NON-BACKTRACKING GRAPH NEURAL NETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

The celebrated message-passing updates for graph neural networks allow the representation of large-scale graphs with local and computationally tractable updates. However, the local updates suffer from backtracking, i.e., a message flows through the same edge twice and revisits the previously visited node. Since the number of message flows increases exponentially with the number of updates, the redundancy in local updates prevents the graph neural network from accurately recognizing a particular message flow for downstream tasks. In this work, we propose to resolve such a redundancy via the non-backtracking graph neural network (NBA-GNN) that updates a message without incorporating the message from the previously visited node. We further investigate how NBA-GNN alleviates the over-squashing of GNNs, and establish a connection between NBA-GNN and the impressive performance of non-backtracking updates for stochastic block model recovery. We empirically verify the effectiveness of our NBA-GNN on long-range graph benchmark and transductive node classification problems.

1 INTRODUCTION

Recently, graph neural networks (GNNs) (Kipf & Welling, 2016; Hamilton et al., 2017; Xu et al., 2019) have shown great success in various applications, including but not limited to, molecular property prediction (Gilmer et al., 2017) and community detection (Bruna & Li, 2017). Such success can be largely attributed to the message-passing structure of GNNs, which provides a computationally tractable way of incorporating the overall graph through iterative updates based on local neighborhoods. However, the message-passing structure also brings challenges due to the parallel updates and memory-less behavior of messages passed along the graph.

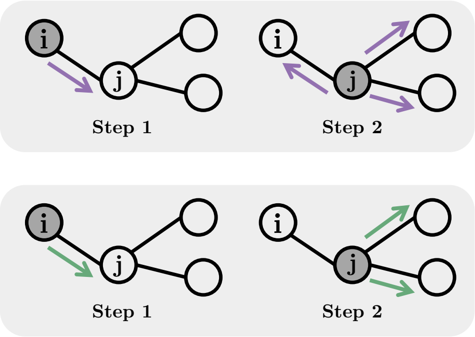


Figure 1: Message flows of simple (above) and non-backtracking (below) updates.

In particular, the message flow in a GNN is prone to backtracking, where the message from vertex i to vertex j is reincorporated in the subsequent message from j to i, e.g., Figure 1. Since the message-passing iteratively aggregates the information, the GNN inevitably encounters an exponential surge in the number of message flows, proportionate to the vertex degrees. This issue is compounded by backtracking, which accelerates the growth of message flows with redundant information.

Interestingly, despite these challenges, non-backtracking updates—a potential solution—have been largely overlooked in the existing GNN research, while they have been thoroughly investigated for non-GNN message-passing algorithms or random walks (Fitzner & van der Hofstad, 2013; Rappaport et al., 2017) (Figure 1). For example, given a pair of vertices i, j, the belief propagation algorithm (Pearl, 1982) forbids an $i \to j$ message from incorporating the $j \to i$ message. Another example is the non-backtracking random walks (Alon et al., 2007) which are non-Markovian walks that do not traverse the same edge twice and revisit the previous node. Such classic algorithms have demonstrated great success in applications like probabilistic graphical model inference and stochastic block models (Massoulié, 2014; Bordenave et al., 2015; Abbe & Sandon, 2015b). In particular, the spectrum of the non-backtracking operator contains more useful information than that of the adjacency matrix in revealing the hidden structure of a graph model (Bordenave et al., 2015).

Contribution. In this work, we propose the non-backtracking graph neural network (NBA-GNN) which employs non-backtracking updates on the messages, i.e., forbids the message from vertex i to vertex j from being incorporated in the message from vertex j to i. To this end, we associate the hidden features with transitions between a pair of vertices, e.g., $h_{j\to i}$, and update them from features associated with non-backtracking transitions, e.g., $h_{k\to j}$ for $k \neq i$.

To motivate our work, we formulate "message flows" as the sensitivity of a GNN with respect to walks in the graph. Then we explain how the message flows are redundant; the GNN's sensitivity of a walk with backtracking transitions can be covered by that of other non-backtracking walks. We explain how the redundancy is harmful to the GNN since the number of walks increases exponentially as the number of layers grows and the GNN becomes insensitive to a particular walk information. Hence, reducing the redundancy by only considering non-backtracking walks would benefit the message-passing updates to better recognize each walk's information. We further make a connection from our sensitivity analysis to the over-squashing phenomenon for GNNs (Topping et al., 2022; Black et al., 2023; Di Giovanni et al., 2023) in terms of access time.

Furthermore, we analyze our NBA-GNNs from the perspective of over-squashing and their expressive capability to recover sparse stochastic block models (SBMs). To this end, we prove that NBA-GNN improves the Jacobian-based measure of over-squashing (Topping et al., 2022) compared to its original GNN counterpart.

Next, we investigate NBA-GNN's proficiency in node classification within SBMs and its ability to distinguish between graphs originating from the Erdős–Rényi model or the SBM, from the results of (Stephan & Massoulié, 2022; Bordenave et al., 2015). Unlike traditional GNNs that operate on adjacency matrices and necessitate an average degree of at least $\Omega(\log n)$, NBA-GNN demonstrates the ability to perform node classification with a substantially lower average degree bound of $\omega(1)$ and $n^{o(1)}$. Furthermore, the algorithm can accurately classify graphs even when the average degree remains constant.

Finally, we empirically evaluate our NBA-GNN on the long-range graph benchmark (Dwivedi et al., 2022) and transductive node classification problems (Sen et al., 2008; Pei et al., 2019). We observe that our NBA-GNN demonstrates competitive performance and even achieves state-of-the-art performance on the long-rage graph benchmark. For the node classification tasks, we demonstrate that NBA-GNN consistently improves over its conventional GNN counterpart.

To summarize, our contributions are as follows:

- We propose NBA-GNN as a solution for the message flow redundancy problem in GNNs.
- We analyze how the NBA-GNN alleviates over-squashing and is expressive enough to recover sparse stochastic block models with an average degree of $o(\log n)$.
- We empirically verify our NBA-GNNs to show state-of-the-art performance on the long-range graph benchmark and consistently improve over the conventional GNNs across various tasks.

