Knowledge-Based Systems 311 (2025) 113094

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

HebCGNN: Hebbian-enabled causal classification integrating dynamic impact valuing

Simi Job [a],*, Xiaohui Tao [a], Taotao Cai [a], Lin Li [b], Haoran Xie [c], Cai Xu [d], Jianming Yong [e]

[a] School of Mathematics, Physics and Computing, University of Southern Queensland, Australia
[b] School of Computer Science and Artificial Intelligence, Wuhan University of Technology, China
[c] School of Data Science, Lingnan University, Hong Kong Special Administrative Region
[d] School of Computer Science and Technology, Xidian University, China
[e] School of Business, University of Southern Queensland, Australia

ARTICLE INFO

Keywords:
Graph Neural Networks
Graph Convolutional Networks
Graph Attention Networks
Causality
Graph classification
Hebbian learning

ABSTRACT

Classifying graph-structured data presents significant challenges due to the diverse features of nodes and edges and their complex relationships. While Graph Neural Networks (GNNs) are widely used for graph prediction tasks, their performance is often hindered by these intricate dependencies. Leveraging causality holds potential in overcoming these challenges by identifying causal links among features, thus enhancing GNN classification performance. However, depending solely on adjacency matrices or attention mechanisms, as commonly studied in causal prediction research, is insufficient for capturing the complex interactions among features. To address these challenges, we present *HebCGNN*, a Hebbian-enabled Causal GNN classification model that incorporates *dynamic impact valuing*. Our method creates a robust framework that prioritizes causal elements in prediction tasks. Extensive experiments on seven publicly available datasets across diverse domains demonstrate that *HebCGNN* outperforms state-of-the-art models.

1. Introduction

Traditional graph classification utilizes GNNs such as Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) to classify nodes or graphs based on structural and topological features. Recent research highlights the importance of causality in machine learning [1,2], demonstrating its benefits in fields like medicine [3] and language modeling [4]. Despite this, causality remains underexplored in graph classification. Incorporating causality in graph classification aims to uncover and leverage causal relationships between nodes and edges, enhancing predictive modeling and pattern recognition by providing deeper insights into data dynamics. As graph datasets become more prevalent, incorporating causality has the potential to enhance decision-making, feature representation, model robustness and prediction interpretability. Despite advancements in GNN classification [5,6], the incorporation of causality into these methods remains largely unaddressed.

In graph-based approaches, directional edges indicate influence direction and adjacency matrices capture structural connections. For example, Cummings and Nassar [7] used directional edges for implicit causality in paper classification, and Kong et al. [2], Zhang et al. [8] used adjacency matrices for causal classification. Nonetheless, directional edges reveal only influence direction without explaining causal mechanisms, and adjacency matrices fail to identify true causal relationships. Consequently, these methods alone are insufficient for robust causal graph classification models. Attention mechanisms, while fundamental for causal learning as shown by Sui et al. [1], do not fully capture causality. They focus on interaction relevance rather than underlying causal mechanisms and their directional influences. Thus, contemporary GNNs face several challenges when relying solely on attention-based methods or adjacency matrices for causal classification.

Fig. 1 illustrates user engagement classification into *low*, *medium* and *high* levels. It compares two approaches: the traditional method (left) uses interaction metrics like *posts liked*, *comments made*, *shares* and *duration*, while the causal model (right) incorporates content quality and interaction frequency for deeper insights. Traditional methods using directional edges and adjacency matrices capture structural connections and influence direction, but miss causality. For example, directional edges connect *posts liked* to *comments made* without detailing their impact on user engagement. Adjacency matrices reveal connections but lack causal context. In contrast, a causal model integrates

* Corresponding author.
E-mail addresses: Simi.Job@unisq.edu.au (S. Job), Xiaohui.Tao@unisq.edu.au (X. Tao), Taotao.Cai@unisq.edu.au (T. Cai), cathylilin@whut.edu.cn (L. Li), hrxie@ln.edu.hk (H. Xie), cxu@xidian.edu.cn (C. Xu), jianming.yong@unisq.edu.au (J. Yong).

https://doi.org/10.1016/j.knosys.2025.113094
Received 24 November 2024; Received in revised form 17 January 2025; Accepted 26 January 2025
Available online 3 February 2025

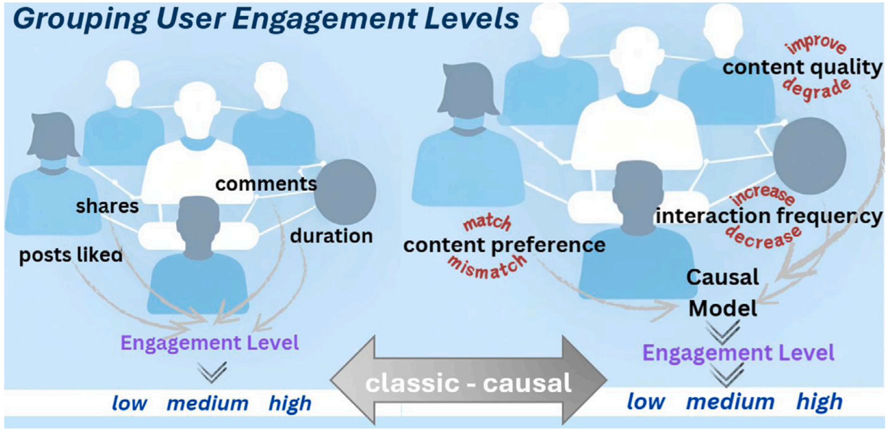


Fig. 1. Comparing two user engagement classification approaches: *traditional* (L) using interaction metrics, and *causal* (R) using content quality and interaction frequency.

causal reasoning to examine how changes in *shares* directly affect *duration* and how *content quality* influences *interaction frequency* and *content preferences*, leading to more precise predictions.

In this paper, we aim to address key challenges in this research domain. While adjacency matrices are effective for representing structural relationships, they lack the contextual depth necessary to capture causal dependencies. Additionally, attention mechanisms alone are insufficient for modeling complex causal relationships, as their sensitivity to irrelevant information can impair accuracy. To overcome these challenges, we present a framework called *HebCGNN*, Hebbian-enabled Causal Graph Neural Networks, building upon the work of Sui et al. [1]. We propose enhancing causal connection extraction using Hebbian learning to capture co-activation patterns, rather than relying solely on attention-based methods [1]. Hebbian learning is adopted in this work due to its biologically inspired mechanism, which is well-suited for capturing co-activation patterns and correlations in complex systems. It strengthens connections between nodes that are frequently co-activated, aligning with the idea that such correlations may suggest causal relationships, though establishing causality typically involves additional considerations beyond simple co-occurrence. This approach also address the issue of attention mechanisms being affected by irrelevant information. In the context of user engagement, Hebbian learning enables the model to better reflect the underlying patterns driving user behavior, distinguishing true causality from mere associations. It also improves robustness by emphasizing persistent co-activation rather than transient correlations. Thus, Hebbian learning enables more accurate causal identification, overcoming the limitations of attention mechanisms and ensuring the framework captures the true nature of the data.

Additionally, we introduce a dynamic causal attention mechanism with adaptive weighting termed *impact valuing*. A tailored loss function assigns lower weight to non-causal elements, higher weight to causal elements, and moderate weight to composite representations. This approach enhances causal significance, minimizes non-causal influence, and preserves overall feature relevance, ensuring a comprehensive capture of causality within its context.

