World Wide Web (2025) 28:30
https://doi.org/10.1007/s11280-025-01343-1

Causal integration in graph neural networks toward enhanced classification: benchmarking and advancements for robust performance

Simi Job[1] · Xiaohui Tao[1] · Taotao Cai[1] · Lin Li[2] · Quan Z. Sheng[3] · Haoran Xie[4] · Jianming Yong[5]

Received: 3 October 2024 / Revised: 3 February 2025 / Accepted: 27 March 2025 /
Published online: 7 April 2025

Abstract
The expansion of Graph Neural Networks (GNNs) has highlighted the importance of evaluating their performance in real-world scenarios. However, existing evaluation frameworks often overlook the integration of causality, a critical component that is essential for more robust evaluation of GNNs. To address this gap, we present a benchmark study that systematically compares standard and causal GNN models with a focus on classification tasks. Our analysis encompasses a careful selection of nine GNN models across seven diverse datasets that span three distinct domains. The results reveal the following: I) Causality-enhanced GNNs consistently outperform their traditional counterparts in graph classification tasks; II) Models integrating causal features exhibit greater generalizability across varied datasets; and III) Incorporation of causal elements significantly improves the predictive accuracy of GNNs. These findings highlight the importance of embedding causality in the evaluation and development of GNNs for improved performance and application.

Keywords Graph neural networks · GCN · GAT · Causality · Graph classification · GraphSAGE

1 Introduction

Graph Neural Networks (GNNs) have emerged as a powerful tool for processing graph-structured data, demonstrating remarkable performance in various tasks such as node

✉ Simi Job
simi.job@unisq.edu.au

1 School of Mathematics, Physics, and Computing, University of Southern Queensland, Toowoomba, Australia

2 School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, China

3 School of Computing, Macquarie University, Sydney, Australia

4 School of Data Science, Lingnan University, Hong Kong Special Administrative Region, Hong Kong, China

5 School of Business, University of Southern Queensland, Springfield, Australia

classification [1], link prediction [2] and graph classification [1–3]. GNNs have found applications in various domains including recommendation [4], urban intelligence [5], medicine [6], community detection [7], fraud detection [8] and so on. Despite their success, GNNs face several limitations including over-smoothing, interpretability and generalizability problems, sensitivity to graph structure and limited ability in capturing long-range dependencies.

Causality, the understanding of cause and effect relationships among various factors, extends beyond mere correlations. It focuses on comprehending the interactions between elements that result in specific outcomes and explores how changes in one aspect can impact another element within a system. Recently, there has been an increased focus on exploring causality, with researchers acknowledging the importance of incorporating causal knowledge into data modelling. Causality has found numerous applications in several domains including economics [9], social sciences [10], medicine [11] and healthcare [12], environmental science [13], recommendation [14, 15] etc. For instance, in medicine, causality can explore factors that impact treatment outcomes and those that increase the risk of medical conditions. In social sciences, it can reveal the causal factors that contribute to economic inequalities. In recommendation systems, causality can uncover factors that influence user preferences and engagement. In these contexts, causal analysis can uncover factors that influence outcomes, improve model interpretability and enhance predictive accuracy. Integrating causality into GNN architecture can significantly mitigate the aforementioned limitations by prioritizing relevant information, capturing long-range dependencies, and promoting the extraction of transferable features, thereby improving generalizability. By examining inherent causal relationships within the data, it becomes possible to enhance GNN performance and application.

In this study, we aim to thoroughly investigate the application of graph neural networks for classification tasks and demonstrate the significance of causally enabled GNNs in identifying true interactions within data. Few studies have systematically benchmarked GNNs with a focus on causal classification. Existing benchmark studies such as [16], which examined graph positional encoding in GNNs, and [17], which explored the use of GNNs for fault diagnosis, have provided foundational insights into these areas. Kosan et al. [18] conducted a benchmark study that focused on GNN explainers, while [19] performed an extensive investigation into deep GNN architectures, experimenting with different model settings across various citation network datasets. All of these studies primarily evaluate GNN performance based on traditional metrics without integrating causal analysis. To address this gap, our research aims to analyse the significance of causality in generalizable graph prediction models. Specifically, we conduct a comprehensive study on the most representative models that are used in graph neural networks classification tasks, with the potential of incorporating causal elements into the respective frameworks. This empirical study contributes to the research community with the following interesting findings:

- The attention-based causal model (CAL framework) consistently outperformed baseline GNN models in larger graph classification tasks, demonstrating its ability to capture complex global patterns and dependencies across networks.
- Baseline GNN models excelled in smaller node classification tasks, highlighting their efficiency in scenarios with limited data and simpler relationships.
- Hyperparameter tuning plays a crucial role in improving model performance, and our research emphasizes the adaptability of causal models to multi-class datasets in graph classification.

These findings highlight the importance of embedding causality in the evaluation and development of GNNs for enhanced performance and application. The remainder of the paper is structured as follows: Section 2 provides an overview of research studies centered

on GNNs and causality. Section 3 outlines the study design. Section 4 presents the results of the empirical study. Finally, Section 5 concludes the paper with a brief summary of our findings.

2 Where GNN meets causality

This section reviews existing literature focusing on graph neural networks (GNNs) and their variants, causality and their applications in classification tasks.

2.1 Graph neural networks

Graph Neural Networks (GNNs) are designed to process graph-structured data, where graphs consist of nodes and edges. Each node has features that represent its characteristics and the edges define the relationships between nodes. GNNs capture dependencies between nodes through message passing, a process in which nodes exchange and aggregate information from their neighbours. In this process, a node updates its representation by incorporating features from itself and its directly connected neighbours.

2.1.1 Graph convolutional networks (GCN)

The most basic GNN is the Graph Convolutional Networks [20]. GCNs employ convolution operations to aggregate features including neighbourhood features. During convolution, information is propagated through nodes, and pooling layers are used for graph pooling. The convolutional layer captures the local graph structure by means of this aggregation operation. After the convolution operation, an activation function such as ReLU is applied. A fully connected layer is then used to combine the learned features from the convolutional layers, enabling the network to make task-specific predictions in the final output layer, as depicted in Figure 1.

2.1.2 GraphSAGE

GraphSAGE (Graph SAmple and aggreGatE) [21] is a type of GNN that uses sampling and aggregation of features from a node's neighborhood to build node representations. It samples a subset of local neighbours for each node and then aggregates the features of these neighbours using different aggregation functions such Mean, LSTM or pooling. This approach enables scalable graph processing by reducing computational complexity, while also generalizing to unseen nodes during inference.

2.1.3 Graph attention networks (GAT)

Graph Attention Networks [22] incorporate an attention mechanism into GNNs to capture complex dependencies in graph data. The attention mechanism computes attention scores for node pairs, which determine the importance of neighbouring nodes' features. These scores are used as attention coefficients to weight the features of neighboring nodes. The weighted features are then aggregated (usually through summation) to update the node's representation. This approach allows the model to focus on the most relevant neighbors and

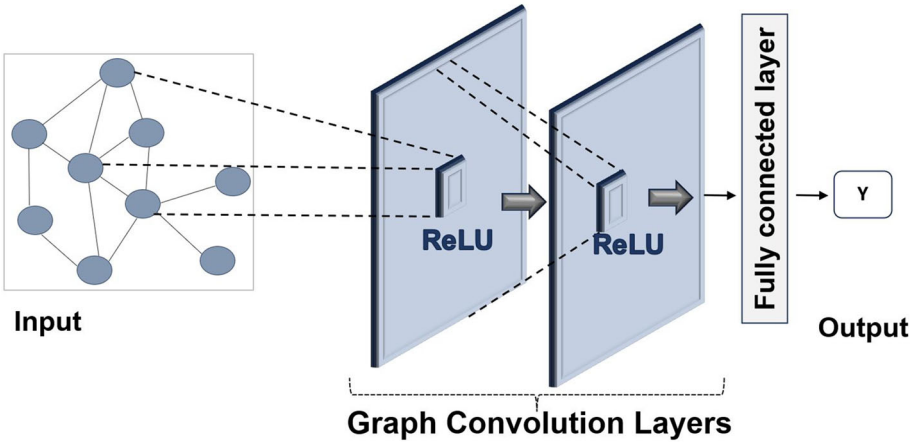

Figure 1 Graph Convolutional Networks (GCN)

effectively handle varying importance among neighbours. A simple representation of the GAT is shown in Figure 2.

2.1.4 Graph isomorphism networks (GIN)

Graph Isomorphism Networks [23] are GNNs specifically designed for the task of determining whether two graphs are structurally identical. GINs employ a customised aggregate function that makes them invariant to node ordering in a graph. They use a sum-based aggregation, where each node combines its features with those of its neighbours, followed by a non-linear activation. This enables GINs to effectively capture graph structures, making them highly suitable for tasks like graph classification.