2 Related works

Non-backtracking algorithms. Non-backtracking updates have been considered by many classical algorithms (Newman, 2013; Kempton, 2016). Belief propagation (Pearl, 1982) infers the marginal distribution on probabilistic graphical models, and has demonstrated success for tree graphs (Kim & Pearl, 1983) and graphs with large girth (Murphy et al., 2013). Moreover, Mahé et al. (2004) and Aziz et al. (2013) suggest graph kernels between labeled graphs utilizing non-backtracking walks, and Krzakala et al. (2013) has first used it for node classification. Furthermore, the non-backtracking has been shown to yield better spectral separation properties, and its eigenspace contains information about the hidden structure of a graph model (Bordenave et al., 2015; Stephan & Massoulié, 2022).

Additionally, Chen et al. (2017) has first used the non-backtracking operator in GNNs, though limited to community detection tasks. Chen et al. (2022) has removed redundancy by computing non-redundant tree for every nodes, inevitably suffering high complexity. We highlight the differences between the related works and our NBA-GNN in Appendix A, due to limited space.

Analyzing over-squashing of GNNs. When a node receives information from a k-hop neighbor node, an exponential number of messages passes through node representations with fixed-sized vectors. This leads to the loss of information denoted as *over-squashing* (Alon & Yahav, 2020), and has been formalized in terms of sensitivity (Topping et al., 2022; Di Giovanni et al., 2023). Hence, sensitivity is defined as the Jacobian of a node feature at a GNN layer with respect to the input node, and can be upper bounded via the graph topology. Stemming from this, graph rewiring methods mitigate over-squashing by adding or removing edges for an optimal graph topology to compute (Topping et al., 2022; Black et al., 2023; Di Giovanni et al., 2023). Another line of work uses global aspects, e.g., connecting a virtual global node (Cai et al., 2023) or using graph Transformers (Ying et al., 2021; Kreuzer et al., 2021; Rampášek et al., 2022; Shirzad et al., 2023).

Expressivity of GNNs with the Weisfeiler-Lehman (WL) test. In the past few years, researchers have conducted significant studies on various aspects of the expressive power of GNNs. One line of research involves comparing GNNs with the WL test (Leman & Weisfeiler, 1968) to assess their expressiveness. For instance, Xu et al. (2019) demonstrated that MPNNs are at best as powerful as the WL test and introduced the graph isomorphism network (GIN), which matches the representational power of the WL test. Besides, Kanatsoulis & Ribeiro (2023) analyze the expressivity of GNNs utilizing linear algebraic tools and eigenvalue decomposition of graph operators.

Expressivity of GNNs for the stochastic block model (SBM). Furthermore, certain studies have analyzed the expressive power of GNNs using variations of the SBM (Holland et al., 1983). Fountoulakis et al. (2022) established conditions for the existence of graph attention networks (GATs) that can precisely classify nodes in the contextual stochastic block model (CSBM) with high probability. Similarly, Baranwal et al. (2022) investigated the effects of graph convolutions within a network on the XOR-CSBM. These works focused primarily on the probability distribution of node features, such as the distance between the means of feature vectors.

3 Non-backtracking graph neural network

3.1 Motivation from sensitivity analysis

We first explain how the conventional message-passing updates are prone to backtracking. To this end, consider a simple, undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{N}(i)$ denotes the neighbor of the node i. Each node $i \in \mathcal{V}$ is associated with a feature x_i. Then the conventional graph neural networks (GNNs), i.e., message-passing neural networks (MPNNs) (Gilmer et al., 2017), iteratively update the t-th layer node-wise hidden feature $h_i^{(t)}$ as follows:

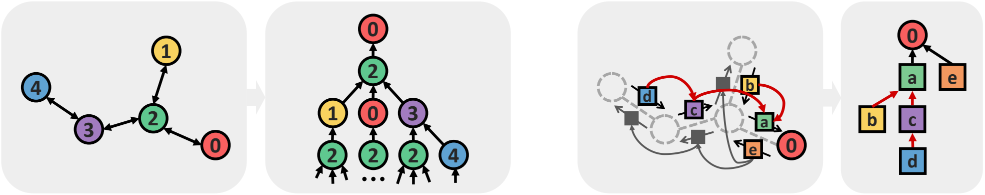
$$h_i^{(t+1)} = \phi^{(t)}\left(h_i^{(t)}, \left\{\psi^{(t)}\left(h_i^{(t)}, h_j^{(t)}\right) : j \in \mathcal{N}(i)\right\}\right), \tag{1}$$

where $\phi^{(t)}$ and $\psi^{(t)}$ are architecture-specific non-linear update and permutation invariant aggregation functions, respectively. Our key observation is that the message from the node feature $h_i^{(t)}$ to the node feature $h_j^{(t+1)}$ is reincorporated in the node feature $h_i^{(t+2)}$, e.g., Figure 3a shows the computation graph of conventional GNNs where redundant messages are incorporated.

Sensitivity analysis. To concretely describe the backtracking nature of message-passing updates, we formulate the sensitivity of the final node feature $h_i^{(T)}$ with respect to the input as follows:

$$\sum_{j\in\mathcal{V}} \frac{\partial h_i^{(T)}}{\partial h_j^{(0)}} = \sum_{s\in\mathcal{W}(i)} \prod_{t=1}^{T} \frac{\partial h_{s(t)}^{(t)}}{\partial h_{s(t-1)}^{(t-1)}}, \tag{2}$$

where $h_i^{(0)} = x_i$, $\mathcal{W}(i)$ denotes the set of T-step walks ending at node i, and $s(t)$ denotes the t-th node in the walk $s \in \mathcal{W}(i)$. Intuitively, this equation shows that a GNN with T layer recognizes the graph via aggregation of random walks with length T. Our key observation from Equation (2) is on how the feature $h_i^{(T)}$ is insensitive to the information to an initial node feature $h_j^{(0)}$, due to the information being "squashed" by the aggregation over the exponential number of walks $\mathcal{W}(i)$. A similar analysis has been conducted on how a node feature $h_i^{(T)}$ is insensitive to the far-away initial node feature $h_j^{(0)} = x_j$, i.e., the over-squashing phenomenon of GNNs (Topping et al., 2022).