The first contribution involves identifying the limitations of current mainstream causal GNN methods that rely on attention methods, which struggle to effectively capture complex causal relationships and are sensitive to irrelevant information. The second contribution is our development of an innovative GNN framework for causal classification, which combines a dynamic causality-based attention method with nodal co-activation pattern identification to highlight causal elements. Finally, we provide an empirical comparison of *HebCGNN* against state-of-the-art models across seven publicly available datasets from various domains, demonstrating that our model outperforms existing baseline methods.

2. Related work

This section briefly reviews relevant works on GNNs, Causality in GNNs and Hebbian Learning.

2.1. Graph neural networks

GNNs have been widely used for classification across various domains, including spatial, temporal and spatio-temporal contexts [9], and in fields such as text [10], image [11], medicine [12] and social networks [13]. [13] introduced *SNGNN*, a similarity-based GNN framework for node classification using mean aggregation. [14] proposed a GNN framework with curriculum learning that combined adaptive graph oversampling and neighbor-based metric learning for node classification. A key challenge for current GNN approaches is capturing underlying dependencies in complex structures, which can potentially be improved by incorporating causality.

2.2. Causality in GNNs

Researchers have recently highlighted the importance of causal learning in prediction tasks. In this regard, Li et al. [15] introduced a framework that enhanced recommendation explainability by utilizing dynamic knowledge graphs and supervised reinforcement learning to model the evolving interactions between items and users, potentially revealing underlying relationships that could provide causal insights in recommendation systems. In the same field, Wang et al. [16] introduced a causality-based recommender system designed to address unobserved confounding variables. Yao and Ge [17] designed a GRU model with attention for quality prediction using Granger Causality (GC). With the growth of graph-structured data, there is increasing focus on causality, with GNNs uncovering causal relationships. Wein et al. [18] developed a GNN framework for brain interaction extraction using RNNs and GC. Monken et al. [19] applied GNNs to analyze causal scenarios in international trade, revealing rapid trade flow shifts. Sui et al. [1] proposed a GNN model with Causal Attention Learning (CAL) for causal classification, leveraging attention mechanisms to differentiate causal from trivial features, and Lin et al. [20] developed a causal graph convolutional network for traffic prediction using RNNs and attention mechanisms. In the same domain, Xiong et al. [21] proposed gated fusion adaptive graph neural network (GFAGNN), which first captures long-term dependencies from raw data using stacked expansion causal convolution, then learns spatial dynamics through an adaptive graph attention network and an adaptive graph convolutional network, and finally extracts temporal features by passing the fused information through a lightweight channel attention mechanism. In contrast, drawing on counterfactual reasoning from causal learning theory, Li et al. [22] proposed an approach that extends recent developments in graph-based causal learning, where the explainable weights of paths are learned to uncover deeper causal relationships within recommendation systems, providing an alternative to traditional attention mechanisms.

Liu et al. [23] introduced Causal Intervention Graph Neural Network (CIGNN) framework, a GNN-based approach for fault diagnosis that employs an attention mechanism to automatically transform sensor signals into graph-based data. The framework incorporates causal intervention to address the confounding effect, thereby improving prediction accuracy. Liu et al. [24] proposed a causality-driven, GNN-based framework for continuous cuffless blood pressure (BP) estimation, aimed at constructing a causal graph that connects BP with wearable features to identify those features that are causally linked to BP variations. The study found that causal features provide better tracking of BP changes compared to Pulse Transit Time (PTT), an indirect measure of BP, and can improve the accuracy of BP measurements. Causal Graphs were also employed by Wang et al. [25] to model dependencies in graph-based collaborative filtering. In this context, causality was used to parameterize a Neural Causal Model, leading to the development of a framework called Neural Causal Graph Collaborative Filtering (NCGCF). NCGCF applied variational inference to approximate neural networks, facilitating the integration of meaningful causal effects from the causal graph to improve graph representation learning. A key challenge for causal GNNs, even those with attention mechanisms, is distinguishing causal from non-causal features. This can be addressed by more effectively prioritizing causal features over less informative ones.

2.3. Hebbian learning

Hebbian Learning, inspired by biological principles, strengthens connections between neurons that activate together. This method has been used to train Convolutional Neural Networks (CNNs) unsupervisedly [26] and in Hopfield networks for pattern inference and classification [27]. Hebbian CNNs have been applied to object recognition [28] and online learning classification [29]. A study by Eckmann et al. [30] demonstrated how synapse-type-specific competition, combined with Hebbian learning, enables the stable development of structured connectivity patterns in cortical circuits. By modeling plasticity at both excitatory and inhibitory synapses, the framework shows how diverse cortical response properties such as response normalization and center–surround suppression, can emerge from a single learning rule, highlighting the role of competitive learning in shaping synaptic connections that organize neural networks and guide the development of specific connectivity patterns. Additionally, Hebbian techniques have also been used to develop an attention-like mechanism using the match-and-control principle to create a transformer-like model [31]. Since this approach strengthens connections between co-activated neurons, it can be useful for deriving causality and may enhance causal prediction capabilities when incorporated into GNNs.

3. Preliminaries

The core components of our framework include Graph Neural Networks, the attention mechanism and Hebbian learning.

3.1. Graph Neural Networks

Graph Neural Networks (GNNs) process graph-structured data using message passing, updating node representations by aggregating neighborhood information. Each node v_i has a feature vector h_i, updated through Eq. (1) [32], where ${h_i}^l$ denotes the representation of node v_i for layer l and $N(v_i)$ represents neighboring nodes of v_i.

$$h_i^{(l+1)} = \text{Aggregate}\left(\left\{ h_j^l \mid v_j \in N(v_i)\right\}\right) \tag{1}$$

Graph Convolutional Networks (GCN) [33] use convolutional layers to aggregate features from neighboring nodes. A typical GCN includes convolutional layers, a linear layer and a non-linear activation function, which processes each node, aggregates features, updates representations, and applies activation.

3.2. Attention mechanism

Attention mechanisms enhance neural networks by assigning importance to input elements, thereby improving information capture. In a graph, each node v_i has a feature representation h_i. Eq. (2) [34] shows computation of node embedding ${h_i}^{(l+1)}$ for layer $l+1$, using multi-head attention with K heads, where W^k is the weight matrix. The attention coefficient between nodes v_i and v_j is given by $\frac{exp(e_{ij})}{\sum_{k\in N(i)} exp(e_{ik})}$, with $N(i)$ denoting the neighbors of v_i, and e_{ij} denoting the similarity measure between v_i and v_j.

$$h_i^{(l+1)} = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\left(\sum_{j\in N(i)} \frac{\exp(e_{ij}^k)}{\sum_{k'\in N(i)}\exp(e_{ik'}^k)} W^k h_j^{(l)}\right)\right) \tag{2}$$

3.3. Hebbian Learning

Hebbian Learning, based on Hebb's rule [35] operates on the principle that simultaneous activation of neurons lead to strengthening of their connections. This concept can be directly beneficial for learning causality, as the consistent occurrence of two events together can signify a causal relationship between them. The Hebbian learning rule (in an unsupervised setting) is represented as Eq. (3), where Δw_{ij} is the weight change from unit i to unit j, ϵ is the learning rate and a_i, a_j are the respective unit activation levels [36]. Incorporating Hebbian updates into neural networks allows them to learn event coexistence and acquire causal knowledge, making them more responsive to causal relationships through iterative learning.

$$\Delta w_{ij} = \epsilon\, a_i\, a_j \tag{3}$$

4. Methodology

In graph modeling, causality signifies one node's influence on another, with directed edges indicating causal dependencies based on Structural Causal Models (SCM) [37]. This section defines the research problem and introduces *HebCGNN* to address the challenges outlined in Section 1.

