FairGT: A Fairness-aware Graph Transformer

Renqiang Luo[1] , **Huafei Huang**[1] , **Shuo Yu**[1,*] , **Xiuzhen Zhang**[2] and **Feng Xia**[2]
[1]Dalian University of Technology, China
[2]RMIT University, Australia
{lrenqiang, hhuafei}@outlook.com, {shuo.yu, f.xia}@ieee.org, xiuzhen.zhang@rmit.edu.au

Abstract

The design of Graph Transformers (GTs) generally neglects considerations for fairness, resulting in biased outcomes against certain sensitive subgroups. Since GTs encode graph information without relying on message-passing mechanisms, conventional fairness-aware graph learning methods cannot be directly applicable to address these issues. To tackle this challenge, we propose FairGT, a Fairness-aware Graph Transformer explicitly crafted to mitigate fairness concerns inherent in GTs. FairGT incorporates a meticulous structural feature selection strategy and a multi-hop node feature integration method, ensuring independence of sensitive features and bolstering fairness considerations. These fairness-aware graph information encodings seamlessly integrate into the Transformer framework for downstream tasks. We also prove that the proposed fair structural topology encoding with adjacency matrix eigenvector selection and multi-hop integration are theoretically effective. Empirical evaluations conducted across five real-world datasets demonstrate FairGT's superiority in fairness metrics over existing graph transformers, graph neural networks, and state-of-the-art fairness-aware graph learning approaches.

1 Introduction

Graph Transformers (GTs) incorporates global attention and facilitates long-range interactions among nodes [Zhu *et al.*, 2023], thus effectively addressing challenges of Graph Neural Networks (GNNs) (e.g., over-smoothing [Guo *et al.*, 2023b] and over-squashing [He *et al.*, 2023]) and handling long-range dependencies [Zhang *et al.*, 2022]. Despite their success, most of the GTs inevitably overlook the bias in the graph data, which leads to discriminatory predictions towards certain sensitive subgroups, such as gender, age, nationality, and race [Caton and Haas, 2023]. That is to say, the issue of fairness in GTs has become a prominent concern when deploying them in real-world scenarios. Effectively employing GTs in practical scenarios, like salary evaluation systems, necessitates generating equitable predictions for individuals with diverse sensitive features.

To quantitatively show the fairness issues exist in GTs, we evaluate one of the most typical fairness-aware GNN methods (i.e., FairGNN [Dai and Wang, 2023]) and three most popular GTs (i.e., GraphTransformer (GraphTrans), Spectral Attention Network (SAN), and Neighborhood Aggregtation Graph Transformer (NAGphormer)) over a real-world dataset (i.e., NBA), with outcomes presented in Table 1. Here, we employ Statistical Parity, i.e., Independence, which is a well adopted notion of fairness assessing the equality of prediction results across diverse demographic groups [Dwork *et al.*, 2012]. The experimental setup aligns with relevant studies [Dwivedi and Bresson, 2020; Kreuzer *et al.*, 2021; Chen *et al.*, 2023], and the higher $\Delta_{\mathbf{SP}}$ value corresponds to the lower fairness. It can be seen that the values of $\Delta_{\mathbf{SP}}$ of GTs are much higher than that of fairness-aware GNN, which indicates the existence of fairness issue in GTs.

Methods	**FairGNN**	**GraphTrans**	**SAN**	**NAGphormer**
$\Delta_{\mathbf{SP}}(\%) \downarrow$	1.32	9.01	29.02	16.74

Table 1: The fairness issue of GTs.

Numerous efforts have been dedicated to improving fairness in graph learning [Xia *et al.*, 2021; Ren *et al.*, 2023]. Specifically, fairness-aware graph learning methods have emerged, which primarily focus on preventing the misuse of sensitive features through additional regularizations or constraints. [Dong *et al.*, 2023a]. FairAC [Guo *et al.*, 2023a] alleviates this problem by introducing a sensitive discriminator to regulate each group of neighboring nodes chosen in the sampling process, effectively generating a bias-free graph. Similarly, FairSample [Cong *et al.*, 2023] enhances model fairness by developing a trainable neighbor sampling policy through reinforcement learning, further optimizing it with the incorporation of a regularization objective.