(a) Computation graph of typical GNNs

(b) Computation graph of NBA-GNNs

Figure 3: Computation graph of typical GNN and NBA-GNN predicting node “0”. (a) Redundant messages increase the size of the computation graph. (b) NBA-GNN assigns a pair of features for each edge and update them via non-backtracking message passing, reducing redundant messages.

Redundancy of walks with backtracking. In particular, a walk s randomly sampled from $\mathcal{W}(i)$ is likely to contain a transition that backtracks, i.e., $s(t) = s(t+2)$ for some $t < T$. Then the walk s would be redundant since the information is contained in two other walks in $\mathcal{W}(i)$: $s(0), \ldots, s(t+1)$ and $s(0), \ldots, s(t+1), s(t) = s(t+2), s(t+3), \ldots, s(T)$. See Figure 2 for an illustration. This idea leads to the conclusion that non-backtracking walks, i.e., walks that do not contain backtracking transitions, are sufficient to express the information in the walks $\mathcal{W}(i)$. Since the exponential number of walks in $\mathcal{W}(i)$ causes the GNN to be insensitive to a particular walk information, it makes sense to design a non-backtracking GNN that is sensitive to the constrained set of non-backtracking walks. We note that similar concepts of redundancy for GNNs have been studied in prior works (Jia et al., 2020; Chen et al., 2022). See Appendix A for a detailed comparison.

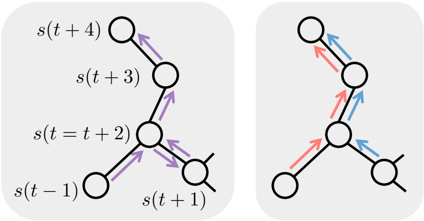


Figure 2: Two non-backtracking walks (right) are sufficient to express information in a walk with backtracking transition.

Relation to over-squashing. Finally, we point out an intriguing motivation for our work in terms of over-squashing. In particular, we note that Di Giovanni et al. (2023) analyzed the lower bound for the Jacobian obstruction that measures the degree of over-squashing in terms of access time with respect to a simple random walk. They conclude that the degree of over-squashing, i.e., the size of Jacobian obstruction, is higher for a pair of nodes with longer access time.

Hence, to design a GNN architecture robust to the over-squashing phenomenon, one could (i) propose a random walk that has shorter access time in general for a pair of nodes in the graph, and (ii) design a GNN that aligns with the optimized random walk. Non-backtracking random walks have been empirically shown and believed to generally yield faster access time than simple random walks (Lin & Zhang, 2019; Fasino et al., 2023). Therefore, one could aim to design a GNN architecture that aligns with the non-backtracking random walks.

However, to the best of our knowledge, there is no formal proof of scenarios where non-backtracking random walks yield a shorter access time. As a motivating example, we provide a theoretical result comparing the access time of non-backtracking and simple random walks for tree graphs.

Proposition 1. *Given a tree $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a pair of nodes $i, j \in \mathcal{V}$, the access time of begrudgingly backtracking random walk is equal or smaller than that of a simple random walk, where the equality holds if and only if the random walk length is 1.*

The begrudgingly backtracking random walk (Rappaport et al., 2017) modifies non-backtracking random walks to remove “dead ends” for tree graphs. Please refer to Appendix B for the proof.

3.2 Method description

In this section, we present our **N**on-**BA**cktracking GNNs (NBA-GNN) with the motivation described in Section 3.1. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, our NBA-GNN associates a pair of hidden features $h_{i \to j}^{(t)}, h_{j \to i}^{(t)}$ for each edge $\{i, j\}$. Then the non-backtracking message passing update for a hidden feature $h_{j\to i}^{(t)}$ is defined as follows:

$$h_{j\to i}^{(t+1)} = \phi^{(t)}\left(h_{j\to i}^{(t)}, \left\{\psi^{(t)}\left(h_{k\to j}^{(t)}, h_{j\to i}^{(t)}\right) : k \in \mathcal{N}(j)\setminus\{i\}\right\}\right), \tag{3}$$

where $\phi^{(t)}$ and $\psi^{(t)}$ are backbone-specific non-linear update and permutation-invariant aggregation functions at the t-th layer, respectively. For example, $\psi^{(t)}$ and $\phi^{(t)}$ are multi-layer perceptron and summation over a set for the graph isomorphism network (Xu et al., 2019), respectively. Given the update in Equation (3), one can observe that the message $h_{i\to j}^{(t)}$ is never incorporated in the message $h_{j\to i}^{(t+1)}$, and hence the update is free from backtracking.

Initialization and node-wise aggregation of messages. The message at the 0-th layer $h_{i\to j}^{(0)}$ is initialized by encoding the node features x_i, x_j, and the edge feature e_{ij} using a non-linear function ϕ. After updating hidden features for each edge based on Equation (3), we apply permutation-invariant pooling over all the messages for graph-wise predictions. Since we use hidden features for each edge, we construct the node-wise predictions at the final layer as follows:

$$h_i = \sigma\left(\rho\left\{h_{j\to i}^{(T)} : j \in \mathcal{N}(i)\right\}, \rho\left\{h_{i\to j}^{(T)} : j \in \mathcal{N}(i)\right\}\right), \tag{4}$$

where σ is a non-linear aggregation function with different weights for incoming edges $j \to i$ and outgoing edges $i \to j$, ρ is a non-linear aggregation function invariant to the permutation of nodes in $\mathcal{N}(i)$. We provide a computation graph of NBA-GNNs in Figure 3b to summarize our algorithm.

Begrudgingly backtracking update. While the messages from our update are resistant to backtracking, a message $h_{j\to i}^{(t)}$ may get trapped in node i for the special case when $\mathcal{N}(i) = \{j\}$. To resolve this issue, we introduce a simple trick coined begrudgingly backtracking update (Rappaport et al., 2017) that updates $h_{i\to j}^{(t+1)}$ using $h_{j\to i}^{(t)}$ only when $\mathcal{N}(i) = \{j\}$. We empirically verify the effectiveness of begrudgingly backtracking updates in Section 5.3.

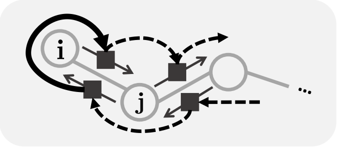


Figure 4: Begrudgingly backtracking update (solid).