4.1. Problem definition

In Causal Classification, given a graph $G_i = (V_i, E_i)$, with nodes V_i and edges E_i, the challenge is to design a causality-based GNN model that effectively manages causal relationships in the graph. The goal is to develop a model f for both graph and node classification, predicting class labels Y while addressing spurious correlations from shortcut features. Specifically, for graph classification, the aim is to develop a causality-based GNN model $f : G_i \rightarrow Y$ to predict the class label of the entire graph G_i, leveraging causal features and avoiding confounding effects. For node classification, the aim is to design a GNN model $g : V_i \rightarrow Y$ to predict each node's class label $v \in V_i$, using causal relationships to improve accuracy and reduce biases. The objective is to ensure that the GNN model effectively utilizes causal features, removes backdoor paths that could cause spurious correlations, and thus enhances prediction reliability.

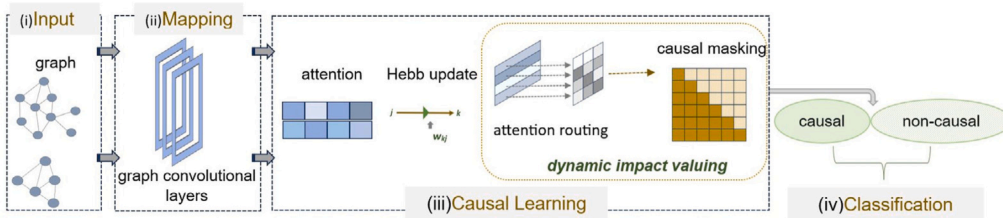


Fig. 2. *HebCGNN* Architecture. (i) Input is pre-processed (ii) Features are input to GCN; convolutional layers aggregate node information, update representations, and apply ReLU. (iii) Output undergoes multi-head attention to generate representations with attention weights, followed by linear transformation and Hebbian Learning to obtain causal and non-causal representations. Dynamic impact valuing with attention routing and causal masking is then applied. Routing determines significance through linear layers, creating a causal mask that adjusts both causal and non-causal attention weights based on routing weights. Impact Coefficients are computed to optimize the weighted representations of causal and shortcut features. (iv) Classifier is applied.

4.2. Overview of HebCGNN

HebCGNN comprises of three major components: Graph Mapping, Causal Learning and Causal Classification, which we discuss in detail in the following sections. Algorithm 1 details the procedure for *HebCGNN* and Fig. 2 illustrates the model's overall architecture. The algorithm starts by initializing node features and the adjacency matrix (line 4), then utilizes GCN and attention to enhance the graph dataset representation (lines 4–8). It then iteratively employs Hebbian learning to derive causal (c) and non-causal (nc) representations (lines 9–12), followed by attention routing and causal masking (M) to compute impact coefficients (δ) for c and nc (lines 13–14). This dynamic process facilitates the generation of causal ($cRepr$) and non-causal ($ncRepr$) representations through impact valuation (line 15).

Algorithm 1 *HebCGNN*: GNNs with Hebbian Learning and dynamic impact valuing for causal learning

1: **Input:** Graph Dataset G
2: **Output:** causal, non-causal representation
3: **for** each graph G **do**
4: Initialize node features X, adjacency matrix A from G
5: Initialize GCN layer
6: $H_g \leftarrow GCN(X, A)$
7: $H_A \leftarrow Attention(H_g, heads)$
8: Apply linear transformation $\rightarrow H_{transformed}$
9: **while** i $\leq$ iterations **do**
10: causal c, non-causal nc $\leftarrow$HebbLearn($H_{transformed}$)
11: $\Delta w_{ij} = \epsilon a_i a_j$
12: **end while**
13: $\omega_{r_c}, \omega_{r_{nc}} \leftarrow$ routing(c), routing(nc)
14: $\delta_c, \delta_{nc} \leftarrow M(\omega_{r_c}, \omega_{r_{nc}})$ ▷ dynamic causal factor, M: mask
15: $cRepr, ncRepr \leftarrow c \odot \delta_c, nc \odot \delta_{nc}$
16: **end for**

Our framework incorporates Hebbian learning to capture feedback loops and recurrent patterns, improving the understanding of contextual causal relationships. Hebbian learning, based on biological principles, strengthens the interactions between nodes that co-occur or influence each other over time. In our model, this process is crucial for refining the feature representations of graph nodes by focusing on causal relationships rather than just statistical correlations. Hebbian learning achieves this by adjusting the weights of connections between nodes that exhibit simultaneous or correlated activation patterns. By leveraging the feedback loops inherent in graph structures, where nodes evolve based on the features of neighboring nodes, Hebbian learning enhances those features that emerge as key causal factors. This makes it easier for the model to differentiate between causal and non-causal features. Non-causal features, termed *shortcut features* in line with causality literature [38], are considered to capture confounding effects, which represent correlations that do not reflect true causality. Through Hebbian learning, the model effectively captures and differentiates between causal and non-causal features. It strengthens causal relationships while reducing the influence of confounders, ensuring that the most relevant connections are emphasized in the learned representations. This leads to a more robust understanding of the graph's underlying causal structure.

The process of *Impact Valuing* is introduced to generate *impact coefficients* for both causal and non-causal representations, leveraging dynamic causal processes and adaptive attention routing. Instead of relying on static weights, the framework uses attention mechanisms to route information dynamically and prioritize relevant data. Additionally, a dynamic causal attention process with causal masks is employed to improve contextual understanding. The *impact coefficients* are applied to both causal and non-causal representations to make informed predictions. By combining Hebbian learning with dynamic causal processes, our model minimizes the sensitivity of attention methods to irrelevant data, adjusting weights based on relevance and focusing on essential information while filtering out irrelevant details.

4.2.1. Complexity analysis of HebCGNN

This section examines the complexity of the framework by analyzing its key components and evaluating their impact on both time and space requirements. The analysis also covers scalability, limitations and potential optimization opportunities.

Time and Space Complexity:

Graph Neural Network Operations: GNNs propagate information across nodes using message passing, with each layer having a time complexity of $O(E)$, where E is the number of edges in the graph. The space complexity is $O(N + E)$, where N is the number of nodes. The time complexity for updating node features in each GNN layer is $O(E \cdot d)$, where d is the feature dimensionality. Storing the node features typically requires $O(N \cdot d)$ space.

Hebbian Learning: To compute the Hebbian updates, the interactions between feature or node pairs must be evaluated, resulting in a time complexity of $O(N^2 \cdot d)$. The process requires $O(N^2 \cdot d)$ space to store interaction-based feature values for each node pair.

Causal Impact Valuing: Evaluating causal dependencies across all node pairs to calculate the dynamic impact of features results in a time complexity of $O(N^2 \cdot d)$. Similarly, storing causal dependencies between nodes and their features requires $O(N^2 \cdot d)$ space.

Scalability and Optimization Opportunities: As graph size grows, computational costs increase, particularly with the $O(N^2 \cdot d)$ terms. This quadratic dependence on node count can challenge scalability, but optimizations can help mitigate these issues. Sparse matrices for adjacency and node features reduce time and space complexities, while batch processing minimizes $O(N^2)$ operations by processing smaller graph subsets. Approximate causal impact methods like sampling, further reduce the computational load.

Managing Trade-offs in Precision and Performance: To address the time and space complexity of the Hebbian-enabled Causal GNN model, various trade-offs can be considered. Reducing the precision of Hebbian updates or causal impact values, such as using approximate values or quantizing interactions, can substantially lower both time and space complexity but may compromise model accuracy. Additionally, limiting the scope of Hebbian learning or causal analysis to local neighborhoods rather than the entire graph provides a balance between computational efficiency and model performance.