2.1.5 Graph learning tasks

Graph tasks typically include graph classification, node classification and link prediction. *Graph classification* is a task where the goal is to predict a label or category for an entire graph. For example, in a dataset of chemical compounds, where each graph represents a

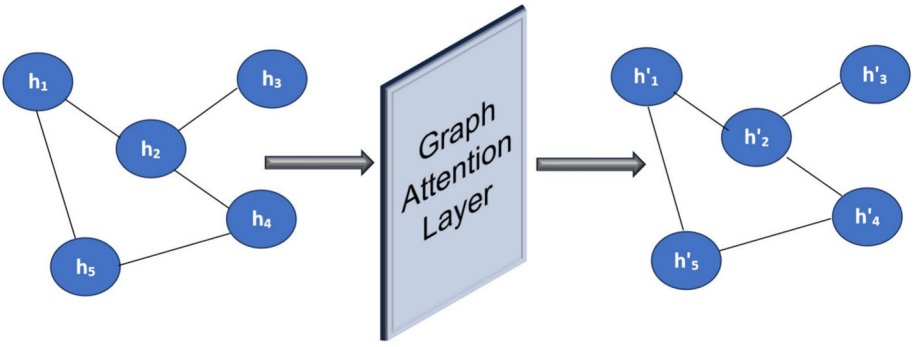

Figure 2 Graph Attention Networks (GAT)

molecule, the task is to classify whether the molecule is 'carcinogenic' or 'non-carcinogenic' based on its structure. *Node classification* is a task where the goal is to predict the labels or categories of nodes in a graph. For example, in a citation network, each node represents a research paper, and the task is to classify each paper into categories like 'Conference Paper' or 'Journal Paper' based on the type of publication. *Link prediction* is a task where the goal is to predict missing or future connections (edges) in a graph. For example, in a social network graph, where users are nodes, the task is to predict user engagement, such as, which users are likely to comment or like a post, based on existing interactions and connections.

GNNs learn node embeddings to capture structural information in graphs. Position-aware Graph Neural Networks (P-GNNs) proposed by You et al. [24], is aimed at capturing the positions of nodes in a graph. Position-aware node embeddings are computed by sampling sets of anchor nodes and estimating the distance between the target and anchor nodes. The node positional information is expected to enhance GNN performance in various tasks such as link prediction and node classification. Understanding the positional context of nodes can contribute to inferring causal relationships by revealing how changes in one node may causally influence others. A data augmented GNN model called the GAUG graph data augmentation framework was proposed by Zhao et al. [25] for the purpose of improving semi-supervised node classification. In the model, neural edge predictors are used as a means of exposing GNNs to likely edges and limiting exposure to unlikely ones. By focusing on probable edges and minimizing exposure to unlikely ones, GNNs can more effectively uncover and model direct causal pathways, essential for rigorous causal analysis in complex systems.

A GCN framework called Stacked and Reconstructed Graph Convolutional Networks for Recommender Systems (STAR-GCN) by Zhang et al. [4] stacked GCN encoder-decoders for learning node representations. They used intermediate supervision and reconstructed masked input node embeddings to generate embeddings for new nodes. In the context of causality studies, such methodologies are beneficial as they facilitate precise node representation by capturing relationships and attributes within a graph effectively. Graph Isomorphism Network (GIN) was used for drug-drug interaction (DDI) predictions with DDIGIN proposed by Wang et al. [26], with the model using Node2Vec for obtaining initial representations. These representations are then optimized by aggregation of first-order neighbouring information from graphs. The GIN framework is expected to improve the expressive power of representations, aiding in the identification and understanding of causal pathways and relationships within complex networks in causal studies.

A temporal GCN called T-GCN was proposed by Zhao et al. [27] for traffic prediction which used GCN for learning spatial dependencies and GRU for capturing temporal dependencies of traffic networks, with nodes representing roads and edges representing road connections. Similarly, a GAT-based spatio temporal framework called ST-GAT [28] employed attention mechanism for traffic speed prediction. The model incorporated individual spatial and temporal dependencies using Individual Spatio-Temporal graph (IST-graph) and Spatio-Temporal point (ST-point) embedding. A self-attention mechanism is employed to learn patterns hidden in IST-dependencies among these ST-points for accurate embeddings. These models address temporal and spatio-temporal dependencies in graphs, which are essential in causal studies for understanding the evolution of changes over time or space.

2.2 Classification tasks

Classification tasks in the graph domain remains a significant challenge in the field of machine learning and the emergence of GNNs has advanced the ability to learn graph representations

and manage large-scale datasets. These tasks include node-level and graph-level classification. In graph-level classification, the goal is to predict the class label of the entire graph, whereas node-level classification predicts labels for individual nodes. For an input graph $G = (V, E)$, where V and E are the sets of nodes and edges respectively, the output is a class label y for the graph. In both node-level and graph-level classification, GNNs compute node embeddings by iteratively updating them through neighborhood aggregation.

In node-level classification, the embeddings of each node are updated layer by layer by aggregating information from its neighbours. The embedding update at layer $l + 1$ and the classification step are shown in equation Eq. 1 [29]. Here, $\mathcal{N}(v)$ is the set of neighboring nodes of node v, $AGGREGATE$ is an aggregation function (such as mean, sum, or max pooling) applied to the neighboring node embeddings, $W^{(l)}$ is the weight matrix for layer l, $b^{(l)}$ is the bias term for layer l and σ is an activation function. Once the node embeddings are computed, the final embedding for each node is passed through a classification layer to predict the label y_v for that specific node.

$$\begin{aligned} h_v^{(l+1)} &= \sigma\left(W^{(l)} \cdot \text{AGGREGATE}\left(\{h_u^{(l)} : u \in \mathcal{N}(v)\}\right) + b^{(l)}\right) \\ y_v &= \text{softmax}(W_c h_v^{(L)} + b_c) \end{aligned} \tag{1}$$

In graph-level classification, node embeddings are computed in the same way as for node-level classification. Once these embeddings are obtained, they are aggregated into a single graph-level embedding h_G, typically using pooling functions such as mean or sum pooling, to predict the class label y of the entire graph as shown in the equation Eq. 2 [29].

$$\begin{aligned} h_G &= \text{POOLING}\left(\{h_v^{(L)} : v \in V\}\right) \\ y &= \text{softmax}(W_c h_G + b_c) \end{aligned} \tag{2}$$

DEMO-Net, a degree-specific GNN framework designed by Wu et al. [30] for node and graph classification, integrates structure-aware neighbourhoods and a degree-aware framework for classification tasks. It is inspired from the Weisfeiler-Lehman graph isomorphism test to identify 1-hop neighbourhood structures. This capability is beneficial in causality studies as it enhances understanding of how network structures influence causal pathways and relationships. In their research, Maurya et al. [31] investigated the node feature aggregation process in node classification to develop the Feature Selection Graph Neural Network (FSGNN), which aims to extract relevant features. The study revealed the presence of less informative features that can adversely affect prediction performance, prompting enhancements in FSGNN to focus on learning the most relevant features. This approach is advantageous for causality studies as it improves the precision in identifying causal features and relationships within complex networks. Wang et al. [32] proposed the minority-weighted GNN (mGNN) for extracting information from imbalanced data, particularly in the context of social network analysis. This approach addresses the challenge of imbalanced classification, focusing specifically on node classification. Its importance in causality studies lies in ensuring that the impact of minority groups on causal pathways is adequately recognized and addressed.

The problem of unattributed node classification was researched by Sun et al. [33], who proposed a generalized equivariance property and a Preferential Labeling technique for addressing this issue. The former permits additional auto-isomorphic permutation and the latter technique achieves the generalized equivariance property asymptotically. This framework is especially beneficial for addressing practical challenges in anonymized social networks and significantly improves the accuracy of identifying causal relationships in these networks and

other complex systems. A GIN-based model called Dynamic Multi-Task Graph Isomorphism Network (DMT-GIN) proposed by Wang et al. [34] transformed fMRI images into brain network structures for classification of Alzheimer's disease. DMT-GIN integrated an attention mechanism to capture node features and graph structural information, thereby advancing the comprehension of how neural network configurations relate to disease progression and aiding causal inference in neuroscience research.

2.2.1 Significance, examples and applications of graph classification

Graph classification plays a crucial role in analyzing relational and interconnected data, such as social networks, molecular structures and recommendation systems. Unlike grid-based data formats such as images or sequences, graph-structured data captures complex dependencies and interactions between entities, making it ideal for tasks where relationships are crucial. GNNs excel at learning from these relational structures, providing a powerful approach for tasks such as predicting protein functions, detecting fraud in social networks and recommending personalized content-areas where traditional machine learning models often face limitations.

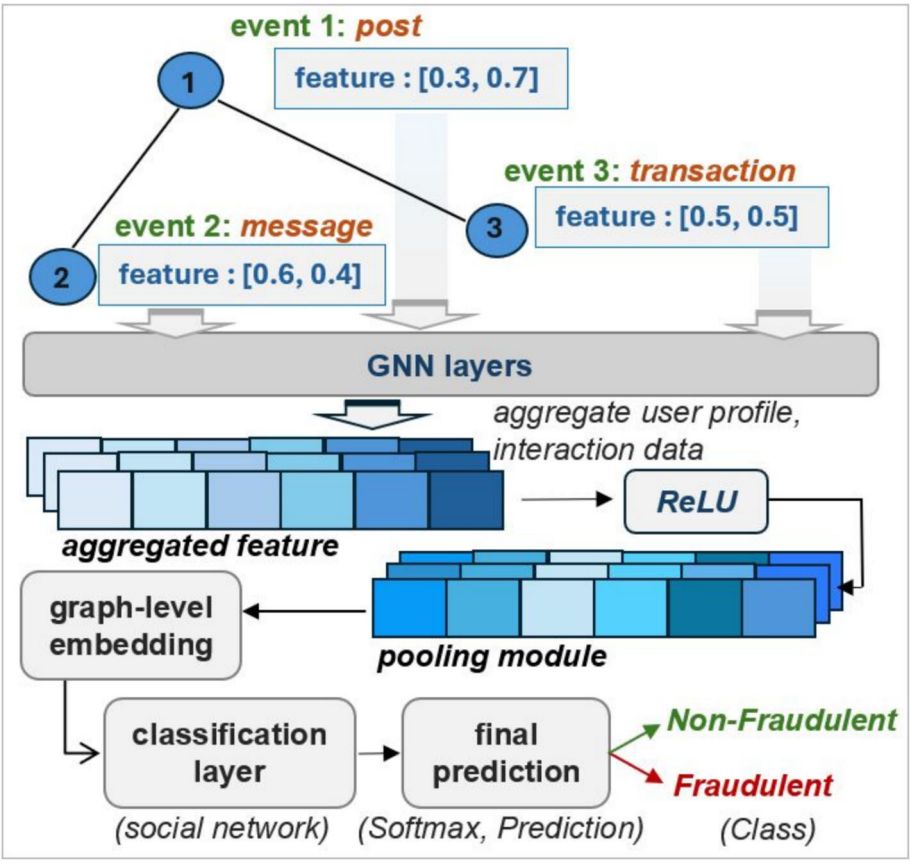

Figure 3 Graph Classification of User Behavior in Social Networks

Figure 3 depicts the classification problem of determining whether a user's behavior in a social network is fraudulent or non-fraudulent based on their interactions and activities. Each user is represented as a node, and their behavior is characterized by features such as activity type (e.g., posts, messages, transactions). The graph structure captures the relationships between users, where edges represent interactions or events between them. In this context, the model's task is to classify whether a user's behavior is suspicious (fraudulent) or typical (non-fraudulent). This is achieved by aggregating the user's own features and those of their neighbours (users they interact with) through GNNs, which allows the model to learn patterns of behaviour within the network. After aggregating and processing the features, the final prediction is made using a classification layer, where the output is a probability distribution (using Softmax), and the model predicts the most likely class: *fraudulent* or *non-fraudulent*.