Implementation. To better understand our NBA-GNN, we provide an example of non-backtracking message-passing updates with a GCN backbone (Kipf & Welling, 2016), coined NBA-GCN. The message update at the t-th layer of NBA-GCN can be written as follows:

$$h_{j\to i}^{(t+1)} = h_{j\to i}^{(t)} + \sigma_{\text{GCN}}\left(\frac{1}{|\mathcal{N}(j)|-1}\mathbf{W}^{(t)} \sum_{k\in\mathcal{N}(j)\setminus\{i\}} h_{k\to j}^{(t)}\right), \tag{5}$$

where σ_{GCN} is an element-wise nonlinear function, e.g., rectified linear unit (Agarap, 2018), $\mathbf{W}^{(t)}$ is the weight matrix, and the messages are normalized by their population $|\mathcal{N}(j)| - 1$.

4 Theoretical analysis

In this section, we analyze the proposed NBA-GNN framework. To be specific, we show that (a) our NBA-GNN improves the upper bound for sensitivity-based measures of GNN over-squashing and (b) NBA-GNNs can detect the underlying structure of SBMs even for very sparse graphs.

4.1 Sensitivity analysis on over-squashing

We first analyze how NBA-GNNs alleviate the over-squashing issue. To this end, following Topping et al. (2022), we use the Jacobian of node-wise output with respect to another node initial feature as a way to assess over-squashing effect. We first derive the Jacobian quantification of over-squashing for the NBA-GNNs, and then demonstrate that the upper bound for Jacobian is larger than that of the conventional GNNs derived by Topping et al. (2022).

To derive our analysis, we introduce the non-backtracking matrix $B \in \{0, 1\}^{2|\mathcal{E}|\times 2|\mathcal{E}|}$ and the incidence matrix $C \in \mathbb{R}^{2|\mathcal{E}|\times|\mathcal{V}|}$ which describe the NBA-GNN message-passing and node-wise aggregation via linear operation, respectively. To be specific, the backtracking matrix B and the incidence matrix C are defined as follows:

$$B_{(\ell\to k),(j\to i)} = \begin{cases} 1 & \text{if } k=j, \ell\neq i \\ 0 & \text{otherwise} \end{cases}, \qquad C_{(k\to j),i} = \begin{cases} 1 & \text{if } j=i \text{ or } k=i \\ 0 & \text{otherwise} \end{cases}.$$

We also define D_{out} and D_{in} as the out-degree and in-degree matrices of NBA-GNN, respectively, counting the number of outgoing and incoming edges for each edge. These are diagonal matrices with $(D_{out})_{(j\to i),(j\to i)} = \sum_{\ell\to k} B_{(j\to i),(\ell\to k)}$ and $(D_{in})_{(j\to i),(j\to i)} = \sum_{\ell\to k} B_{(\ell\to k),(j\to i)}$. Next, we introduce $\widehat{B}$ as the normalized non-backtracking matrix augmented with self-loops: $\widehat{B} = (D_{out}+I)^{-\frac{1}{2}}(B+I)(D_{in}+I)^{-\frac{1}{2}}$. Finally, we let $\tilde{C}$ denote a matrix where $\tilde{C}_{(k\to j),i} = C_{(k\to j),i} + C_{(j\to k),i}$. Then, one obtains the following sensitivity bound of NBA-GNN.

Lemma 1. *Consider two nodes $i, j \in \mathcal{V}$ with distance T given a $(T-1)$-layer NBA-GNN as described in Equation (3) and Equation (4). Suppose $|\nabla\phi^{(t)}|, |\nabla\sigma| \leq \alpha$, $|\nabla\psi^{(t)}|, |\nabla\rho| \leq \beta$, and $|\nabla\phi| \leq \gamma$ for $0 \leq t < T$. Then the following holds:*

$$\left\| \frac{\partial h_j}{\partial x_i} \right\| \leq (\alpha\beta)^T \gamma (\tilde{C}^\top \widehat{B}^{(T-1)} \tilde{C})_{j,i}.$$

We provide the proof in Appendix C. Lemma 1 states how *the over-squashing effect is controlled by the power of $\widehat{B}$*. Consequently, one can infer that increasing the upper bound likely leads to mitigation of the GNN over-squashing effect (Topping et al., 2022).

From this motivation, we provide an analysis to support our claim on how the NBA-GNN suffers less from over-squashing effect due to its larger sensitivity bound.

Theorem 1. *Let $\widehat{A}$ denote the degree-normalized matrix. Then, for any pair of nodes $i, j \in \mathcal{V}$ with distance T, the sensitivity bound of NBA-GNN is larger than that of conventional GNNs (Topping et al., 2022), i.e., $(\tilde{C}^\top \widehat{B}^{T-1} \tilde{C})_{j,i} \geq (\widehat{A}^T)_{j,i}$. For d-regular graphs, $(\tilde{C}^\top \widehat{B}^{T-1} \tilde{C})_{j,i}$ decays slower by $O(d^{-T})$, while $(\widehat{A}^T)_{j,i}$ decays with $O((d+1)^{-T})$.*

Note that $(\alpha\beta)^T(\widehat{A}^T)_{j,i}$ provides an upper bound for the sensitivity term in conventional GNNs (Topping et al., 2022). We provide the proof in Appendix C. Our proof is based on comparing the degree-normalized number of non-backtracking and simple walks from node i to node j.

4.2 Expressive power of NBA-GNN in the lens of SBMs

In the literature on the expressive capabilities of GNNs, comparisons with the well-known k-WL test are common. However, even when certain models surpass the k-WL test in performance, evaluating their relative merits remains a nontrivial task. Furthermore, due to the substantial performance gap between the 1-WL and 3-WL tests, many algorithms fall into a range between these two tests, making it more difficult to compare them with each other. It is also worth noting that comparing GNNs with the WL test does not always accurately reflect their performance on real-world datasets.

To address these issues, several studies have turned to spectral analysis of GNNs. From a spectral viewpoint, GNNs can be seen as functions of the eigenvectors and eigenvalues of the given graph. NT & Maehara (2019) showed that GNNs operate as low-pass filters on the graph spectrum, and Balcilar et al. (2020) analyzed the use of various GNNs as filters to extract the relevant graph spectrum and measure their expressive power. Moreover, Oono & Suzuki (2020) argue that the expressive power of GNNs is influenced by the topological information contained in the graph spectrum.