4.3. Graph mapping

In *HebCGNN*, graph datasets are pre-processed and input into GCN for Graph Mapping to learn node representations. The GCN's convolutional layers aggregate neighboring node information to capture the graph structure and update features, which are then processed by the ReLU activation function. This is denoted in relation to the input feature matrix X, the adjacency matrix A, and the weight matrix Θ, as shown in Eq. (4), where $GCN(X, A, \Theta) = A \cdot X \cdot \Theta$ and Z denotes the output feature matrix.

$$Z = ReLU(GCN(X, A, \Theta)) \tag{4}$$

4.4. Causal learning

In causal learning, identifying causal features is essential for accurate predictions and minimizing the impact of confounding factors. Contextual features, often more closely related to the graph's causal structure, reveal how nodes influence each other across different contexts. By distinguishing these from object features, which may reflect mere correlations, the model prioritizes causally significant features. To achieve this, the feature representation obtained from the Mapping unit is passed to the Causal Learning unit, where it undergoes three processes: attention learning, Hebbian learning and Dynamic Impact Valuing.

4.4.1. Attention mechanism

The attention mechanism acts as an effective method for identifying key features and relationships in a graph by assigning weights to nodes and their neighbors according to their contextual importance. The input GCN feature matrix X is processed through a multi-head attention unit with four heads ($K = 4$), each learning distinct patterns. The attention mechanism computes a set of attention scores A_k for each head, using a dot-product similarity to capture pairwise relationships. The attention mechanism operates as shown in Eq. (5), where q_i is the query vector for node i, k_j is the key vector for node j and $N(i)$ represents the neighbors of node i.

$$\text{Attention score } A_k(i,j) = \frac{\exp(q_i \cdot k_j)}{\sum_{j' \in N(i)} \exp(q_i \cdot k_{j'})} \tag{5}$$

The outputs from each attention head are aggregated by concatenating and applying a linear transformation to obtain the final feature matrix as shown in Eq. (6), where H_1, H_2, H_3, H_4 are the outputs from the four attention heads, and W_{attn} is the weight matrix and b_{attn} is the bias.

$$H_{\text{attn}} = \text{concat}\left(H_1, H_2, H_3, H_4\right) \cdot W_{\text{attn}} + b_{\text{attn}} \tag{6}$$

The attention weights are then normalized using a softmax function to focus on the most relevant features as denoted in Eq. (7). The weights $\alpha_{i,j}$ emphasize the contextual and object-related aspects of each feature.

$$\alpha_{i,j} = \text{softmax}\left(A_k(i,j)\right) \tag{7}$$

4.4.2. Hebbian learning

The output of the attention mechanism, H_{attn}, representing the significance of node interactions, is subsequently passed to the Hebbian learning unit to generate both causal and non-causal representations. While the attention unit highlights relevant features, the Hebbian unit strengthens connections among causally related ones. This ensures that the model not only highlights important contextual relationships through attention but also reinforces these relationships dynamically as causal links. In the context of GNNs, Hebbian learning is implemented by iteratively updating connection weights between nodes based on their feature interactions. The feature matrix x represents the nodes and their attributes, and the Hebbian learning rule adjusts weights H_w to reinforce causal relationships by magnifying connections between nodes that influence each other. This iterative process ensures that causal dependencies are strengthened over time, while non-causal influences are minimized. This process is repeated for input x as $Hebbian^N(x)$ (with N = 2), refining representations over time. Hebbian weights, H_W, are updated with a learning rate $\eta = 0.01$ based on interactions within the feature matrix x (Eq. (8)). These weights generate object and contextual representations, capturing node attributes and their contextual relationships. Through iterative updates, this process results in two types of representations: causal representations, where stronger connections reflect causal influences between nodes, and non-causal representations, where weakened connections indicate no true causal relationship. In this way, the attention mechanism guides the model to focus on relevant features, while Hebbian learning ensures that these features are causally reinforced and differentiated from spurious (non-causal) associations.

$$H_W = H_W + \eta \times \left(x^\top \cdot x\right) \tag{8}$$

4.4.3. Dynamic impact valuing

In our framework, the distinction between causal and non-causal (shortcut) features is critical for accurate causal classification. Causal features represent true cause–effect relationships between nodes, whereas shortcut features often capture spurious correlations or confounding effects. Therefore, we emphasize identifying shortcut features and assigning them optimal weights via *impact valuing*, rather than assigning full weight in classification. This approach enhances classification performance by balancing the contributions of both causal and shortcut features. Prior to reaching the readout layers for final classification, both causal and non-causal elements undergo *impact valuation*, achieved through attention routing and causal masking as outlined in Lines 13–14 of Algorithm 1.

Attention routing

Attention routing controls information propagation through the network and assigns significance scores to various elements in the input representation. It involves applying a linear layer to both contextual and object representations, followed by normalization, to determine the relative significance of causal and non-causal elements. The linear transformation L_t for input x_c is given by Eq. (9), where W is the weight matrix and b is the bias vector. This process allows the model to allocate attention to both past and present information, preserving the causal sequence.

$$L_t = W \cdot x_c + b \tag{9}$$

The causal interpretation of attention routing is ensured through dynamic causal masking and Hebbian learning. The causal mask restricts each feature to attend only to preceding elements, preserving causal relationships. Attention routing assigns importance to true cause–effect features, while Hebbian learning strengthens causal connections and weakens spurious ones. This combination of causal masking and Hebbian learning emphasize causal features and filter out non-causal influences, ensuring reliable results.

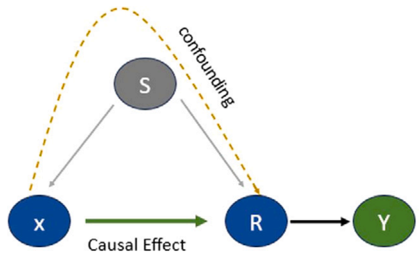


Fig. 3. Causal Classification Model with input graph X, its representations R, and shortcut features S, causing a confounding effect between X and predictions Y.

Causal masking

Using the attention weights from attention routing, a dynamic causal mask is created to reflect causal relationships between elements, ensuring that each element focuses only on preceding (causal) elements and disregards future (non-causal) influences. This causal mask ensures that each feature can only attend to preceding elements in the causal chain. The resulting *Impact Coefficient*, IC_c, is computed by applying the causal mask $Mask_c$ to the attention weights and is used to optimize the weighted representations of causal and shortcut features, as shown in Eq. (10), where α_c represents the attention score for the causal feature. This dynamic causal masking integrates causal relationships into the representations obtained through Hebbian learning. The loss function, $\mathcal{L}$ (Eq. (11)) prioritizes causal factors (c) with high weight (wt) and assigns lower importance to shortcut features (nc). A combination of both causal and shortcut features, *composite* (co), receives intermediate weight in the loss formulation. Details on the loss function values are provided in Section 5.1.

$$IC_c = \text{Mask}_c \cdot \alpha_c \tag{10}$$

$$\mathcal{L} = wt_c \times L_c + wt_{nc} \times L_{nc} + wt_{co} \times L_{co} \tag{11}$$

The causal mask is acyclic by design, ensuring that each element can attend only to preceding elements. This unidirectional flow ensures that no cycles are formed in the information flow, as each element's focus is directed backward along the causal chain, preventing circular dependencies and ensuring valid causal relationships. This acyclical property is essential for maintaining a clear causal structure and preventing feedback loops that could impair the model's ability to distinguish between genuine causal influences and spurious correlations.