In the realm of GTs, however, there has been a noticeable lack of exploration concerning fairness considerations. The distinctive architecture of GTs, which enables direct node-to-node interactions instead of relying on neighbor information, poses a unique challenge for fairness-aware methods based on the message-passing mechanism [Wu *et al.*, 2023]. Sim-

*Corresponding author.

ply removing sensitive features may not sufficiently enhance fairness, as the correlation between these features and other factors could still introduce bias [Mehrabi *et al.*, 2021]. As a result, current fairness-aware solutions for graph learning methods cannot be readily applied to tackle the unique fairness issues associated with GTs.

In this work, we present a Fairness-aware Graph Transformer (called FairGT[1]), which enhances the independence of sensitive features during the training process and thus improving fairness. Firstly, FairGT employs an efficient strategy by selecting the eigenvector of the adjacency matrix as the fair structural topology encoding. This departure from the common practice in GTs [Dwivedi and Bresson, 2020; Kreuzer *et al.*, 2021; Chen *et al.*, 2023], which typically involves utilizing eigenvectors corresponding to the s smallest non-trivial eigenvalues of the Laplacian matrix, contributes to its efficiency. Secondly, we introduce a fairness-aware node feature encoding method to keep the independence of sensitive features. Furthermore, we perform a theoretical analysis to evaluate the correlation between the sensitive feature and the two aforementioned encodings introduced by FairGT.

FairGT leverages these two fairness-aware graph information encodings (i.e., structural topology and node feature) as inputs. In structural topology encoding, our alternative approach utilizes eigenvectors corresponding to the t largest magnitude eigenvalue of the adjacency matrix, providing a fairer representation of the structural topology. This encoding maintains original sensitive feature distribution. Furthermore, we construct a sensitive feature complete graph. Based on this graph, FairGT encodes node features from k-hop, while preserving crucial sensitive features for each node. This comprehensive approaches not only enhance the graph information encoding but also ensure the independence of sensitive feature, thereby contributing to a fairness-aware training process.Our main contributions can be summarized as follows:

- To our best knowledge, this work presents the first attempt to address the fairness issue in GTs. We propose FairGT, a novel fairness-aware Graph Transformer, which strategically maintains the independence of sensitive features in the training process.
- We introduce an innovative eigenvector selection mechanism into structural topology encoding, followed by a fairness-aware node feature encoding. These encodings are designed to ensure the independence of sensitive attributes during the training process of Transformer.
- By investigating the correlations between graph information encoding and sensitive features, we provide a theoretical analysis to illustrate the effectiveness of FairGT.
- To validate the effectiveness of FairGT, we conduct comprehensive empirical evaluations on five real-world datasets. The results showcase the superior performance of FairGT when compared to existing state-of-the-art graph Transformers, GNNs, and fairness-aware GNNs.

[1]The source codes and detailed proofs are available at https://github.com/yushuowiki/FairGT.