The eigenvalues and the corresponding adjacency matrix eigenvectors play a pivotal role in establishing the fundamental limits of community detection in SBM, as evidenced by Abbe et al. (2015), Abbe & Sandon (2015a), Hajek et al. (2016), Yun & Proutiere (2016), and Yun & Proutière (2019). The adjacency matrix exhibits a spectral separation property, and an eigenvector containing information about the assignments of the vertex community becomes apparent (Lei & Rinaldo, 2015). Furthermore, by analyzing the eigenvalues of the adjacency matrix, it is feasible to determine whether a graph originates from the Erdős–Rényi (ER) model or the SBM (Erdős et al., 2013; Avrachenkov et al., 2015). However, these spectral properties are particularly salient when the average degree of the graph satisfies $\Omega(\log n)$. For graphs with average degrees $o(\log n)$, vertices with higher degrees

Table 1: Comparison of conventional MPNNs and GNNs in the long-range graph benchmark, with and without Laplacian positional encoding. We also denote the relative improvement by Imp.

Model	Peptides-func		Peptides-struct		PascalVOC-SP	
	AP ↑	Imp.	MAE ↓	Imp.	F1 ↑	Imp.
GCN	0.5930 ± 0.0023		0.3496 ± 0.0013		0.1268 ± 0.0060	
+ NBA	0.6951 ± 0.0024	+17%	0.2656 ± 0.0009	+22%	0.2537 ± 0.0054	+100%
+ NBA+LapPE	**0.7206 ± 0.0028**	+22%	**0.2472 ± 0.0008**	+29%	**0.3005 ± 0.0010**	+137%
GIN	0.5498 ± 0.0079		0.3547 ± 0.0045		0.1265 ± 0.0076	
+ NBA	0.6961 ± 0.0045	+27%	0.2534 ± 0.0025	+29%	0.3040 ± 0.0119	+140%
+ NBA+LapPE	**0.7071 ± 0.0067**	+29%	**0.2424 ± 0.0010**	+32%	**0.3223 ± 0.0010**	+155%
GatedGCN	0.5864 ± 0.0077		0.3420 ± 0.0013		0.2873 ± 0.0219	
+ NBA	0.6429 ± 0.0062	+10%	0.2539 ± 0.0011	+26%	0.3910 ± 0.0010	+36%
+ NBA+LapPE	**0.6982 ± 0.0014**	+19%	**0.2466 ± 0.0012**	+28%	**0.3969 ± 0.0027**	+38%

predominate, affecting eigenvalues and complicating the discovery of the underlying structure of the graph (Benaych-Georges et al., 2019).

In contrast, the non-backtracking matrix exhibits several advantageous properties even for constant-degree cases. In (Stephan & Massoulié, 2022), the non-backtracking matrix demonstrates a spectral separation property and establishes the presence of an eigenvector containing information about vertex community assignments, when the average degree only satisfies $\omega(1)$ and $n^{o(1)}$. Furthermore, Bordenave et al. (2015) have demonstrated that by inspecting the eigenvalues of the non-backtracking matrix, it is possible to discern whether a graph originates from the ER model or the SBM, even when the graph's average degree remains constant. This capability sets NBA-GNNs apart and enhances their performance in both node and graph classification tasks, especially in sparse settings. These lines of reasoning lead to the formulation of the following theorem.

Theorem 2. ***(Informal)*** *Assume that the average degree in the stochastic block model satisfies the conditions of being at least $\omega(1)$ and $n^{o(1)}$. In such a scenario, the NBA-GNN can map from graph $\mathcal{G}$ to node labels.*

Theorem 3. ***(Informal)*** *Suppose we have a pair of graphs with a constant average degree, one generated from the stochastic block model and the other from the Erdős–Rényi model. In this scenario, the NBA-GNN is capable of distinguishing between them.*

For an in-depth exploration of this argument, please refer to Appendix D. The rationale behind these valuable properties of the non-backtracking matrix B in sparse scenarios lies in the fact that the matrix B^k exhibits similarity to the k-hop adjacency matrix, while A^k is mainly influenced by high-degree vertices. For these reasons, NBA-GNNs would outperform traditional GNNs in both node and graph classification tasks, particularly in sparse graph environments.

5 Experiment

In this section, we assess the effectiveness of NBA-GNNs across multiple benchmarks on graph classification, graph regression, and node classification tasks. Our method shows competitive performance compared to graph Transformers within long-range graph benchmarks, and robust improvements in handling transductive node classification tasks. We also conduct ablation studies to verify our method, and provide experimental details in Appendix E.

5.1 Long-range graph benchmark

The long-range graph benchmark (LRGB) (Dwivedi et al., 2022) is a set of tasks that require learning long-range interactions. We validate our method using three datasets from the LRGB benchmark: graph classification (`Peptides-func`), graph regression (`Peptides-struct`), and node classification (`PascalVOC-SP`). We use performance scores from (Dwivedi et al., 2022) and from each baseline papers: subgraph based GNNs (Abu-El-Haija et al., 2019; Michel et al., 2023; Giusti et al.,

Table 2: Evaluation of NBA-GNN on the LRGB benchmark. We color the **first-**, **second-** and **third-**best results. Performance within a standard deviation of one another is considered equal. Non-reported values are denoted by -.