Graph classification has practical applications across various domains, including chemistry, bioinformatics, social networks and recommendation systems. These applications leverage graph structures to uncover complex patterns and relationships that other methods may overlook. In *chemistry and drug discovery*, graph classification is used to predict molecular properties, such as toxicity or bioactivity, by representing molecules as graphs with atoms as nodes and bonds as edges [35]. This aids in identifying potential drug candidates and accelerates the drug discovery process. In *bioinformatics*, graph classification analyzes protein-protein interaction (PPI) networks, where proteins are nodes and interactions are edges [36]. It helps predict protein functions, classify diseases, and reveal insights into complex biological processes by examining the structure of the interaction network. In *social networks*, graph classification is applied to analyze user interactions [37], represented as edges between nodes (users). This facilitates community detection, fraud detection and user behavior prediction, improving the understanding of social dynamics and enhancing targeted marketing or content recommendations. In *recommendation systems*, graph classification models user-item interactions as bipartite graphs, with users and items as nodes and interactions (such as ratings or clicks) as edges [38]. This enables the prediction of user preferences and supports personalized recommendations, boosting user engagement and experience.

2.3 Causality

Causality [39] involves understanding how different elements in data interact in terms of cause and effect. Understanding these relationships is crucial as they provide insights into how changes in one variable influence changes in another, going beyond mere correlations to uncover the mechanisms driving observed patterns. Current GNN methods often rely on correlation-based learning, which can miss important causal relationships that drive data dynamics, particularly in complex datasets where correlations alone may be insufficient for meaningful analysis. This reliance on correlation limits traditional GNNs by preventing them from capturing the true cause-effect relationships, reducing their ability to make accurate predictions or uncover underlying patterns [40]. By identifying causal relationships, researchers can isolate the factors that directly influence outcomes, making causality an effective tool for feature selection [41]. Thus, integrating causal features into GNNs is crucial, as it not only helps models distinguish between correlation and causality but also leads to a more robust and interpretable learning process. Recent research has increasingly focused on leveraging causal inference to extract relevant features effectively, which improves both the accuracy and interpretability of machine learning models while deepening one's understanding of the underlying data dynamics.

Yu et al. [41] developed a causality-based feature selection package called CausalFS encompassing the most representative algorithms in this domain. The methods included

constraint-based algorithms and score-based approaches with Markov Boundary (MB) learning in different scenarios such as simultaneous MB learning, Divide-and-conquer MB learning and MB learning with relaxed assumptions. Causal representation learning [42] involves discovering causal variables from raw observations and is a challenging task in causal learning. The function approximation capabilities of GNNs serve to model nonlinear causal relations in large-scale graph data. Moreover, causality plays a crucial role in improving feature selection and hence is critical for GNN classification tasks. Causal models also provide insights into the underlying relationships in graph structures and hence contribute to model interpretability. A causal attention learning model called CAL was proposed by Sui et al. [3] for graph classification. The model discovers causal patterns in data through attention mechanism and employs an attention-based GNN for this purpose. A similar approach was used by Wang et al. [43] to build Causal-Trivial Attention Graph Neural Network (CTA-GNN) for discovering causality patterns by diminishing confounding effects of shortcut features. CTA-GNN was employed in fault diagnosis of complex industrial processes, wherein the industrial system entities were modeled as nodes, with their interactions represented as edges. Mutual information (MI) may also be useful for deriving causal relationships from graphs. For instance, high MI can indicate a potential causal relationship, subject to other influential factors in the data. Unsupervised Hierarchical Graph Representation (UHGR) was proposed by Ding et al. [1] for classification tasks using MI. The model was based on MI maximization between global and local parts for learning structural information. Di et al. [2] also used MI maximization for classification tasks and link prediction tasks using GNNs. The authors accomplished MI maximization by neighbourhood enlargement in GNN aggregation. They further verified the model's reliability through experiments on datasets from diverse domains.

3 Research design

An empirical study was conducted with nine representative models employed in graph classification tasks and the details are discussed in this section. These models were chosen through a thorough examination of current literature, considering their potential for future exploration in the field of causality-oriented GNNs. The outcomes of the empirical investigation are anticipated to probe the applicability of the algorithms across diverse settings and domains, aiming to illustrate both the strengths and constraints of the chosen models. Expanding on these findings, researchers can explore and develop innovative algorithms for tasks involving causality using GNNs.

3.1 Research questions

Our study endeavours to investigate prominent Graph Neural Network algorithms used in classification, aiming to identify architectures with substantial potential for integrating causality. The research questions are carefully formulated to specifically examine the performance of GNNs in classification tasks, with a goal of improving generalizability and accuracy rates, and understanding the role of hyperparameter tuning. The particular focus lies on assessing the potential improvements in classification performance when causality is incorporated. Addressing these research questions would provide future researchers with opportunities to identify mechanisms for extracting causal relationships from data, subsequently utilizing them to build resilient models. The research questions (RQ) and the corresponding hypotheses (H) explored in this study are as follows:

- **RQ1.** Which GNN architectures consistently exhibit the highest classification performance across all datasets?
  H. The causality-enabled *GAT-CAL* architecture with its attention mechanism is expected to demonstrate superior classification performance compared to other architectures, specifically the non-GAT models.
- **RQ2.** How do the baseline GNN architectures compare with causality-enhanced GNN models in terms of performance and generalizability across domains?
  H. Although baseline GNN models such as GCN may exhibit better computational performance, causality-enhanced GNN models are expected to demonstrate meaningful classification performance and generalizability. This is attributed to their ability to integrate causal features.
- **RQ3.** Are causality-enabled GNN architectures sensitive to hyperparameters and can hyperparameter tuning improve their performance?
  H. Hyperparameter tuning is expected to hold the potential to improve causal classification performance, particularly for architectures sensitive to hyperparameter changes.

3.2 Datasets

This section describes the datasets utilized for experimental studies in this research. The datasets are selected from three distinct domains: bio-chemical, citation networks and social networks, with the purpose of generalizing the research framework, rendering it adaptable to various domains. These datasets are publicly available and are easily accessible. Moreover, these datasets have been widely used in similar studies, making them well-suited for reproducibility and comparability with similar models. The summary of the datasets are given in Table 1.

3.2.1 Bio-Chemical datasets

- NCI1 [44] is a cheminformatics dataset where each graph represents a chemical compound. The nodes represent atoms in a molecule and the edges represent bonds between atoms.
- Proteins [45] is a proteins dataset with nodes representing amino acids and belong to two classes, enzymes or non- enzymes. Two nodes are connected with edges if they are less than 6 angstrom in distance.

Table 1 Summary of datasets used in the study

Dataset	Domain	# graphs	Avg. nodes	# nodes (1^{st} graph)	Avg. edges	# Edges (1^{st} graph)	# class
Cora	Citation	1	-	2708	-	10556	7
Citeseer	Citation	1	-	3327	-	9104	6
NCI1	Bio-Chemical	4110	29.87	21	32.30	42	2
Proteins	Bio-Chemical	1113	39.06	42	72.82	162	2
Mutag	Bio-Chemical	188	17.93	17	19.79	38	2
IMDB-B	Social	1000	19.77	20	96.53	146	2
REDDIT-B	Social	2000	429.63	218	497.75	480	2

The citation datasets are used for node classification tasks. The bio-chemical and social datasets are used for graph classification tasks, where number of nodes and edges are for the 1^{st} graph

- Mutag [46] dataset consists of nitroaromatic compounds, with nodes representing atoms and the edges representing bonds between atoms. The dataset has two classes according to their mutagenic effect on a bacterium. The main limitation of this dataset is its modest size, although this aspect can be advantageous for conducting preliminary experiments with novel algorithms.

These datasets, which are made available through the TUDataset package, are commonly used as benchmarks in the graph classification domain, facilitating the comparison of different graph-based algorithms. Moreover, they consist of molecular structures derived from real-world compounds, enhancing its applicability to practical scenarios. Proteins and NCI1 have been used by Ding et al. [1], with all three datasets employed for classification tasks in studies including [3, 23, 47, 48].

3.2.2 Citation network datasets

Cora [49] and Citeseer [50] are citation network datasets containing scientific publications categorised into seven and six classes respectively. Nodes and edges in these datasets represent paper and citation relationships respectively. These benchmark datasets are extensively used in research and represent real-world citation networks, commonly employed for classification tasks across multiple studies [1, 51, 52]. Their moderate size facilitates extensive experimentation with graph-based methods. However, these datasets lack temporal information and require enhanced feature representations, which are their primary limitations.