Aligning attention routing with causal masking

In this model, attention routing assigns importance to context and object representations by applying linear transformations and normalization, which helps the model focus on significant features. The attention weights are computed through dedicated MLP layers for both context and object representations. The causal mask works in conjunction with attention routing by conditioning the attention weights, ensuring that each feature attends only to preceding (causal) elements. This dynamic mask prevents future (non-causal) dependencies and enforces a unidirectional flow of information. By restricting attention to past elements, the model prioritizes meaningful cause–effect features and filters out non-causal influences.

4.5. Causal classification and mechanisms

Causal classification involves categorizing graphs or nodes based on causal relationships, where edges show connections that affect class labels. The goal is to improve classification accuracy using these causal links. Fig. 3 is a simplified depiction of this process, where input data X generates causal representations R to predict labels Y, while shortcut features S might cause confounding. Attention mechanisms target relevant graph parts, and Hebbian Learning enhances causality extraction by detecting patterns and co-occurrences. Impact valuation refines this by prioritizing features based on their classification effect. Our model dynamically evaluates causality through routing to compute impact coefficients for causal and shortcut features, using a weighted loss function. This enhances key causal connections, improves predictions and reduces confounding effects. The causal mechanism of the framework is outlined below.

Causal Pathways: The model uses graph convolution and attention mechanisms to capture causal pathways between nodes, while Hebbian learning helps identify causally relevant connections over time, revealing the underlying causal mechanisms. The causal mask and impact coefficients further ensure that these pathways remain clear and direct, allowing the model to focus on the true cause–effect relationships. Through this process, the model builds and refines causal pathways, identifying how features interact causally.

Causal Sufficiency: The framework ensures causal sufficiency by incorporating all relevant features needed to identify causal relationships. Through Graph Mapping and Causal Learning, the model processes both contextual and object-related features. The attention mechanism highlights key causal features, while Hebbian learning refines them by strengthening causal interactions. Impact valuation and causal masking further prioritize true causal factors, ensuring that no critical features are omitted from the model.

Feedback Loops: The model leverages Hebbian learning to emphasize feedback loops and recurrent patterns in the graph structure, iteratively adjusting weights to build causal dependencies over time. A causal mask is applied to enforce a unidirectional flow of information, ensuring that each element can only attend to preceding (causal) elements, thus preventing feedback loops and maintaining a clear temporal or causal order. This acyclical constraint avoids circular causality, reinforcing the model's ability to correctly distinguish between cause and effect throughout the learning process.

Confounding Mitigation: The framework addresses confounding by differentiating between causal features and shortcut features. Hebbian learning adjusts the weights based on observed co-occurrences, while the attention mechanism prioritizes key relationships. By distinguishing true causal links from spurious correlations, the model minimizes confounding effects. The impact valuation process further ensures that shortcut features, which are often indicative of confounding, do not bias the classification.

Additionally, by dynamically adjusting feature importance through attention mechanisms and Hebbian learning, the framework ensures that the identified causal pathways are both relevant and data-driven. This process reduces the risk of confounders biasing the causal structure and enhances the identifiability of causal relationships, ensuring that the model accurately identifies and prioritizes true causal influences over spurious correlations.

5. Experiments

This section outlines the experiment design, results and analyses for evaluating the *HebCGNN* model.

5.1. Experiment design

The study addresses the following research questions:

RQ1. What impact does incorporating causality-aware graph models have on GNNs' classification performance?
RQ2. What effect does dynamic impact valuing of features have on improving causal classification?
RQ3. What are the contributions made by the various components in the framework towards enhancing GNN?

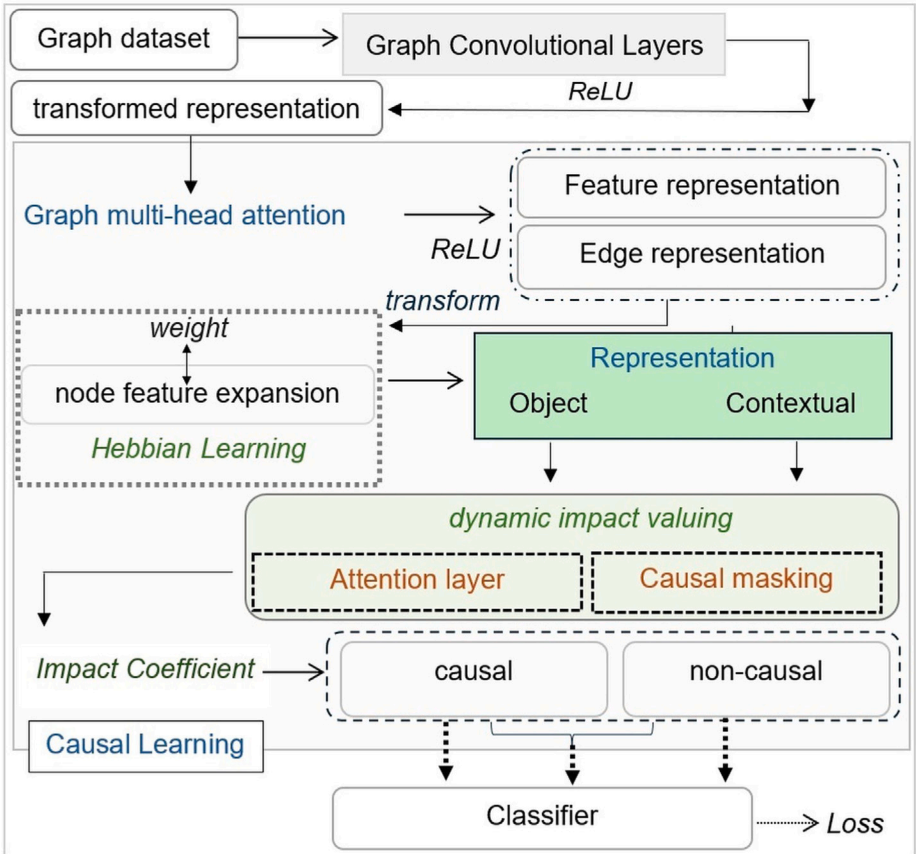


Fig. 4. Workflow for *HebCGNN*.

Table 1
Summary of datasets used in the study. The citation datasets are used for node classification tasks. The bio-chemical and social datasets are used for graph classification tasks, where the number of nodes and edges are for the 1st graph.

Dataset	# Graphs	Avg. nodes	# Nodes	Avg. edges	# Edges	# Classes
Cora	1	–	2708	–	10 556	7
Citeseer	1	–	3327	–	9104	6
NCI1	4110	29.87	21	32.30	42	2
Proteins	1113	39.06	42	72.82	162	2
Mutag	188	17.93	17	19.79	38	2
IMDB-B	1000	19.77	20	96.53	146	2
Reddit-B	2000	429.63	218	497.75	480	2

The experimental design for our model is shown in Fig. 4. The graph dataset was processed through graph convolutional layers and attention layers with four heads, to obtain feature representations. These representations were then refined through a recurrent Hebbian learning unit with a learning rate $\epsilon = 0.01$, producing causal, shortcut, and composite (a fusion of causal and shortcut) representations. An *impact coefficient* was applied to these representations through impact valuing (described in Section 4), before being forwarded to their respective readout layers for final predictions. A custom loss function weighted causal elements at over 50%, shortcut features at 15%, and the composite representation at approximately 25%.

5.1.1. Experimental settings

Experiments were conducted using Visual Studio Code 1.84 on Ubuntu 22.04 with Python 3.11 and PyTorch, leveraging an NVIDIA GeForce RTX 3070 Ti GPU. The model was trained with a learning rate of *1e–3*, batch size of 128 and a drop out rate of 0.2 over 20 epochs with early stopping.