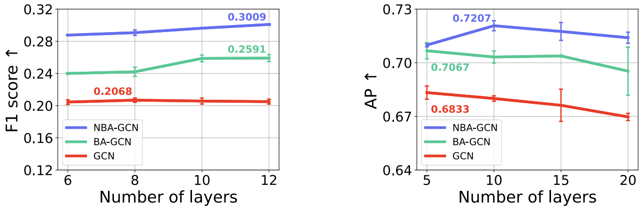
Method	Model	`Peptides-func` AP ↑	`Peptides-struct` MAE ↓	`VOC-SP` F1 ↑
GNNs	GCN	0.5930 ± 0.0023	0.3496 ± 0.0013	0.1268 ± 0.0060
	GIN	0.5498 ± 0.0079	0.3547 ± 0.0045	0.1265 ± 0.0076
	GatedGCN	0.5864 ± 0.0077	0.3420 ± 0.0013	0.2873 ± 0.0219
	GatedGCN+PE	0.6069 ± 0.0035	0.3357 ± 0.0006	0.2860 ± 0.0085
Subgraph GNNs	MixHop-GCN	0.6592 ± 0.0036	0.2921 ± 0.0023	0.2506 ± 0.0133
	MixHop-GCN+LapPE	0.6843 ± 0.0049	0.2614 ± 0.0023	0.2218 ± 0.0174
	PathNN	0.6816 ± 0.0026	0.2545 ± 0.0032	-
	CIN++	0.6569 ± 0.0117	0.2523 ± 0.0013	-
Transformers	Transformer+LapPE	0.6326 ± 0.0126	0.2529 ± 0.0016	0.2694 ± 0.0098
	GraphGPS+LapPE	0.6535 ± 0.0041	0.2500 ± 0.0005	**0.3748 ± 0.0109**
	SAN+LapPE	0.6384 ± 0.0121	0.2683 ± 0.0043	0.3230 ± 0.0039
	Exphormer	0.6527 ± 0.0043	0.2481 ± 0.0007	**0.3966 ± 0.0027**
	Graph MLP-Mixer/ViT	0.6970 ± 0.0080	**0.2449 ± 0.0016**	-
Rewiring methods	DIGL+MPNN	0.6469 ± 0.0019	0.3173 ± 0.0007	0.2824 ± 0.0039
	DIGL+MPNN+LapPE	0.6830 ± 0.0026	0.2616 ± 0.0018	0.2921 ± 0.0038
	DRew-GCN+LapPE	**0.7150 ± 0.0044**	0.2536 ± 0.0015	0.1851 ± 0.0092
	DRew-GIN+LapPE	**0.7126 ± 0.0045**	0.2606 ± 0.0014	0.2692 ± 0.0059
	DRew-GatedGCN+LapPE	0.6977 ± 0.0026	0.2539 ± 0.0007	0.3314 ± 0.0024
NBA-GNNs (Ours)	NBA-GCN	0.6951 ± 0.0024	0.2656 ± 0.0009	0.2537 ± 0.0054
	NBA-GCN+LapPE	**0.7207 ± 0.0028**	**0.2472 ± 0.0008**	0.3005 ± 0.0010
	NBA-GIN	0.6961 ± 0.0045	0.2775 ± 0.0057	0.3040 ± 0.0119
	NBA-GIN+LapPE	**0.7071 ± 0.0067**	**0.2424 ± 0.0010**	0.3223 ± 0.0063
	NBA-GatedGCN	0.6429 ± 0.0062	0.2539 ± 0.0011	**0.3910 ± 0.0010**
	NBA-GatedGCN+LapPE	0.6982 ± 0.0014	**0.2466 ± 0.0012**	**0.3969 ± 0.0027**

Table 3: Comparison of conventional GNNs and their NBA-GNN counterpart on transductive node classification tasks, with and without positional encoding. We mark the best numbers in bold.

Model	`Cora`	`CiteSeer`	`PubMed`	`Texas`	`Wisconsin`	`Cornell`
GCN	0.8658±0.0060	0.7532±0.0134	0.8825±0.0042	0.6162±0.0634	0.6059±0.0438	0.5946±0.0662
+NBA	**0.8722±0.0095**	0.7585±0.0175	0.8826±0.0044	**0.7108±0.0796**	**0.7471±0.0386**	0.6108±0.0614
+NBA+LapPE	0.8720±0.0129	**0.7609±0.0186**	**0.8827±0.0048**	0.6811±0.0595	**0.7471±0.0466**	**0.6378±0.0317**
GraphSAGE	0.8632±0.0158	0.7559±0.0161	0.8864±0.0030	0.7108±0.0556	0.7706±0.0403	0.6027±0.0625
+NBA	**0.8702±0.0083**	0.7586±0.0213	**0.8871±0.0044**	0.7270±0.0905	**0.7765±0.0508**	**0.6459±0.0691**
+NBA+LapPE	0.8650±0.0120	**0.7621±0.0172**	0.8870±0.0037	**0.7486±0.0612**	0.7647±0.0531	0.6378±0.0544
GAT	0.8694±0.0119	0.7463±0.0159	0.8787±0.0046	0.6054±0.0386	0.6000±0.0491	0.4757±0.0614
+NBA	**0.8722±0.0120**	0.7549±0.0171	**0.8829±0.0043**	0.6622±0.0514	0.7059±0.0562	**0.5838±0.0558**
+NBA+LapPE	0.8692±0.0098	**0.7561±0.0175**	0.8822±0.0047	**0.6730±0.0348**	**0.7314±0.0531**	0.5784±0.0640
GatedGCN	0.8477±0.0156	0.7325±0.0192	**0.8671** ±0.0060	0.6108±0.0652	0.5824±0.0641	0.5216±0.0987
+NBA	**0.8523±0.0095**	**0.7405±0.0187**	0.8661±0.0035	0.6162±0.0490	0.6431±0.0356	**0.5649±0.0532**
+NBA+LapPE	0.8517±0.0130	0.7379±0.0193	0.8661±0.0047	**0.6243±0.0467**	**0.6569±0.0310**	0.5405±0.0785

2023), graph Transformers (Kreuzer et al., 2021; Rampášek et al., 2022; Shirzad et al., 2023; He et al., 2023), and graph rewiring methods (Gasteiger et al., 2019; Gutteridge et al., 2023). For NBA-GNNs and NBA-GNNs with begrudgingly backtracking, we report the one with better performance. Furthermore, LapPE, i.e., Laplacian positional encoding (Dwivedi et al., 2020), is applied, as it enhances the performance of NBA-GNNs in common cases.

As one can see in Table 1, NBA-GNNs show improvement regardless of the combined backbone GNNs, i.e., GCN (Kipf & Welling, 2016), GIN (Xu et al., 2019), and GatedGCN (Bresson & Laurent, 2017). Furthermore, even against a variety of baselines in Table 2, at least one NBA-GNNs shows competitive performance with the best baseline in LRGB. Also, it is noteworthy that the improvement of NBA-GNNs is higher in dense graphs, `PascalVOC-SP` having an average degree of 8 while `Peptides-func`, `Peptides-struct` has an average degree of 2.