3.2.3 Social network datasets

- IMDB-BINARY (IMDB-B) [53] is a movie collaboration dataset consisting of movie information from IMDB. The nodes in a graph represent actors/actresses, with an edge between nodes for actors from same movie. The dataset contains collaboration graphs based on genres, with ego-networks for each individual. The graph is labelled based on movie genres viz. Romance or Action. This dataset has been used for classification tasks by [3]. However, due to its limitation to the movie genre, there is a need to investigate the dataset's generalizability to other domains.
- REDDIT-BINARY (REDDIT-B) [53] consists of data related to online discussion threads, where nodes represent users and an edge between two nodes denote that a correspondence (comments) has been made between these two users. The graph is labelled based on whether it belongs to a question/answer-based or a discussion-based community.

Few advantages of these datasets are their real-world relevance and the availability of a substantial amount of labelled data for training. Furthermore, these two datasets are widely utilized in graph-based research, facilitating comparative studies [23, 54, 55].

3.3 Experiment design

The experiments are conducted as follows: The graph datasets are split into training and testing sets using KFold to generate indices for data splitting. A 5-fold cross validation is employed to evaluate the performance and generalization capability of the models. For each architecture, such as GCN or GAT, a GNN model is created, incorporating a final classification layer. These models are then trained and evaluated using standard performance metrics,

including accuracy, precision, recall and F-scores. Additionally, a sensitivity study is carried out to examine how variations in hyperparameters impact model performance. These metrics are selected based on their ability to assess different aspects of the model's classification performance. While accuracy delivers a thorough evaluation of a model's correctness across all classes, its efficacy diminishes in imbalanced datasets. In such cases, precision, recall, and F-scores become crucial for a more precise assessment. Precision quantifies the accuracy of positive predictions, while recall measures the model's capacity to capture all positive instances. The F1-score balances precision and recall into a single metric.

F-score is a highly effective metric for evaluating *Cora* and *Citeseer* (node classification) and *IMDB-B, Reddit-B, Proteins*, and *NCI1* (graph classification) because these datasets, while having some degree of imbalance, do not have extreme class distributions that would significantly affect performance. In these cases, F-score effectively balances precision and recall, providing a fair evaluation of the model's ability to perform across both majority and minority classes. For *Mutag*, despite its moderate class imbalance, F-score remains highly effective as it balances precision and recall, which is crucial for evaluating imbalanced datasets. The smaller size of the dataset means that each instance has a larger impact on the evaluation, but the primary benefit of F-score is that it ensures the model does not overly favour the majority class (non-mutagenic), keeping a strong focus on the minority class (mutagenic).

3.4 Models

The most representative and the most extensively employed models in GNN-based studies were investigated and subsequently, nine relevant models were selected for this study as follows:

- GCN [52]: The research primarily focuses on graph neural networks for classification tasks and hence the most commonly used GNN architecture viz. GCN is used as a baseline model. Comparing the fundamental structure of GCN with more advanced versions of GNNs would aid in determining whether enhanced GNN variants offer a substantial improvement in performance. GCN utilizes the convolutional mechanism to propagate information across a graph by aggregating features from neighbouring nodes. GCN is highly effective in capturing local graph structures.
- GAT [51]: GAT is a state-of-the-art GNN model that employs an attention mechanism, allowing nodes to selectively prioritize different neighbors when aggregating information. In contrast to GCN, GAT excels in capturing long-range dependencies, making it well-suited for tasks that necessitate the capture of both local and global patterns within the graph. With its attention mechanism, GAT assigns adaptive weights to the neighbors of each node during information aggregation. GAT can dynamically adjust neighbourhood importance based on learned attention weights and hence is advantageous in graph classification tasks.
- GIN [23]: GIN is permutation invariant and hence can address the over-smoothing problems encountered by GCN. GIN is most typical for processing isomorphic graphs and is researched for its flexibility in capturing complex graph patterns. GIN consists of isomorphism layers that are invariant to node ordering. The node features are aggregated using summation allowing the model to capture relationships within a graph. GIN, characterized by its isomorphic and permutation invariant properties, has considerable potential in effectively capturing complex structural patterns present within graphs. Therefore, it

is crucial to explore the model's capabilities in the context of graph classification rather than focusing primarily on node classification.

- GraphSAGE [21]: GraphSAGE involves node sampling followed by aggregation of the sampled node features using an aggregation function such as mean aggregation or pooling aggregation. With its localized sampling and aggregation approach, the model can efficiently handle large graphs. Moreover, GraphSAGE exhibits insensitivity to node ordering, making it a key contender as a model for tasks related to graph classification. GraphSAGE is investigated for its ability to capture more localized and diverse information from the graph. GraphSAGE is also the state-of-the-art model in efficiently adapting to diverse graph types.
- GCN-CAL, GAT-CAL, GIN-CAL [3]: The CAL framework was selected for its use of causality in GNN classification. The framework introduced causal attention learning to GCN, GAT and GIN architectures for enabling causal classification. The model primarily used a GNN-based encoder for obtaining node representations, followed by utilization of two MLPs for estimating edge and node-level attention scores. Leveraging attention scores is a fundamental approach for extracting causality through the use of GNN architectures. Therefore, this model was investigated for its performance in tasks related to causal classification.
- UHGR-GAT, UHGR-GCN [1]: The UHGR framework uses an encoder for constructing node representation, followed by graph pooling. The UHGR models use mutual information and hence we investigate this method for its ability to derive causality from graph data. A discriminator module is used for training the encoder for mutual information maximization. The learned graph representations are employed for node and graph classification tasks. In this work, two UHGR variants, using GAT and GCN, are studied for their potential for mutual information-based classification.

The experiments on the selected models were carried out using the seven datasets described in Section 3.2. Further elaboration on the selected models is provided here. Sui et al. [3] proposed Causal Attention Learning (CAL) with mitigation of confounding effects using softmask estimation from attention scores. The graph is decomposed to causal and trivial attended graphs with two GNN layers. The authors proposed disentanglement of causal and trivial features, with GNNs filtering shortcut patterns for capturing causal features. The CAL framework was experimented with GCN, GAT and GIN architectures and for these three models, the settings used by Sui et al. [3] were mostly reproduced for experimentation. A 5-fold cross-validation is conducted, deviating from the 10-fold cross-validation used in the initial study. We provide the code at https://github.com/sj20000/EmpiricalStudy for reference purposes. Although the original study trained the models for 100 epochs, this study trains them for 20 epochs to ensure consistency across all models in the investigation. The same settings were also applied to the GCN, GAT, GIN and GraphSAGE models. Evaluation was performed on two tasks viz. node classification using citation datasets and graph classification using the bio-chemical and social datasets. The evaluation metrics used were classification accuracy, Precision/Recall scores and F1-Scores.

4 Empirical study outcomes

The test accuracy results of these studies are summarised in Table 2. The highest accuracy rate for each dataset among all models is shown in bold, with the second highest highlighted in italics. The results reported for the citation datasets pertain to the node classification task,

Table 2 Test Accuracy (%) of classification tasks (the highest scores are bolded, and the second highest scores are italicized)

Model	NCI1	Proteins	Mutag	Cora	Citeseer	IMDB-B	Reddit-B
GCN [52]	*80.68*	75.47	84.10	79.74	65.30	**73.60**	*91.25*
GAT [51]	79.54	75.02	**89.42**	*83.03*	**68.41**	72.00	90.85
GIN [23]	80.44	75.29	85.14	80.82	61.80	72.50	88.75
GraphSAGE [21]	74.21	**75.83**	83.49	76.55	61.66	73.00	77.20
GCN-CAL [3]	**81.07**	75.29	78.68	80.21	61.74	72.70	91.15
GAT-CAL [3]	80.49	*75.56*	*86.26*	**83.16**	*65.61*	*73.20*	**91.30**
GIN-CAL [3]	80.36	73.76	71.86	80.53	59.97	72.60	89.65
UHGR-GAT [1]	59.80	70.09	77.77	67.55	53.30	55.00	64.66
UHGR-GCN [1]	61.07	67.92	74.11	62.65	50.25	56.20	64.77

whereas those for the bio-chemical and social network datasets are for the graph classification task. The precision/recall results for GCN-CAL, GAT-CAL, GCN, GAT and GraphSAGE are plotted for the *Mutag*, *Cora* datasets and *IMDB-B* datasets as shown in Figure 5. The F1-scores for GCN, GAT, GraphSAGE, GCN-CAL, GAT-CAL, UHGR-GCN and UHGR-GAT models on all the datasets are shown in Figure 4.

