Revisiting Semi-Supervised Learning with Graph Embeddings* arXiv:1603.08861, 2016

[18] Shchur, Oleksandr and Mumme, Maximilian and Bojchevski, Aleksandar and Gunnemann, Stephan *Pitfalls of Graph Neural Network Evaluation* arXiv:1811.05868, 2018

[19] Wu, Zhenqin and Ramsundar, Bharath and Feinberg, Evan N. and Gomes, Joseph and Geniesse, Caleb and Pappu, Aneesh S. and Leswing, Karl and Pande, Vijay *MoleculeNet: A Benchmark for Molecular Machine Learning* arXiv:1703.00564, 2017

[20] Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; and Coleman, R. G. *ZINC: a free tool to discover chemistry for biology.* Journal of chemical information and modeling 52(7): 1757–1768. 2012.

[21] Dwivedi, Vijay Prakash and Bresson, Xavier *A Generalization of Transformer Networks to Graphs* arXiv:2012.09699, 2020

[22] Kristof T. Schütt, Pieter-Jan Kindermans, Huziel E. Sauceda, Stefan Chmiela, Alexandre Tkatchenko, Klaus-Robert Müller *SchNet: A continuous-filter convolutional neural network for modeling quantum interactions* arXiv:1706.08566, 2017

15% Overall Similarity

Top sources found in the following databases:

- 11% Internet database
- Crossref database
- 0% Submitted Works database
- 10% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	arxiv.org Internet	3%
2	dspace.isical.ac.in:8080 Internet	1%
3	library.isical.ac.in:8080 Internet	1%
4	export.arxiv.org Internet	<1%
5	web.archive.org Internet	<1%
6	pure.uva.nl Internet	<1%
7	"ECAI 2020", IOS Press, 2020 Crossref	<1%
8	"Graph Neural Networks: Foundations, Frontiers, and Applications", Spr... Crossref	<1%

9	Hu, Weihua. "On the Predictive Power of Graph Neural Networks", Stanf... Publication	<1%
10	hdl.handle.net Internet	<1%
11	geeksforgeeks.org Internet	<1%
12	Nico Piatkowski, Katharina Morik, Nils Kriege, Christopher Morris et al. ... Crossref	<1%
13	ndltd.ncl.edu.tw Internet	<1%
14	arxiv-vanity.com Internet	<1%
15	ijcai.org Internet	<1%
16	github.com Internet	<1%
17	Hui Xu, Liyao Xiang, Femke Huang, Yuting Weng, Ruijie Xu, Xinbing Wa... Crossref	<1%
18	Jiang Qiangrong, Xiong Zhikang, Zhai Can. "Graph Kernels and Applica... Crossref	<1%
19	slideshare.net Internet	<1%
20	Kong, Kezhi. "Towards Generalized and Scalable Machine Learning on ... Publication	<1%

21	Silvia Beddar-Wiesing, Giuseppe Alessio D'Inverno, Caterina Graziani, ... Crossref	<1%
22	Zhihao Dong, Yuanzhu Chen, Terrence S. Tricco, Cheng Li, Ting Hu. "Eg... Crossref	<1%
23	openreview.net Internet	<1%
24	pytorch-geometric.readthedocs.io Internet	<1%
25	coursehero.com Internet	<1%
26	Aburidi, Mohammed. "Optimization for Robust and Interpretable Learni... Publication	<1%
27	Anna Fergusson, Maxine Pfannkuch. "Development of an informal test ... Crossref	<1%
28	raw.githubusercontent.com Internet	<1%
29	Du, Boxin. "Multi-Network Association: Algorithms and Applications.", ... Publication	<1%
30	Pandey, Shweta. "A New Perspective on Graph Equivalence : k-Hop Si... Publication	<1%
31	Sanyal, Deepayan. "Semi-Supervised Clustering Of Stable Instances", I... Publication	<1%
32	Wijesinghe, Asiri Suranga. "Geometric Learning on Graph Structured D... Publication	<1%

33	essay.utwente.nl Internet	<1%
34	m.moam.info Internet	<1%
35	scribd.com Internet	<1%
36	Braga, Diogo Filipe Ribeiro Ferreira. "Graph Kernels and Neural Networ... Publication	<1%
37	John Shi, Mark Cheung, Wendy Summer, Jose M. F. Moura. "A Dual Ap... Crossref	<1%
38	ar5iv.labs.arxiv.org Internet	<1%
39	iris.unipa.it Internet	<1%
40	madoc.bib.uni-mannheim.de Internet	<1%
41	Chengjie Huang, Ci Chen, Zhi Liu, Zhenni Li. "Adaptive Neural Consens... Crossref	<1%
42	Giuseppe Alessio D'Inverno, Simone Brugiapaglia, Mirco Ravanelli. "Ge... Crossref	<1%
43	Kinaci, Alper. "First Principles Investigation of Hydrogen Storage in Int... Publication	<1%
44	Martino Lupini, Laura Mančinska, David E. Roberson. "Nonlocal games ... Crossref	<1%

45	Srinivasan, Balasubramaniam. "On Higher Order Graph Representation ... Publication	<1%
46	Wei, Xinxu. "Learning Deep Discriminative Features for Retinal Vessel ... Publication	<1%
47	faculty.nps.edu Internet	<1%
48	manualzz.com Internet	<1%
49	"Neural Computing for Advanced Applications", Springer Science and B... Crossref	<1%
50	Bisheng Yang, Zhen Dong, Fuxun Liang, Xiaoxin Mi. "Ubiquitous Point C... Publication	<1%
51	Chen, Yongqiang. "Learning Causality for Modern Machine Learning.", ... Publication	<1%
52	Guanbo Wang, Guoguang Cheng, Yanling Zhang, Lie Chen, Lei Hui, Qia... Crossref	<1%
53	K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, K.-R. ... Crossref	<1%
54	Liu, Juncheng. "Toward Effective and Efficient Graph Neural Networks ... Publication	<1%
55	Mardikoraem, Mehrsa. "Navigating Protein Fitness Landscapes With M... Publication	<1%
56	Yan, An. "Efficient Style-Sensitive Natural Language Processing: Enhan... Publication	<1%

57	d-nb.info Internet	<1%
58	dspace.cvut.cz Internet	<1%
59	escholarship.org Internet	<1%
60	fau.digital.flvc.org Internet	<1%
61	impa.usc.edu Internet	<1%
62	mafiadoc.com Internet	<1%
63	philpapers.org Internet	<1%
64	shortscience.org Internet	<1%
65	theses.gla.ac.uk Internet	<1%
66	nssce.ac.in Internet	<1%
67	Kaixuan Chen, Shunyu Liu, Tongtian Zhu, Ji Qiao et al. "Improving Expr... Crossref	<1%
68	Lainscsek, C.. "Equivariance identification using delay differential equa... Crossref	<1%

69	Xing Gao, Wenrui Dai, Chenglin Li, Hongkai Xiong, Pascal Frossard. "M... Crossref	<1%
70	"Advances in Knowledge Discovery and Data Mining", Springer Science... Crossref	<1%
71	Gao-Peng Ren, Ke-Jun Wu, Yuchen He. "Enhancing Molecular Represe... Crossref	<1%
72	Zhi-Peng Li, Si-Guo Wang, Qin-Hu Zhang, Yi-Jie Pan, Nai-An Xiao, Jia-Y... Crossref	<1%

0% words excluded by Custom Sections

CUSTOMIZED SECTIONS EXCLUDED

isi

Pages: 3

6 excluded words