Model	BG.	AP ↑	MAE ↓
GCN	✗	0.7015	**0.2472**
	✓	**0.7207**	0.2547
GIN	✗	0.6825	**0.2424**
	✓	**0.7071**	0.2479
GatedGCN	✗	0.6710	**0.2466**
	✓	**0.6982**	0.2489

(a) F1 for changes in number of layers in PascalVOC-SP

(b) AP for changes in number of layers in Peptides-func

(c) Performance of begrudgingly, non-backtracking in sparse graphs

Figure 5: Ablation studies on the components of NBA-GNN.

5.2 Transductive node classification tasks

We conduct experiments on three citation networks (Cora, CiteSeer, and Pubmed) (Sen et al., 2008), and three heterophilic datasets (Texas, Wisconsin, and Cornell) (Pei et al., 2019) for transductive node classification. In our experiments, we employed various GNN architectures (Kipf & Welling, 2016; Hamilton et al., 2017; Veličković et al., 2018; Li et al., 2016), along with their corresponding NBA versions, as illustrated in Table 3. The results indicate that the NBA operation improves the performance of all GNN variants. Specifically, it demonstrates significant enhancements, particularly on heterophilic datasets. Given that related nodes in heterophilic graphs are often widely separated (Zheng et al., 2022), the ability of NBA-GNN to alleviate over-squashing plays a vital role in classifying nodes in such scenarios.

5.3 Ablation studies

Here, we conduct ablation studies to empirically verify our framework. For simplicity, we use BA for backtracking GNNs and BG for begrudgingly backtracking GNNs. All experiments are averaged over 3 seeds. We use hyper-parameters for GCN from Tönshoff et al. (2023).

Non-backtracking vs. backtracking GNNs. We first verify whether the performance improvements indeed stem from the non-backtracking updates. To this end, we compare our NBA-GNN with a backtracking variant, coined BA-GNN. To be specific, BA-GNN allows backtracking update prohibited in NBA-GNN , i.e., using $h_{i\to j}^{(\ell)}$ to update $h_{j\to i}^{(\ell+1)}$. From Figures 5a and 5b, one can observe how NBA-GNN outperforms the BA-GNN consistently irrelevant to the numbers of layers. Intriguingly, one can also observe that BA-GNN outperforms the naïve backbone, i.e., GCN, consistently.

Begrudgingly backtracking updates in sparse graphs. Additionally, we investigate the effect of begrudgingly backtracking in sparse graphs, i.e., Peptides-func, and Peptides-struct, in Figure 5c. One can see that begrudgingly backtracking is effective in Peptides-func, and shows similar performance in Peptides-struct (Note that the PascalVOC-SP does not have a vertex with degree one).

6 Conclusion

We have introduced message passing framework applicable to any GNN architectures, bringing non-backtracking into graph neural networks. As theoretically shown, NBA-GNNs mitigate over-squashing in terms of sensitivity, and enhance its expressive power for both node and graph classification tasks on SBMs. Additionally, we demonstrate that NBA-GNNs achieve competitive performance on the LRGB benchmark and outperform conventional GNNs across various tasks.

Limitations. The space complexity of our framework is larger than the conventional GNNs. It creates $2|\mathcal{E}|$ messages considering directions, and $\sum_{i\in\mathcal{V}}\binom{d_i}{2}$ connections between them where d_i is the degree of a node i. The time complexity is also $\sum_{i\in\mathcal{V}}\binom{d_i}{2}$, since we do not add additional message-passing to the backbone GNN. We provide more detail of the space and time complexity in Appendix F. Therefore, investigation of space complexity reduction for our work would be an interesting future work.

Ethics Statement. The authors are committed to upholding the ICLR Code of Ethics and maintaining the highest ethical standards in research and scholarly activities. Additionally, it is crucial to acknowledge the potential ethical considerations associated with the utilization of our model. Although the model can offer valuable insights and advantages, there exists a possibility of misuse or unintended consequences. For instance, it could be exploited to extract sensitive information, such as affiliations or political inclinations, from social networks. Therefore, it is important to apply such models carefully and actively work towards preventing any adverse side effects. Responsible implementation is essential to ensure that the potential benefits of our work are maximized while minimizing any potential risks or adverse implications.

Reproducibility. The code for all experiments conducted in this paper is included in the accompanying zip file. We also provide comprehensive proofs for the theoretical analysis in Appendices B, C, and D. Further information about the experiments, including details about the datasets and models, can be found in Section 5. For a deeper dive into experiment specifics such as hyperparameter settings, please refer to Appendix E.

REFERENCES

Emmanuel Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

Emmanuel Abbe and Colin Sandon. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 670–688. IEEE, 2015a.

Emmanuel Abbe and Colin Sandon. Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic bp, and the information-computation gap. *arXiv preprint arXiv:1512.09080*, 2015b.

Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *IEEE Transactions on information theory*, 62(1):471–487, 2015.

Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, pp. 21–29. PMLR, 2019.

Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

Noga Alon, Itai Benjamini, Eyal Lubetzky, and Sasha Sodin. Non-backtracking random walks mix faster. *Communications in Contemporary Mathematics*, 9(04):585–603, 2007.

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.

Konstantin Avrachenkov, Laura Cottatellucci, and Arun Kadavankandy. Spectral properties of random matrices for stochastic block model. In *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pp. 537–544. IEEE, 2015.

Furqan Aziz, Richard C Wilson, and Edwin R Hancock. Backtrackless walks on a graph. *IEEE transactions on neural networks and learning systems*, 24(6):977–989, 2013.

Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. Analyzing the expressive power of graph neural networks in a spectral perspective. In *International Conference on Learning Representations*, 2020.

Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Effects of graph convolutions in deep networks. *arXiv preprint arXiv:2204.09297*, 2022.

Florent Benaych-Georges, Charles Bordenave, and Antti Knowles. Largest eigenvalues of sparse inhomogeneous erdős–rényi graphs. *The Annals of Probability*, 47(3):1653–1676, 2019.

Mitchell Black, Zhengchao Wan, Amir Nayyeri, and Yusu Wang. Understanding oversquashing in gnns through the lens of effective resistance. In *International Conference on Machine Learning*, pp. 2528–2547. PMLR, 2023.