4.1 Experimental results and discussions

On analysing the classification performance results based on accuracy scores, as presented in Table 2, the GAT architecture demonstrates superior performance compared to the other base models across most datasets. The classification framework augmented with causal elements, CAL [3] demonstrated the model's effectiveness with the integration of causality. This was specifically observed with the GAT-CAL model, for which the performance was higher compared to other architectures, except for the NCI1 dataset. For the UHGR models [1], experimentation on the seven datasets demonstrated significant decline in performance for the social network datasets and the NCI1 dataset, when compared to the other models as shown in Table 2. The precision and recall scores are mostly evenly distributed for the Mutag and Cora datasets. However, the scores for the IMDB-B dataset exhibit a substantial difference,

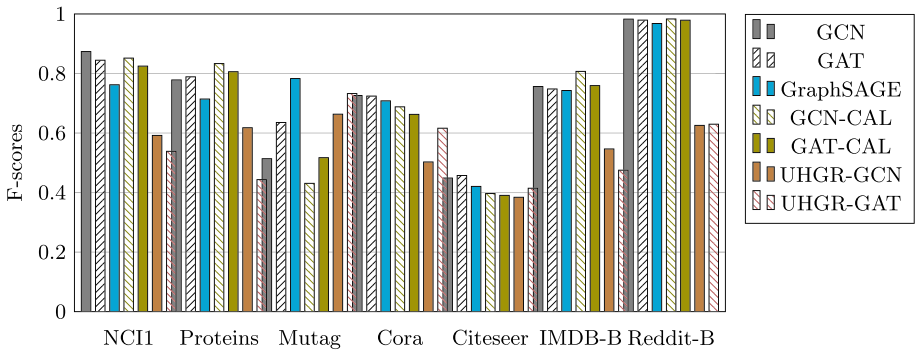

Figure 4 F-scores for the key models on datasets. GCN, GAT and GraphSAGE are foundational GNN models. GCN-CAL and GAT-CAL integrate causality into GNNs, while UHGR-GCN and UHGR-GAT leverage mutual information

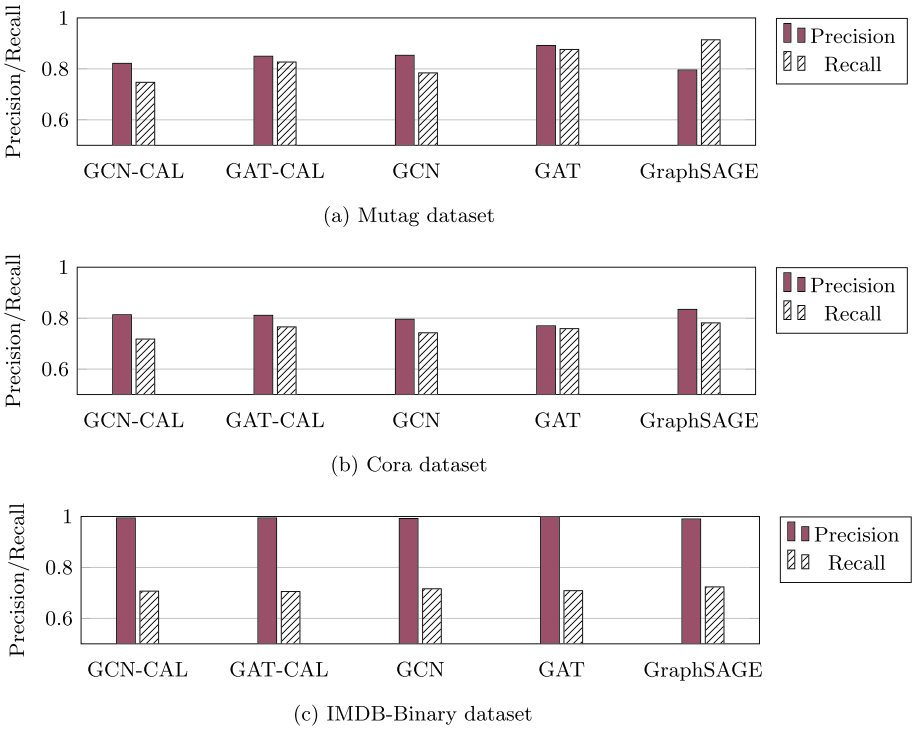

Figure 5 Precision-recall scores

likely due to a model that is overly conservative in predicting positives, resulting in fewer but more accurate positive predictions (Fig. 5).

Upon analyzing the F1-scores, as presented in Figure 4, it is evident that performance varies across algorithms on the Mutag dataset. Apart from this, all the models demonstrate effective generalization across all datasets, with the exception of the UHGR model on NCI1 and the social network datasets. The UHGR-GAT model exhibits a decline in performance on the protein dataset as well. Overall, it is observed from the F1-scores that GCN and GAT performs better than the other models on node classification tasks. For graph classification tasks, the CAL framework presents a promising approach and is mostly generalizable across datasets, with a dip in performance with the Mutag dataset. The performance of the GraphSAGE architecture has been notable on the Mutag dataset, whereas all other models exhibit lower performance for this dataset. While the CAL framework has generally demonstrated comparable performance to other models, except on the Mutag dataset, its failure to achieve significantly higher results suggests a limitation in causal modeling. Similarly, though the UHGR framework has demonstrated results on par with the other models for the Mutag and citation datasets, this model has also fallen short of achieving exceptional results. This suggests that while MI significantly contributes to the classification performance, further improvements are necessary to enhance the model. Here, we discuss some of the advantages and disadvantages of these models.

- GCN [52] has higher computational efficiency and is particularly suitable when local neighbourhood information is relevant for the problem. This characteristic is notably reflected in its high F-score values for node classification.

- GAT [51] is better suited for inductive settings and excels at capturing intricate relationships through its attention mechanism. However, these advantages come with a few drawbacks, including the computational complexity and hyperparameter sensitivity of the GAT architecture. The sensitivity to L2 regularization and learning rates is particularly evident in its performance on the Mutag and citation datasets respectively.
- GraphSAGE [21] is a framework that is suitable for large-scale graphs. A limitation of this architecture is that it relies on sampling a fixed-size neighbourhood for each node, and hence cannot capture distant information beyond this boundary. It is noted that GraphSAGE demonstrates greater robustness in capturing social networks compared to molecular graphs, which could be attributed to its limited capacity in capturing complex molecular structures.
- GIN [23] is more adept at capturing global graph structures as compared to other architectures. However, GIN does not incorporate node features during aggregation, leading to loss of significant node information. On the other hand, GCN offers a more comprehensive node representation by leveraging node features in the aggregation function. This is evident in the consistently strong performance of GIN in graph classification tasks, as indicated by the high F1-scores and accuracy rates.
- The CAL [3] framework adopts a causal classification approach utilizing the attention mechanism. While this approach leverages weighted information to infer causality and capture dependencies among causal features, handling long-range causality may pose challenges for this model. With the exception of Mutag, all three CAL models demonstrate strong performance across all datasets.
- The UHGR [1] framework utilized mutual information maximization (MIM) for classification tasks. MIM has the capability to learn meaningful representations by capturing both local and global data patterns. The UHGR model demonstrated positive performance outcomes in node classification tasks, emphasizing the significance of mutual information in understanding dependencies among individual nodes. However, the graph classification performance results do not indicate significant improvements for base models augmented with MIM. It is also essential to recognize that MIM alone does not inherently produce causal representations and would benefit from integrating mechanisms that specifically address dependencies, such as causality-focused feature extraction techniques.

In summary, it is important to emphasize that these frameworks do not inherently capture causal elements, necessitating additional steps for the integration of causality in classification tasks.

4.2 Scalability and generalizability

The scalability and generalizability of the models can be evaluated by examining their performance on datasets of varying sizes and complexities. Strong results on datasets such as *Proteins* and *IMDB-B* (both with around 1000 graphs) suggest that the models are capable of handling moderately sized graphs with complex relational structures. Their success on *Reddit-B* (with 2000 graphs) further demonstrates their ability to scale to larger datasets, highlighting their versatility across domains.

However, the *NCI1* dataset, with over 4000 graphs, presents a greater scalability challenge. While the models perform reasonably well on this dataset, they may struggle with larger or more complex datasets due to increased computational demands or the need for advanced optimization techniques. Since performance on *NCI1* improves mainly through learning rate adjustments, optimizing hyperparameters will be essential for enhancing scalability on larger

datasets. In terms of generalizability, the framework demonstrates flexibility across domains such as biological, entertainment and social networks, suggesting potential for applications including medicine and recommendation systems. However, its reliance on specific hyperparameters for certain datasets, along with varying performance on *Cora, NCI1, Mutag* and *Citeseer*, indicates the need for further improvements to enhance robustness. Addressing these scalability and generalizability challenges will strengthen the models' applicability to real-world tasks involving large-scale, complex graph data.

4.3 Sensitivity analysis

A sensitivity study has been performed for all the models based on learning rates and L2 regularization or weight decay. The learning rate (LR) and weight decay (WD) are *1e-3* and 0 in the original studies. The results for varying learning rates with WD=0 are shown in Table 3. The results for varying weight decay rates for LR=*1e-3* are shown in Table 4. The top F-score for each dataset among all models is shown in bold, while the second highest is italicized. The results of the analysis are summarised as follows:

- From the experimental findings presented in Table 3, it is evident that different learning rates result in diverse performances for all models across the entirety of datasets. A pronounced variation in F-scores is noticeable for all models, except the UHGR models, across the Mutag, Cora, and Citeseer datasets. This suggests that the impact of the learning rate is predominantly influenced by dataset characteristics, as opposed to the model architecture.
- The impact of varying the weight decay rate as shown in Table 4, is notably pronounced for the two UHGR models, with significant variations observed across F-Scores, except possibly for UHGR-GCN on the Cora dataset. The CAL framework displays variances primarily on the citation datasets, suggesting a potential sensitivity of the model to node classification tasks concerning WD. The impact of varying WD on other models is marginal, except for GCN and GAT on the Mutag dataset, where noticeable effects are observed. This suggests the dataset's sensitivity to the regularization mechanism.