5.1.2. Datasets

We used datasets from biochemistry, citation and social networks in our experiments to evaluate our model, with a summary provided in Table 1.

Bio-Chemical: *NCI1* [39] is a cheminformatics dataset where each graph represents a chemical compound. *Proteins* [40] comprises nodes representing amino acids. *Mutag* [41] is a nitroaromatic compound dataset, where nodes correspond to atoms and edges denote bonds. These datasets are established benchmarks in graph studies [1,42,43].

Citation: *Cora* [44] and *Citeseer* [45] are citation network datasets that categorize scientific publications into 7 and 6 classes respectively. Nodes represent papers, and edges represent citation relationships, highlighting their significance in graph classification research across numerous studies [5,6].

Social: IMDB-Binary (IMDB-B) and REDDIT-Binary (REDDIT-B) [46] are widely used in graph-related studies [47,48], with the former categorizing movie collaborations by genres and the latter categorizing online discussions into Q&A or discussion communities.

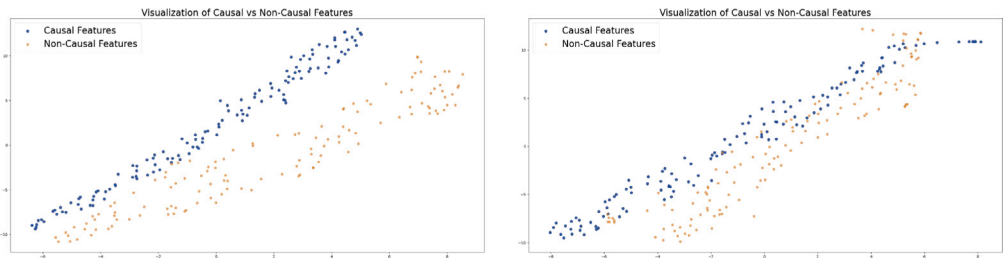
5.1.3. Baseline models

Traditional GNN models are: (1) ***GCN*** [5] uses convolution to propagate information through feature aggregation across a graph. (2) ***GAT*** (Graph Attention Network) [6] uses attention to weight neighboring node information, capturing dependencies. (3) ***GraphSAGE*** (Graph Sample and Aggregation) [49] learns node representations by sampling and aggregating neighbor features, iteratively updating embeddings.

Causal GNN models are: (1) ***GCN-CAL and GAT-CAL*** [1] use Causal Attention Learning (CAL) to explore causality in GNN classification, leveraging a GNN-based encoder for node representations and linear layers for attention, with GCN and GAT models respectively. (2) ***GCN-ICL and GAT-ICL*** [50] use Information-based Causal Learning (ICL) to

Table 2
F-scores for node classification (citation datasets) and graph classification (socio/bio datasets) tasks.

Model	NCI1	Proteins	Mutag	Cora	Citeseer	IMDB-B	REDDIT-B
GCN [5]	0.8737	0.7783	0.5138	0.7261	0.4488	0.7562	0.9826
GAT [6]	0.8446	0.7888	0.6355	0.7241	0.4574	0.7480	0.9793
GraphSAGE [49]	0.7621	0.7142	0.7830	0.7081	0.4207	0.7429	0.9679
GCN-CAL [1]	0.8517	0.8333	0.4310	0.6880	0.3967	0.8071	0.9832
GAT-CAL [1]	0.8248	0.8060	0.5172	0.6627	0.3906	0.7593	0.9790
GCN-ICL [50]	0.7907	0.7093	0.8582	0.7394	0.4271	0.7079	0.9121
GAT-ICL [50]	0.7748	0.7134	0.8542	0.7447	0.5176	0.7282	0.9074
HebCGNN	**0.9084**	**0.8533**	**0.8932**	**0.7563**	**0.5818**	**0.9447**	**0.9935**
Improvement of *HebCGNN* over best baseline	3.47%	2.00%	3.50%	1.16%	6.42%	13.76%	1.03%



(a) NCI1 dataset (b) IMDB-B dataset

Fig. 5. Depiction of causal (in blue) and non-causal (in orange) representations.

explore causality in GNN classification, combining information theory with causal analysis.

5.2. Experiment results

This section presents and analyses the experiment results.

5.2.1. Performance metrics

The model is evaluated with 5-fold cross-validation. F-score is used as the primary evaluation metric to balance Precision and Recall, providing a robust measure of model performance. It is well-suited for graph data due to its resistance to noise and its advantage over accuracy scores in handling class imbalance.

5.2.2. Performance comparison and analysis

In this section, we present the experimental results summarized in Table 2. The results cover both graph (bio-chemical, social datasets) and node (citation datasets) classification tasks. The evaluation metric used is F-score as discussed earlier. The *HebCGNN* model consistently outperforms baseline models in F-scores across all datasets for both tasks. Notably, improvements of approximately 3.5–13.8% are observed for the *NCI1*, *Mutag*, *Citeseer* and *IMDB-B* datasets. Specifically, *HebCGNN* shows F-score gains of 2–3.5% for bio-chemical datasets and 1–13.8% for social network datasets. The marginal 1% improvement for the *Reddit-B* dataset, which maintains F-scores above 90% across all models, may be due to its unique characteristics, such as highly discriminative features or consistent data patterns. In node classification also, *HebCGNN* outperforms baselines, with F-score improvements ranging from 1.2%–6.4%. The *Citeseer* dataset is particularly notable, with baseline models showing F-scores below 50% (except GAT-ICL model), highlighting the effectiveness of causal classification. These results show that integrating a well-designed causality module significantly enhances GNN classification performance. Overall, *HebCGNN* exhibits strong generalization, outperforming all baseline models with higher F-scores across all datasets.

To evaluate the model's ability to differentiate between causal and non-causal representations, we visualized a subset of these using t-SNE plots. Fig. 5 shows causal representations in blue and non-causal representations in orange for NCI1 (Fig. 5(a)) and IMDB-B (Fig. 5(b)) datasets. The plots reveal distinct clusters, indicating that our model effectively distinguishes between these two feature types and encodes them into separate spaces, confirming its effectiveness in differentiating causal and non-causal embeddings.

Statistical Analysis: We performed a paired t-test to compare our model's F-scores against a baseline GNN model with only the attention unit (*HebCGNN* without Hebbian unit or impact valuing) for all datasets. The null hypothesis assumes no difference in mean scores between the models, while the alternative hypothesis suggests a difference. The test yielded a statistic of 3.0595, and a p-value of 0.0222. With a significance level of $\alpha = 0.05$, we reject the null hypothesis, indicating a significant difference in F-scores between the two models.

Ablation Study This section evaluates how each component of the *HebCGNN* model affects classification performance, focusing on the roles of impact valuing causal and shortcut features, and the effects of the attention module, impact-valued Hebbian update and custom loss function.

Causal and Non-Causal factors. The *HebCGNN* model utilizes a custom loss function with an optimal ratio of non-causal and composite representations as detailed in *Design*. This section investigates how varying these ratios affects model performance. We keep the causal element's weighting static and adjust the shortcut and composite elements, analyzing their impact. Fig. 6 shows performance under different weightings for shortcut and composite elements, including seven scenarios besides our proposed model, labeled *(H)*. Impact coefficients for these scenarios are fixed at 0.6 and 0.4 for causal and shortcut features, respectively, differing from the dynamic impact valuation used in *HebCGNN*. The original weighting of the loss function in *HebCGNN* (outlined in *Design*), without dynamic impact valuing, is depicted as (.3,.5) in the figure, which was found to be optimal based on our experiments. These data points are indicated by the gray circles in the figure. The *HebCGNN* model, denoted as *(H)*, is marked with brown squares. As seen in the figure, setting aside the *HebCGNN* model, the loss weighting values of (.3,.5) are optimal for all datasets, except *Mutag*. Notably, datasets such as *Citeseer*, *Proteins* and *IMDB-B* exhibit a considerable improvement in F-scores with loss weighting values of (.3,.5). The weighting values of (.3,.5) along with dynamic causal impact valuing, as shown by *(H)* in the figure, enhance model performance.