Charles Bordenave, Marc Lelarge, and Laurent Massoulié. Non-backtracking spectrum of random graphs: community detection and non-regular ramanujan graphs. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 1347–1357. IEEE, 2015.

Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

Joan Bruna and X Li. Community detection with graph neural networks. *stat*, 1050:27, 2017.

Chen Cai, Truong Son Hy, Rose Yu, and Yusu Wang. On the connection between mpnn and graph transformer. *arXiv preprint arXiv:2301.11956*, 2023.

Rongqin Chen, Shenghui Zhang, Ye Li, et al. Redundancy-free message passing for graph neural networks. *Advances in Neural Information Processing Systems*, 35:4316–4327, 2022.

Zhengdao Chen, Xiang Li, and Joan Bruna. Supervised community detection with line graph neural networks. *arXiv preprint arXiv:1705.08415*, 2017.

Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Lio, and Michael M Bronstein. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In *International Conference on Machine Learning*, pp. 7865–7885. PMLR, 2023.

Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.

Vijay Prakash Dwivedi, Ladislav Rampášek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. *Advances in Neural Information Processing Systems*, 35:22326–22340, 2022.

László Erdős, Antti Knowles, Horng-Tzer Yau, and Jun Yin. Spectral statistics of erdős—rényi graphs i: Local semicircle law. *The Annals of Probability*, pp. 2279–2375, 2013.

Dario Fasino, Arianna Tonetto, and Francesco Tudisco. Hitting times for non-backtracking random walks. *arXiv preprint arXiv:2105.14438*, 2021.

Dario Fasino, Arianna Tonetto, and Francesco Tudisco. Hitting times for second-order random walks. *European Journal of Applied Mathematics*, 34(4):642–666, 2023.

Robert Fitzner and Remco van der Hofstad. Non-backtracking random walk. *Journal of Statistical Physics*, 150:264–284, 2013.

Kimon Fountoulakis, Amit Levi, Shenghao Yang, Aseem Baranwal, and Aukosh Jagannath. Graph attention retrospective. *arXiv preprint arXiv:2202.13060*, 2022.

Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. *Advances in neural information processing systems*, 32, 2019.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Lorenzo Giusti, Teodora Reu, Francesco Ceccarelli, Cristian Bodnar, and Pietro Liò. Cin++: Enhancing topological message passing. *arXiv preprint arXiv:2306.03561*, 2023.

Benjamin Gutteridge, Xiaowen Dong, Michael M Bronstein, and Francesco Di Giovanni. Drew: Dynamically rewired message passing with delay. In *International Conference on Machine Learning*, pp. 12252–12267. PMLR, 2023.

Bruce Hajek, Yihong Wu, and Jiaming Xu. Achieving exact cluster recovery threshold via semidefinite programming. *IEEE Transactions on Information Theory*, 62(5):2788–2797, 2016.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *NeurIPS*, 30, 2017.

Xiaoxin He, Bryan Hooi, Thomas Laurent, Adam Perold, Yann LeCun, and Xavier Bresson. A generalization of vit/mlp-mixer to graphs. In *International Conference on Machine Learning*, pp. 12724–12745. PMLR, 2023.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. Ogb-lsc: A large-scale challenge for machine learning on graphs. *arXiv preprint arXiv:2103.09430*, 2021.

Zhihao Jia, Sina Lin, Rex Ying, Jiaxuan You, Jure Leskovec, and Alex Aiken. Redundancy-free computation for graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 997–1005, 2020.

Charilaos I. Kanatsoulis and Alejandro Ribeiro. Representation power of graph neural networks: Improved expressivity via algebraic analysis, 2023.

Mark Condie Kempton. Non-backtracking random walks and a weighted ihara's theorem. *Open Journal of Discrete Mathematics*, 2016.

JinHyung Kim and Judea Pearl. A computational model for causal and diagnostic reasoning in inference systems. In *International Joint Conference on Artificial Intelligence*, pp. 0–0, 1983.

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

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.

Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.

Jing Lei and Alessandro Rinaldo. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, pp. 215–237, 2015.

AA Leman and Boris Weisfeiler. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya*, 2(9):12–16, 1968.

Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. Gated graph sequence neural networks. In *Proceedings of ICLR'16*, 2016.

Yuan Lin and Zhongzhi Zhang. Non-backtracking centrality based random walk on networks. *The Computer Journal*, 62(1):63–80, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Pierre Mahé, Nobuhisa Ueda, Tatsuya Akutsu, Jean-Luc Perret, and Jean-Philippe Vert. Extensions of marginalized graph kernels. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 70, 2004.

Laurent Massoulié. Community detection thresholds and the weak ramanujan property. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 694–703, 2014.

Gaspard Michel, Giannis Nikolentzos, Johannes F Lutzeyer, and Michalis Vazirgiannis. Path neural networks: Expressive and accurate graph neural networks. In *International Conference on Machine Learning*, pp. 24737–24755. PMLR, 2023.

Kevin Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. *arXiv preprint arXiv:1301.6725*, 2013.

MEJ Newman. Spectral community detection in sparse networks. *arXiv preprint arXiv:1308.6494*, 2013.

Hoang NT and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020.

Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 133–136, 1982.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2019.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

Brian Rappaport, Anuththari Gamage, and Shuchin Aeron. Faster clustering via non-backtracking random walks. *stat*, 1050:26, 2017.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *International Conference on Machine Learning*, 2023.

Ludovic Stephan and Laurent Massoulié. Non-backtracking spectra of weighted inhomogeneous random graphs. *Mathematical Statistics and Learning*, 5(3):201–271, 2022.

Jan Tönshoff, Martin Ritzert, Eran Rosenbluth, and Martin Grohe. Where did the gap go? reassessing the long-range graph benchmark. *arXiv preprint arXiv:2309.00367*, 2023.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=7UmjRGzp-A.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

Se-Young Yun and Alexandre Proutiere. Optimal cluster recovery in the labeled stochastic block model. *Advances in Neural Information Processing Systems*, 29, 2016.

Se-Young Yun and Alexandre Proutière. Optimal sampling and clustering in the stochastic block model. *Advances in Neural Information Processing Systems*, 32, 2019.

Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S Yu. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082*, 2022.