4.4 Analysis of evaluation result

Drawing from the evaluation results, the research questions are addressed as follows:

- **RQ1.** Based on the findings presented in Table 2, it is evident that the GCN-CAL model achieves approximately 1% higher accuracy than the GAT-CAL model on the NCI1 dataset. Similarly, GraphSAGE achieves slightly less than 1% higher accuracy than the GAT-CAL model on the Proteins dataset, while GCN exhibits less than 1% higher accuracy than the GAT-CAL model on the IMDB-B dataset. On the Mutag and Citeseer datasets, the GAT model outperforms the GAT-CAL model by an additional 3% accuracy each. Overall, the GAT-CAL architecture consistently delivers strong performance across all datasets.
- **RQ2.** Based on the F-scores presented in Figure 4, it is observed that the causal framework achieves high F-scores for the proteins and social network datasets, with the GAT-CAL model performing the best among them. For the remaining datasets, the GCN, GIN and GAT frameworks show superior performance. Based on the results from varying learning rates as shown in Table 3, it is confirmed that the causality-based CAL framework consistently achieves high F-scores across different learning rates for the Proteins dataset and

Table 3 Classification F-Scores with time (secs) for varying Learning Rates (the highest scores are bolded, and the second highest scores are italicized)

Model	LR	NCI1		Proteins		Mutag		Cora		Citeseer		IMDB-B		REDDIT-B	
		time	f-score	time	f-score	time	f-score	time	f-score	time	f-score	time	f-score	time	f-score
GCN [52]	1e-2	41.8	0.8117	17.9	0.7141	7.9	0.7829	51.7	0.6814	43.1	0.5067	14.2	0.7904	51.1	0.9616
	1e-3	45.1	**0.8737**	17.4	0.7783	7.6	0.5138	51.4	0.7261	49.7	0.4488	14.8	0.7562	54.6	0.9826
	1e-4	44.6	0.8408	17.7	0.8289	7.5	0.3875	53.0	0.4598	48.6	0.3476	14.8	0.8246	56.3	0.9905
	1e-5	43.7	0.7228	17.7	0.7026	7.6	0.3770	51.4	0.2189	53.5	0.1779	14.8	0.7583	47.4	0.9628
GAT [51]	1e-2	48.0	0.8181	19.6	0.7205	8.1	*0.7835*	64.8	0.7100	48.3	**0.5802**	14.6	0.7676	64.6	0.9508
	1e-3	47.3	0.8446	19.1	0.7888	8.2	0.6355	54.9	0.7241	53.4	0.4574	15.4	0.7480	65.8	0.9793
	1e-4	49.1	0.7837	18.9	0.8544	8.2	0.5316	54.0	0.4488	52.0	0.2538	15.7	0.7735	70.5	0.9967
	1e-5	47.6	0.7018	19.3	0.8247	8.1	0.4763	53.8	0.2430	52.08	0.1511	15.7	0.7523	67.4	0.9241
GIN [23]	1e-2	39.7	0.7246	16.2	0.6833	7.2	0.6376	33.3	0.5789	30.3	0.3058	49.1	0.6456	49.5	0.8820
	1e-3	39.9	0.8507	16.0	0.7800	7.1	0.6929	30.8	0.6953	30.6	0.4258	18.1	0.6609	49.2	0.9947
	1e-4	40.3	0.8486	16.2	0.8629	7.1	0.4843	36.9	0.4234	30.7	0.3141	14.5	**0.8482**	51.3	0.9970
	1e-5	39.0	0.7891	16.2	0.8754	7.2	0.4112	43.0	0.2199	30.2	0.1617	12.8	0.7468	53.9	0.9864
Graph-	1e-2	42.3	0.7610	16.5	0.7631	7.8	0.8011	37.8	0.6579	37.3	0.4661	19.3	0.7982	49.9	0.9615
SAGE [21]	1e-3	43.8	0.7621	17.3	0.7142	7.9	0.7830	37.9	0.7081	37.0	0.4207	18.5	0.7429	71.7	0.9679
	1e-4	45.1	0.7618	16.8	0.7723	7.8	0.6131	39.1	0.3051	35.4	0.2352	19.7	0.6979	48.5	0.9867
	1e-5	43.4	0.8637	17.5	0.7790	7.9	0.4720	38.2	0.2208	36.5	0.1905	30.9	0.6585	61.5	0.8499

Table 3 continued

Model	LR	NCI1		Proteins		Mutag		Cora		Citeseer		IMDB-B		REDDIT-B	
		time	f-score	time	f-score	time	f-score	time	f-score	time	f-score	time	f-score	time	f-score
GCN-CAL	1e-2	57.4	0.8331	21.7	0.7640	9.1	0.7453	72.2	*0.7327*	70.6	0.4944	18.9	0.8069	105.4	0.9688
[3]	1e-3	55.9	*0.8517*	21.4	0.8333	9.5	0.4310	74.7	0.6880	76.5	0.3967	19.1	0.8071	104.9	0.9832
	1e-4	54.4	0.7732	21.5	0.8803	9.5	0.5061	71.8	0.3337	74.8	0.2294	18.9	*0.8247*	97.8	*0.9980*
	1e-5	59.2	0.6762	21.7	0.7096	9.4	0.4896	74.2	0.1955	76.0	0.1313	19.4	0.6995	97.0	0.8703
GAT-CAL	1e-2	56.9	0.8299	22.6	0.7380	9.6	0.7560	79.9	**0.7364**	71.1	*0.5182*	20.6	0.8054	115.9	0.9453
[3]	1e-3	64.0	0.8248	22.3	0.8060	10.1	0.5172	79.0	0.6627	70.3	0.3906	20.5	0.7593	120.7	0.9790
	1e-4	60.9	0.7261	22.2	*0.8976*	10.3	0.3713	79.7	0.2726	71.6	0.1856	20.1	0.7396	124.5	0.9912
	1e-5	57.1	0.6790	22.8	0.8573	10.0	0.4282	91.9	0.2356	70.5	0.1603	19.3	0.6120	127.8	0.9819
GIN-CAL	1e-2	55.6	0.7389	20.9	0.7675	8.8	0.6109	56.4	0.3470	59.3	0.2837	18.8	0.6336	99.3	0.9002
[3]	1e-3	50.0	0.8130	20.1	0.8492	8.9	0.2664	62.6	0.6174	61.9	0.3629	18.5	0.7051	100.0	0.9875
	1e-4	49.3	0.7713	20.1	**0.8993**	9.1	0.2846	58.9	0.3372	53.9	0.2058	17.9	0.7739	99.3	**0.9996**
	1e-5	53.8	0.7305	20.1	0.8769	9.1	0.3710	57.1	0.2765	57.4	0.1353	17.4	0.6832	101.9	0.9571
UHGR-	1e-2	1693	0.5386	493.4	0.6277	96.2	0.7425	8.7	0.5864	9.5	0.4358	418.0	0.4653	798.2	0.6257
GAT [1]	1e-3	1701	0.5386	449.6	0.4435	95.1	0.7326	8.7	0.6162	9.5	0.4146	416.8	0.4752	763.8	0.6297
	1e-4	1743	0.4851	449.4	0.5603	96.7	0.7168	8.5	0.5926	9.3	0.3830	421.4	0.5524	751.5	0.6099
	1e-5	1884	0.5702	453.6	0.6336	96.2	0.7287	8.9	0.5842	9.4	0.4624	412.6	0.4831	750.8	0.6376
UHGR-	1e-2	1672	0.5425	458.1	0.6633	94.6	0.7524	8.5	0.5224	9.0	0.4310	420.9	0.4910	758.9	0.6415
GCN [1]	1e-3	1693	0.5920	456.8	0.6178	92.1	0.6633	8.1	0.5028	8.9	0.3842	417	0.5465	765.1	0.6257
	1e-4	1764	0.5603	474.2	0.6673	94.8	0.7425	8.4	0.5308	9.5	0.4086	432.4	0.4633	746.9	0.5663
	1e-5	1749	0.6198	466.6	0.6752	93.1	**0.7940**	8.4	0.5192	9.4	0.3902	421.0	0.5168	742.7	0.5782

Table 4 Classification F-Scores with time (secs) for varying Weight Decay rates (LR=1e-3) (the highest scores are bolded, and the second highest scores are italicized)

Model	WD	NCI1		Proteins		Mutag		Cora		Citeseer		IMDB-B		REDDIT-B	
		time	f-score	time	f-score	time	f-score	time	f-score	time	f-score	time	f-score	time	f-score
GCN	1e-2	49.1	0.8662	19.1	0.7805	8.6	0.4125	50.9	*0.7551*	49.1	**0.5133**	17.2	0.7336	62.2	0.9808
[52]	1e-3	49.9	**0.8705**	18.6	0.7806	8.3	0.4805	47.5	**0.7591**	48.4	0.4925	17.4	0.7568	60.0	0.9823
	1e-4	50.7	*0.8676*	18.5	0.7808	9.0	0.4921	48.8	0.7408	49.8	0.4688	17.2	0.7618	61.4	0.9830
GAT	1e-2	52.1	0.8436	19.7	0.7906	10.1	0.5863	49.0	0.7269	49.7	0.4962	21.2	0.7429	71.3	0.9840
[51]	1e-3	53.6	0.8446	19.8	0.7945	8.4	0.6717	50.8	0.7342	49.7	0.5030	18.4	0.7492	72.1	0.9790
	1e-4	52.4	0.8380	19.2	0.7898	9.0	0.6291	50.0	0.7381	49.6	*0.5071*	18.0	0.7486	77.3	0.9730
GIN	1e-2	43.1	0.8311	16.2	0.7742	7.5	0.6002	36.6	0.6862	34.5	0.4468	14.9	0.6480	52.7	**0.9960**
[23]	1e-3	44.2	0.8498	16.6	0.7794	7.6	0.6602	34.6	0.6971	33.8	0.4660	15.0	0.6492	52.0	*0.9944*
	1e-4	45.0	0.8437	16.4	0.7797	8.1	0.6696	33.4	0.6852	34.4	0.4365	15.6	0.6496	50.3	0.9937
Graph-	1e-2	47.5	0.7447	17.8	0.7167	8.5	**0.7871**	40.6	0.72172	44.2	0.4648	16.2	0.7479	41.8	0.9738
SAGE	1e-3	48.1	0.7670	18.9	0.7179	7.8	0.7798	41.0	0.7137	40.7	0.4261	16.5	0.7452	44.4	0.9558
[21]	1e-4	49.2	0.7598	17.6	0.7243	7.9	*0.7862*	41.8	0.7222	42.5	0.4093	16.3	0.7478	44.8	0.9717
GCN-	1e-2	64.3	0.8443	22.9	0.8452	10.4	0.4034	87.2	0.7128	87.9	0.3983	22.3	0.7970	114.5	0.9855
CAL [3]	1e-3	63.4	0.8493	23.1	0.8317	10.2	0.4296	89.5	0.6891	89.3	0.3970	22.3	**0.8113**	112.9	0.9820
	1e-4	60.9	0.8472	22.4	0.8330	10.1	0.4474	81.0	0.6921	84.7	0.3804	22.7	*0.8084*	111.4	0.9832