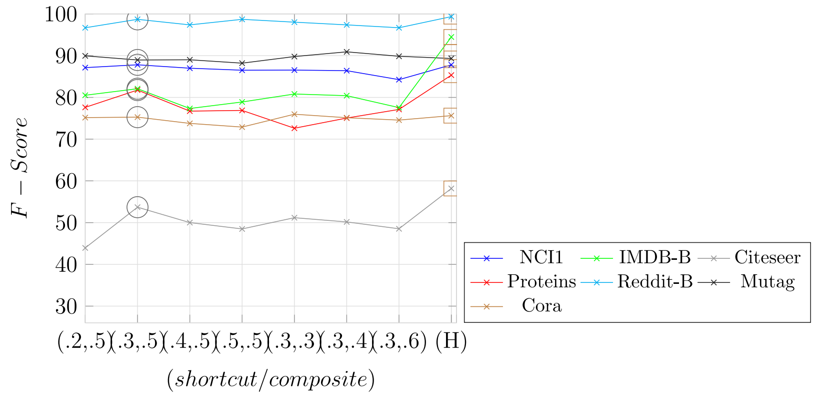


Fig. 6. F-scores: varied shortcut/composite impact values.

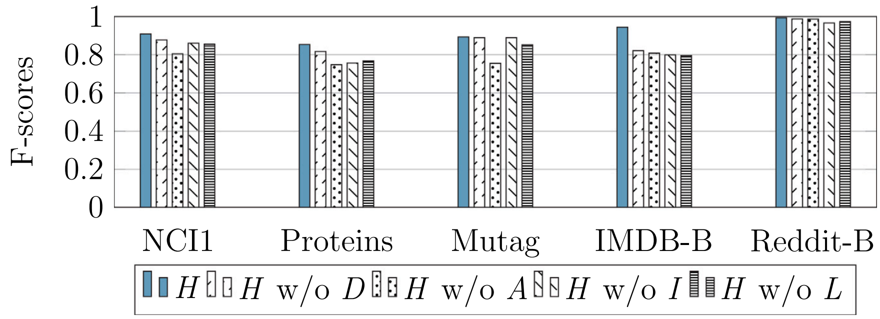


Fig. 7. F-scores for diverse components (*D*: Dynamic impact value, *A*: Attention, *I*: Impact value, *L*: Custom loss) in *HebCGNN* (*H*).

Component-level Study. We examine the contributions of three components to the *HebCGNN* model for graph classification tasks, with the results presented in Fig. 7. The cyan-colored bar represents our proposed model, *H*. We examine *HebCGNN* under various conditions: without dynamic impact valuing of the causal and shortcut features from the Hebbian unit (*H w/o D*), without the attention component (*H w/o A*), without impact valuing of the causal and shortcut features from the Hebbian unit (*H w/o I*), and without the custom loss function (*H w/o L*). The findings reveal that each component significantly enhances F-scores. While attention mechanism plays a key role in the classification of *NCI1* and *Mutag* datasets, a significant improvement is observed when all three components are incorporated in the architecture. Omitting dynamic impact valuing after Hebbian update leads to a performance drop of 0.37% to 12.3% compared to using fixed impact valuing. Removing the custom loss function results in a performance decline of 2% to 14.8% across all datasets.

Summary of Results. We address our research questions as follows:

- **RQ1.** Our model exhibits significant improvements in classification F-scores over all baseline models, highlighting the role of causality in GNN classification.
- **RQ2.** The results of the sensitivity analyses illustrate that dynamic impact valuing of causal and shortcut features play a crucial role in causal classification. This is particularly evident for *Citeseer*, *Proteins* and *IMDB-B* datasets, where dynamic impact valuing demonstrates a considerable improvement in performance over the static impact coefficient of (.3,.5). Furthermore, removing the impact valuing component during the training process leads to a decline in performance across all datasets.
- **RQ3.** Based on the ablation study, it can be concluded that each of the three components: attention, impact valuing and custom loss units, make a substantial contribution to the process of causal classification. This effect is particularly pronounced for *NCI1* and *Proteins* datasets.

6. Conclusion

In this paper, we introduced *HebCGNN*, a causal classification framework for graph neural networks that integrates Hebbian learning and dynamic impact valuing. By incorporating causality, *HebCGNN* enhances classification performance and extracts meaningful insights from graphs beyond what can be achieved solely through adjacency matrices. Hebbian learning enhances the attention mechanism, distinguishing between causal and non-causal components, which are then optimally processed through dynamic impact valuing to enhance performance. Experimental results demonstrate that *HebCGNN* outperforms existing state-of-the-art methods. Future work include optimizing its scalability to handle large-scale graph data efficiently. This will involve improving the computational efficiency of both the Graph Mapping and Causal Learning components, particularly in the context of the attention and Hebbian learning processes.

CRediT authorship contribution statement

Simi Job: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation. **Xiaohui Tao:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Taotao Cai:** Writing – review & editing, Validation, Supervision, Methodology, Investigation. **Lin Li:** Writing – review & editing, Methodology, Conceptualization. **Haoran Xie:** Writing – review & editing, Methodology, Conceptualization. **Cai Xu:** Writing – review & editing, Conceptualization. **Jianming Yong:** Writing – review & editing, Supervision, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Datasets are publicly available.

References

[1] Y. Sui, X. Wang, J. Wu, M. Lin, X. He, T.-S. Chua, Causal attention for interpretable and generalizable graph classification, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 1696–1705.

[2] W. Kong, M. Qiu, M. Li, X. Jin, L. Zhu, Causal graph convolutional neural network for emotion recognition, IEEE Trans. Cogn. Dev. Syst. (2022).

[3] H. Ahmad, M. Endres, K. Newman, P. Santiesteban, E. Shedden, W. Weimer, Causal relationships and programming outcomes: A transcranial magnetic stimulation experiment, in: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, 2024, pp. 1–13.

[4] A. Lampinen, S. Chan, I. Dasgupta, A. Nam, J. Wang, Passive learning of active causal strategies in agents and language models, Adv. Neural Inf. Process. Syst. 36 (2024).

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

[6] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, 2017, arXiv preprint arXiv:1710.10903.

[7] D. Cummings, M. Nassar, Structured citation trend prediction using graph neural networks, in: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2020, pp. 3897–3901.

[8] J. Zhang, X. Zhang, G. Chen, Q. Zhao, Granger-causality-based multi-frequency band EEG graph feature extraction and fusion for emotion recognition, Brain Sci. 12 (12) (2022) 1649.

[9] S. Job, X. Tao, T. Cai, H. Xie, L. Li, J. Yong, Q. Li, Exploring causal learning through graph neural networks: an in-depth review, 2023, arXiv preprint arXiv:2311.14994.

[10] Y. Gu, Y. Wang, H.-R. Zhang, J. Wu, X. Gu, Enhancing text classification by graph neural networks with multi-granular topic-aware graph, IEEE Access 11 (2023) 20169–20183.

[11] D. Yao, Z. Zhi-li, Z. Xiao-feng, C. Wei, H. Fang, C. Yao-ming, W.-W. Cai, Deep hybrid: multi-graph neural network collaboration for hyperspectral image classification, Def. Technol. 23 (2023) 164–176.