Table 4 continued

Model	WD	NCI1		Proteins		Mutag		Cora		Citeseer		IMDB-B		REDDIT-B	
		time	f-score	time	f-score	time	f-score	time	f-score	time	f-score	time	f-score	time	f-score
GAT-	1e-2	67.2	0.8394	23.7	0.8150	10.3	0.497	79.2	0.6702	90.6	0.3683	22.8	0.7449	130.5	0.9841
CAL [3]	1e-3	69.1	0.8387	23.7	0.8027	10.2	0.5039	79.9	0.6593	94.0	0.3744	23.5	0.7562	128.2	0.9788
	1e-4	67.4	0.8445	23.7	0.8040	10.5	0.5155	93.7	0.6616	82.2	0.3840	23.7	0.7630	127.6	0.9821
GIN-	1e-2	56.1	0.8098	21.0	**0.8640**	10.6	0.2498	59.2	0.6409	57.4	0.3929	20.8	0.7592	113.6	0.9802
CAL [3]	1e-3	58.3	0.8108	21.2	0.8485	10.4	0.2458	63.3	0.6339	57.2	0.3636	21.2	0.7513	108.7	0.9860
	1e-4	56.3	0.8056	21.6	*0.8488*	9.6	0.2188	67.1	0.6251	60.7	0.3210	20.1	0.7627	111.0	0.9880
UHGR	1e-2	1708	0.5940	514.2	0.6118	107.6	0.6396	9.8	0.5774	10.7	0.4230	418.2	0.4891	765.2	0.6059
-GAT	1e-3	1719	0.5584	523.0	0.6712	111.5	0.6811	8.9	0.6052	10.7	0.4432	416.4	0.4871	758.6	0.5920
[1]	1e-4	1762	0.5445	521.4	0.6356	113.5	0.7306	9.5	0.5808	10.2	0.4506	412.5	0.6039	781.6	0.5504
UHGR	1e-2	1829	0.5900	472.6	0.6158	91.8	0.6495	9.0	0.5116	10.1	0.4064	438.9	0.5089	786.4	0.6039
-GCN	1e-3	1838	0.5861	443.1	0.6752	91.4	0.7524	8.9	0.5216	9.8	0.4268	444.2	0.5386	741.9	0.6297
[1]	1e-4	1793	0.5722	438.8	0.6495	93.2	0.6633	9.1	0.5160	9.7	0.3756	451.1	0.5069	718.2	0.6376

both social network datasets, outperforming other models in these instances. However, its performance on other bio-chemical and citation datasets indicates better F-scores only for specific learning rates, with Citeseer demonstrating improved performance in just one instance. These findings underscore the efficacy of the causal framework, particularly for graph classification tasks involving larger datasets.

- **RQ3.** The experimental results from Table 3 reveal that the performance of all models across the datasets is notably influenced by varying learning rates. A distinct decline in performance is particularly evident with the learning rate of *1e-5*, impacting most models, although exceptions include the UHGR models on select datasets, certain frameworks on the Proteins dataset, and GraphSAGE on the NCI1 dataset. In contrast, the varying weight decay rates presented in Table 4, consistently result in different outcomes specifically for the UHGR models.

4.5 Case study

A brief case study was undertaken to assess the models' adaptability to multi-class datasets in the context of graph classification. To achieve this, the multi-class variants of the IMDB and REDDIT datasets were employed. The IMDB-MULTI, REDDIT-MULTI-5K and REDDIT-MULTI-12K datasets consists of 3, 5 and 11 classes respectively. Memory constraints prevent the experimentation of UHGR models on the latter dataset.

The experimental configurations remain consistent with the previous settings in our study and the test accuracy and F-Score results are shown in Table 5. The highest accuracy and F-score for each dataset are indicated in bold, while the second highest is italicized. Upon examining these findings, the following observations are noted:

- In the IMDB-MULTI dataset, the GIN model achieves the top F1-score of 0.7460, closely followed by its causal variant. For the REDDIT datasets, both GCN and GAT models, along with their respective CAL architectures, exhibit similarly high levels of performance across both the REDDIT-MULTI-5K and REDDIT-B datasets.
- A notable observation is the marked difference in GraphSAGE performance between the REDDIT-MULTI-5K dataset, where it performs poorly, and the REDDIT-MULTI-12K dataset, where it outperforms other GNN models to achieve the highest scores. This disparity may stem from the complexities or compatibility issues of GraphSAGE's neigh-

Table 5 Test Accuracy (%), F-scores with time(secs) for graph classification tasks on multi-class datasets (the highest scores are bolded, and the second highest scores are italicized)

Model	IMDB-MULTI			REDDIT-MULTI-5K			REDDIT-MULTI-12K		
	time	accuracy	f-score	time	accuracy	f-score	time	accuracy	f-score
GCN [52]	19.6	**51.00**	0.6652	127.8	*56.49*	**0.7805**	258.9	49.25	0.7302
GAT [51]	20.4	50.73	0.6722	169.7	55.85	0.7475	351.1	*49.48*	0.7197
GIN [23]	16.2	49.33	**0.7460**	113.8	54.15	0.6912	245.4	48.17	0.7268
GraphSAGE [21]	21.6	*50.80*	*0.6882*	116.52	38.85	0.0307	197.8	34.29	**0.9247**
GCN-CAL [3]	23.1	**51.00**	0.6379	260.5	**56.57**	*0.7778*	494.3	49.00	*0.7483*
GAT-CAL [3]	25.7	49.33	0.6164	273.5	55.69	0.7613	675.1	**49.57**	0.7320
GIN-CAL [3]	21.1	48.73	0.6838	255.5	54.11	0.6343	502.1	47.29	0.7044
UHGR-GAT [1]	319.9	38.66	0.4158	935.4	31.97	0.4356	–	–	–
UHGR-GCN [1]	296.3	42.66	0.4158	898.5	37.18	0.3861	–	–	–

borhood aggregation method with the node relationships in the 5-class dataset. Across both datasets, the CAL framework consistently achieves high F-scores, highlighting the importance of causal inference in larger multi-class datasets.

- Upon careful analysis, it becomes evident that multi-class datasets generally exhibit lower performance compared to their binary counterparts, a trend particularly noticeable in the Reddit datasets. However, the causal framework consistently exhibits robust performance across all these datasets demonstrating its adaptability to multi-class scenarios. Nonetheless, further refinement in selecting causal features is required to achieve optimal results.

5 Conclusions

This paper presented an in-depth study on the application of graph neural networks for potential causal classification. Through a rigorous evaluation, we have analyzed the performance and applicability of standard GNN models, as well as those specifically designed for causality analysis. On an overall analysis of the results, it is evident that the attention-based causal model (the CAL framework) outperformed baseline GNN models in larger graph classification tasks, while the baseline models excelled in smaller node classification tasks. Our research also highlights the importance of hyperparameter tuning for improving model performance and underscores the adaptability of causal models to multi-class datasets in graph classification, suggesting their potential suitability for integration into real-world systems.

Our study offers insights into the performance of various GNN architectures and their generalizability on datasets from diverse domains. Additionally, we investigated the effectiveness of causality-based GNN frameworks in node and graph classification tasks, including a sensitivity analysis to understand the impact of hyperparameter variations on model performance. While the study extends to the application in multi-class graph classification, it reveals that reliance on attention mechanisms or mutual information estimation alone does not suffice for accurate causality inference in GNN models. This underscores the necessity for further research and the development of more sophisticated techniques to construct a robust causal GNN classification framework. Additionally, while F-score, precision and recall are the standard metrics within the scope of our work and in the context of classification, extending our work would require measuring counterfactual prediction stability, considering causal explainability using methods such as SHAP and addressing challenges related to imbalanced datasets.

Acknowledgements This work is partially supported by grants from Australian Research Council (No. DP220101360) and the SAGE Athena Swan Scholarship, UniSQ.

Author Contributions S.J, X.T, L.L. and H.X. designed the study. S.J., X.T., T.C., L.L., and H.X. conceived and planned the experiments. S.J. carried out the experiments. S.J., X.T., T.C., and L.L. contributed to the interpretation of the results. S.J. took the lead in writing the manuscript, and all authors provided critical feedback and helped shape the research, analysis and manuscript. X.T., T.C., and J.Y. contributed to supervision. X.T. and J.Y. contributed to project management.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data Availability Datasets are available in:
1) https://chrsmrrs.github.io/datasets/docs/datasets
2) https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html

Declarations

Competing Interests The authors declare that they have no known competing financial or non-financial interests that are directly or indirectly related to the work submitted for publication.

Competing interests The authors declare no competing interests.

References

1. Ding, F., Zhang, X., Sybrandt, J., Safro, I.: Unsupervised hierarchical graph representation learning by mutual information maximization. arXiv:2003.08420 (2020)
2. Di, X., Yu, P., Bu, R., Sun, M.: Mutual information maximization in graph neural networks. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–7 (2020). IEEE
3. Sui, Y., Wang, X., Wu, J., Lin, M., He, X., Chua, T.-S.: Causal attention for interpretable and generalizable graph classification. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1696–1705 (2022)
4. Zhang, J., Shi, X., Zhao, S., King, I.: Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems. arXiv:1905.13129 (2019)
5. Li, J., Zhang, T., Tian, H., Jin, S., Fardad, M., Zafarani, R.: Graph sparsification with graph convolutional networks. Int. J. Data Sci. Anal., 1–14 (2022)
6. Gharsallaoui, M.A., Tornaci, F., Rekik, I.: Investigating and quantifying the reproducibility of graph neural networks in predictive medicine. In: Predictive Intelligence in Medicine: 4th International Workshop, PRIME 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, October 1, 2021, Proceedings 4, pp. 104–116 (2021). Springer
7. Luo, L., Fang, Y., Cao, X., Zhang, X., Zhang, W.: Detecting communities from heterogeneous graphs: A context path-based graph neural network model. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 1170–1180 (2021)
8. Li, P., Yu, H., Luo, X., Wu, J.: Lgm-gnn: A local and global aware memory-based graph neural network for fraud detection. IEEE Trans. Big. Data (2023)
9. Xie, M., Irfan, M., Razzaq, A., Dagar, V.: Forest and mineral volatility and economic performance: evidence from frequency domain causality approach for global data. Resourc. Pol. **76**, 102685 (2022)
10. Ridley, M., Rao, G., Schilbach, F., Patel, V.: Poverty, depression, and anxiety: causal evidence and mechanisms. Science **370**(6522), 0214 (2020)
11. Brouwers, M.C., Simons, N., Stehouwer, C.D., Isaacs, A.: Non-alcoholic fatty liver disease and cardiovascular disease: assessing the evidence for causality. Diabetologia **63**, 253–260 (2020)
12. Prosperi, M., Guo, Y., Sperrin, M., Koopman, J.S., Min, J.S., He, X., Rich, S., Wang, M., Buchan, I.E., Bian, J.: Causal inference and counterfactual prediction in machine learning for actionable healthcare. Nat. Mach. Intell. **2**(7), 369–375 (2020)
13. Adebayo, T.S.: Environmental consequences of fossil fuel in spain amidst renewable energy consumption: a new insights from the wavelet-based granger causality approach. Int. J. Sustain. Dev. World Ecol. **29**(7), 579–592 (2022)
14. Wang, Y., Chu, Z., Ouyang, X., Wang, S., Hao, H., Shen, Y., Gu, J., Xue, S., Zhang, J., Cui, Q., *et al.*: Llmrg: Improving recommendations through large language model reasoning graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 19189–19196 (2024)
15. Yang, Z., Khatibi, E., Nagesh, N., Abbasian, M., Azimi, I., Jain, R., Rahmani, A.M.: Chatdiet: empowering personalized nutrition-oriented food recommender chatbots through an llm-augmented framework. Smart Health **32**, 100465 (2024)
16. Dwivedi, V.P., Joshi, C.K., Luu, A.T., Laurent, T., Bengio, Y., Bresson, X.: Benchmarking graph neural networks. arXiv:2003.00982 (2020)

17. Li, T., Zhou, Z., Li, S., Sun, C., Yan, R., Chen, X.: The emerging graph neural networks for intelligent fault diagnostics and prognostics: a guideline and a benchmark study. Mech. Syst. Signal Process. **168**, 108653 (2022)
18. Kosan, M., Verma, S., Armgaan, B., Pahwa, K., Singh, A., Medya, S., Ranu, S.: Gnnx-bench: Unravelling the utility of perturbation-based gnn explainers through in-depth benchmarking. arXiv:2310.01794 (2023)
19. Chen, T., Zhou, K., Duan, K., Zheng, W., Wang, P., Hu, X., Wang, Z.: Bag of tricks for training deeper graph neural networks: a comprehensive benchmark study. IEEE Trans. Pattern Anal. Mach. Intell. **45**(3), 2769–2781 (2022)
20. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv.org (2017)
21. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Adv. Neural Inf. Process. Syst. **30**, (2017)
22. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. arXiv.org (2018)
23. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? arXiv:1810.00826 (2018)
24. You, J., Ying, R., Leskovec, J.: Position-aware graph neural networks. In: International Conference on Machine Learning, pp. 7134–7143 (2019). PMLR
25. Zhao, T., Liu, Y., Neves, L., Woodford, O., Jiang, M., Shah, N.: Data augmentation for graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 11015–11023 (2021)
26. Wang, S., Su, X., Zhao, B., Hu, P., Bai, T., Hu, L.: An improved graph isomorphism network for accurate prediction of drug-drug interactions. Mathematics **11**(18), 3990 (2023)
27. Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., Li, H.: T-gcn: a temporal graph convolutional network for traffic prediction. IEEE Trans. Intell. Trans. Syst. **21**(9), 3848–3858 (2019)
28. Song, J., Son, J., Seo, D.-h., Han, K., Kim, N., Kim, S.-W.: St-gat: A spatio-temporal graph attention network for accurate traffic speed prediction. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, pp. 4500–4504 (2022)
29. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. **32**(1), 4–24 (2020)
30. Wu, J., He, J., Xu, J.: Net: Degree-specific graph neural networks for node and graph classification. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 406–415 (2019)
31. Maurya, S.K., Liu, X., Murata, T.: Simplifying approach to node classification in graph neural networks. J. Comput. Sci. **62**, 101695 (2022)
32. Wang, K., An, J., Zhou, M., Shi, Z., Shi, X., Kang, Q.: Minority-weighted graph neural network for imbalanced node classification in social networks of internet of people. IEEE Int. Things J. **10**(1), 330–340 (2022)
33. Sun, Z., Zhang, W., Mou, L., Zhu, Q., Xiong, Y., Zhang, L.: Generalized equivariance and preferential labeling for gnn node classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 8395–8403 (2022)
34. Wang, Z., Lin, Z., Li, S., Wang, Y., Zhong, W., Wang, X., Xin, J.: Dynamic multi-task graph isomorphism network for classification of alzheimer's disease. Appl. Sci. **13**(14), 8433 (2023)
35. Li, J., Cai, D., He, X.: Learning graph-level representation for drug discovery. arXiv:1709.03741 (2017)
36. Jha, K., Saha, S., Singh, H.: Prediction of protein-protein interaction using graph neural networks. Sci. Rep. **12**(1), 8360 (2022)
37. Shrestha, P., Maharjan, S., Arendt, D., Volkova, S.: Learning from dynamic user interaction graphs to forecast diverse social behavior. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2033–2042 (2019)
38. Ahale, P., Pattanshetti, T., Nayak, S.: Effectiveness of graph neural networks for user-user-item recommendation systems. In: 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 1527–1532 (2021). IEEE
39. Pearl, J.: Causality, 2nd edition Cambridge University Press, New York (2009)
40. Jiang, W., Liu, H., Xiong, H.: When graph neural network meets causality: opportunities, methodologies and an outlook. arXiv:2312.12477 (2023)
41. Yu, K., Guo, X., Liu, L., Li, J., Wang, H., Ling, Z., Wu, X.: Causality-based feature selection: methods and evaluations. ACM Comput. Surv. (CSUR) **53**(5), 1–36 (2020)
42. Schölkopf, B., Locatello, F., Bauer, S., Ke, N.R., Kalchbrenner, N., Goyal, A., Bengio, Y.: Toward causal representation learning. Proc. IEEE **109**(5), 612–634 (2021)

43. Wang, H., Liu, R., Ding, S.X., Hu, Q., Li, Z., Zhou, H.: Causal-trivial attention graph neural network for fault diagnosis of complex industrial processes. IEEE Trans. Ind. Inf., (2023)
44. Wale, N., Watson, I.A., Karypis, G.: Comparison of descriptor spaces for chemical compound retrieval and classification. Knowl. Inf. Syst. **14**, 347–375 (2008)
45. Borgwardt, K.M., Ong, C.S., Schönauer, S., Vishwanathan, S., Smola, A.J., Kriegel, H.-P.: Protein function prediction via graph kernels. Bioinformatics **21**(suppl_1), 47–56 (2005)
46. Debnath, A.K., Compadre, R.L., Debnath, G., Shusterman, A.J., Hansch, C.: Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity. J. Med. Chem. **34**(2), 786–797 (1991)
47. Zhou, Y., Huo, H., Hou, Z., Bu, F.: A deep graph convolutional neural network architecture for graph classification. Plos One **18**(3), 0279604 (2023)
48. Xie, Y., Lv, S., Qian, Y., Wen, C., Liang, J.: Active and semi-supervised graph neural networks for graph classification. IEEE Trans. Big Data **8**(4), 920–932 (2022)
49. McCallum, A.K., Nigam, K., Rennie, J., Seymore, K.: Automating the construction of internet portals with machine learning. Inf. Retr. (Boston) **3**(1), 127 (2000)
50. Giles, C.L., Bollacker, K.D., Lawrence, S.: Citeseer: An automatic citation indexing system. In: Proceedings of the Third ACM Conference on Digital Libraries. DL '98, pp. 89–98. Association for Computing Machinery, New York, NY, USA (1998). https://doi.org/10.1145/276675.276685
51. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv:1710.10903 (2017)
52. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv:1609.02907 (2016)
53. Yanardag, P., Vishwanathan, S.: Deep graph kernels. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1365–1374 (2015)
54. Zhou, P., Wu, Z., Wen, G., Tang, K., Ma, J.: Multi-scale graph classification with shared graph neural network. World Wide Web **26**(3), 949–966 (2023)
55. Moon, H.-J., Cho, S.-B.: A subgraph embedded gin with attention for graph classification. In: International Conference on Intelligent Data Engineering and Automated Learning, pp. 356–367 (2023). Springer