[12] W. Wu, X. Liu, R.B. Hamilton, A.A. Suriawinata, S. Hassanpour, Graph convolutional neural networks for histologic classification of pancreatic cancer, Arch. Pathol. Lab. Med. (2023).

[13] M. Zou, Z. Gan, R. Cao, C. Guan, S. Leng, Similarity-navigated graph neural networks for node classification, Inform. Sci. 633 (2023) 41–69.

[14] X. Li, Z. Fan, F. Huang, X. Hu, Y. Deng, L. Wang, X. Zhao, Graph neural network with curriculum learning for imbalanced node classification, Neurocomputing 574 (2024) 127229.

[15] Y. Li, H. Chen, Y. Li, L. Li, S.Y. Philip, G. Xu, Reinforcement learning based path exploration for sequential explainable recommendation, IEEE Trans. Knowl. Data Eng. 35 (11) (2023) 11801–11814.

[16] Y. Wang, D. Liang, L. Charlin, D.M. Blei, Causal inference for recommender systems, in: Proceedings of the 14th ACM Conference on Recommender Systems, 2020, pp. 426–431.

[17] L. Yao, Z. Ge, Causal variable selection for industrial process quality prediction via attention-based GRU network, Eng. Appl. Artif. Intell. 118 (2023) 105658.

[18] S. Wein, W.M. Malloni, A.M. Tomé, S.M. Frank, G.-I. Henze, S. Wüst, M.W. Greenlee, E.W. Lang, A graph neural network framework for causal inference in brain networks, Sci. Rep. 11 (1) (2021) 8061.

[19] A. Monken, F. Haberkorn, M. Gopinath, L. Freeman, F.A. Batarseh, Graph neural networks for modeling causality in international trade, in: The International FLAIRS Conference Proceedings, Vol. 34, 2021.

[20] J. Lin, Z. Li, Z. Li, L. Bai, R. Zhao, C. Zhang, Dynamic causal graph convolutional network for traffic prediction, 2023, arXiv preprint arXiv:2306.07019.

[21] L. Xiong, X. Yuan, Z. Hu, X. Huang, P. Huang, Gated fusion adaptive graph neural network for urban road traffic flow prediction, Neural Process. Lett. 56 (1) (2024) 9.

[22] Y. Li, X. Sun, H. Chen, S. Zhang, Y. Yang, G. Xu, Attention is not the only choice: counterfactual reasoning for path-based explainable recommendation, IEEE Trans. Knowl. Data Eng. (2024).

[23] R. Liu, Q. Zhang, D. Lin, W. Zhang, S.X. Ding, Causal intervention graph neural network for fault diagnosis of complex industrial processes, Reliab. Eng. Syst. Saf. 251 (2024) 110328.

[24] L. Liu, H. Lu, M. Whelan, Y. Chen, X. Ding, CiGNN: A causality-informed and graph neural network based framework for cuffless continuous blood pressure estimation, IEEE J. Biomed. Health Inf. (2024).

[25] X. Wang, Q. Li, D. Yu, W. Huang, Q. Li, G. Xu, Neural causal graph collaborative filtering, Inform. Sci. (2024) 120872.

[26] G. Lagani, F. Falchi, C. Gennaro, G. Amato, Comparing the performance of Hebbian against backpropagation learning using convolutional neural networks, Neural Comput. Appl. 34 (8) (2022) 6503–6519.

[27] F. Alemanno, M. Aquaro, I. Kanter, A. Barra, E. Agliari, Supervised hebbian learning, Europhys. Lett. 141 (1) (2023) 11001.

[28] T. Miconi, Hebbian learning with gradients: Hebbian convolutional neural networks with modern deep learning frameworks, 2021, arXiv preprint arXiv:2107.01729.

[29] F.J. Aguilar Canto, Convolutional neural networks with hebbian-based rules in online transfer learning, in: Advances in Soft Computing: 19th Mexican International Conference on Artificial Intelligence, MICAI 2020, Mexico City, Mexico, October 12–17, 2020, Proceedings, Part I 19, Springer, 2020, pp. 35–49.

[30] S. Eckmann, E.J. Young, J. Gjorgjieva, Synapse-type-specific competitive Hebbian learning forms functional recurrent networks, Proc. Natl. Acad. Sci. 121 (25) (2024) e2305326121.

[31] I.T. Ellwood, Short-term Hebbian learning can implement transformer-like attention, PLoS Comput. Biol. 20 (1) (2024) e1011843.

[32] L. Wu, P. Cui, J. Pei, L. Zhao, X. Guo, Graph neural networks: foundation, frontiers and applications, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 4840–4841.

[33] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2017, arXiv.org.

[34] DGL, Graph neural networks and its variants, 2018, https://docs.dgl.ai/en/0.8.x/tutorials/models.

[35] D.O. Hebb, The Organisation of Behaviour: A Neuropsychological Theory, Science Editions New York, 1949.

[36] Y. Munakata, J. Pfaffly, Hebbian learning and development, Dev. Sci. 7 (2) (2004) 141–148.

[37] J. Pearl, An introduction to causal inference, Int. J. Biostat. 6 (2) (2010).

[38] D. Bellamy, M.A. Hernán, A. Beam, A structural characterization of shortcut features for prediction, Eur. J. Epidemiol. 37 (6) (2022) 563–568.

[39] N. Wale, I.A. Watson, G. Karypis, Comparison of descriptor spaces for chemical compound retrieval and classification, Knowl. Inf. Syst. 14 (2008) 347–375.

[40] K.M. Borgwardt, C.S. Ong, S. Schönauer, S. Vishwanathan, A.J. Smola, H.-P. Kriegel, Protein function prediction via graph kernels, Bioinform. 21 (suppl_1) (2005) i47–i56.

[41] A.K. Debnath, R.L. Lopez de Compadre, G. Debnath, A.J. Shusterman, C. Hansch, Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity, J. Med. Chem. 34 (2) (1991) 786–797.

[42] Y. Zhou, H. Huo, Z. Hou, F. Bu, A deep graph convolutional neural network architecture for graph classification, Plos One 18 (3) (2023) e0279604.

[43] Y. Xie, S. Lv, Y. Qian, C. Wen, J. Liang, Active and semi-supervised graph neural networks for graph classification, IEEE Trans. Big Data 8 (4) (2022) 920–932.

[44] A.K. McCallum, K. Nigam, J. Rennie, K. Seymore, Automating the construction of internet portals with machine learning, Inf. Retr. (Boston) 3 (1) (2000) 127.

[45] C.L. Giles, K.D. Bollacker, S. Lawrence, CiteSeer: An automatic citation indexing system, in: Proceedings of the Third ACM Conference on Digital Libraries, DL '98, Association for Computing Machinery, New York, NY, USA, 1998, pp. 89–98, http://dx.doi.org/10.1145/276675.276685.

[46] P. Yanardag, S. Vishwanathan, Deep graph kernels, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1365–1374.

[47] P. Zhou, Z. Wu, G. Wen, K. Tang, J. Ma, Multi-scale graph classification with shared graph neural network, World Wide Web 26 (3) (2023) 949–966.

[48] H.-J. Moon, S.-B. Cho, A subgraph embedded GIN with attention for graph classification, in: International Conference on Intelligent Data Engineering and Automated Learning, Springer, 2023, pp. 356–367.

[49] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, Adv. Neural Inf. Process. Syst. 30 (2017).

[50] Z. Zhao, P. Wang, H. Wen, Y. Zhang, Z. Zhou, Y. Wang, A twist for graph classification: Optimizing causal information flow in graph neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 17042–17050.