2 Related Work

2.1 Graph Transformers

With the advancement of Transformer architectures, it has been observed that leveraging the global receptive field of Transformers on graphs yields effectiveness [Ying *et al.*, 2021]. For instance, GraphTrans [Wu *et al.*, 2021] employs Transformer-based self-attention to acquire long-range pairwise relationships, incorporating a novel readout mechanism to derive a global graph embedding. SAN [Kreuzer *et al.*, 2021] utilizes Laplacian positional encodings for nodes and integrated two attention mechanisms: one for virtual fully-connected graphs and another for actual graph edges. However, the above methods are mostly designed based on Laplacian positional encodings (that is to calculate the whole Laplacian matrix), which requires $O(n^3)$ complexity. NAGphormer [Chen *et al.*, 2023] encodes structural topology using eigenvector selection from the Laplacian matrix, and conceptualize each node as a sequence composed of tokens. This module aggregates neighborhood features across multiple hops, generating diverse representations and forming a sequence of token vectors as input for each node. Gapformer [Liu *et al.*, 2023] converts large-scale graph nodes into a reduced set via local or global pooling, enabling attention computation exclusively with these pooling nodes. This mitigates the impact of irrelevant nodes while maintaining long-range information and reducing computational complexity to linearity. Nevertheless, it is imperative to acknowledge that prevailing GTs typically lack explicit considerations for algorithmic fairness. Results obtained from the application of these methods to real-world datasets often expose discernible fairness issues.

2.2 Fairness-aware GNNs

Fairness-aware GNNs have garnered significant attention [Wang *et al.*, 2022; Cheng *et al.*, 2024], which primarily revolve around two methodologies: pre-processing and in-processing. Some existing fairness-aware methods adopt pre-processing techniques. For example, Graphair [Ling *et al.*, 2023] automatically identifies fairness-aware augmentations within input graphs based on the structural topology, aiming to circumvent sensitive features while retaining other pertinent information. However, the unique architecture of GTs, enabling direct node-to-node interactions, is different from traditional structural topology [Yin and Zhong, 2023]. The pre-processing techniques may not be directly applicable to balancing input sensitive features of GTs. In-processing methods focus on making sensitive features independent by altering loss functions or imposing constraints. For instance, NIFTY [Agarwal *et al.*, 2021] optimizes alignment between predictions derived from perturbed and unperturbed sensitive features. Similarly, FairGNN [Dai and Wang, 2023] is an innovative model restricting sensitive features via estimation functions and adversarial debiasing loss. SRGNN [Zhang *et al.*, 2024] presents a fair structural rebalancing algorithm, leveraging gradient constraints to disentangle node representations from sensitive features. However, these constraint-based methods may encounter limitations when applied to

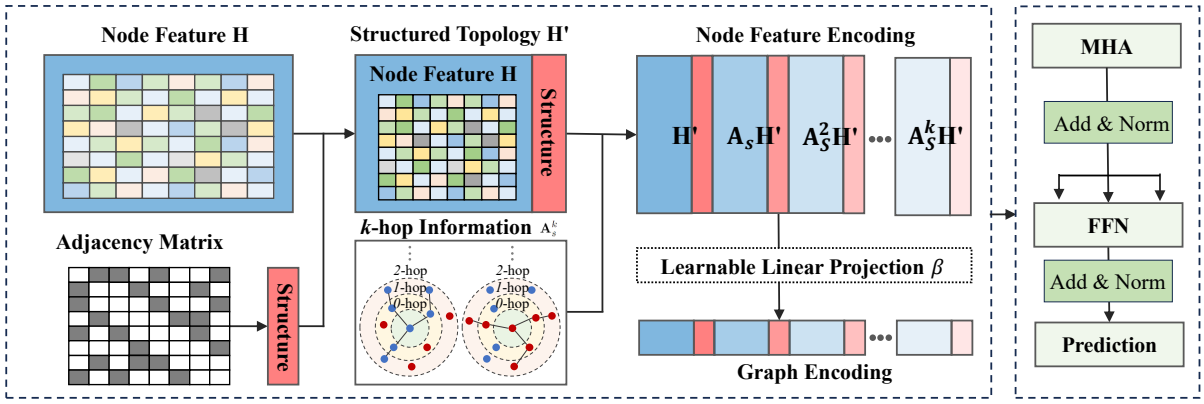

Figure 1: The illustration of FairGT.

GTs owing to the unique nature of graph information encoding.

3 Preliminaries

3.1 Notations

We first introduce the major notations used throughout the paper in Table 2. Unless otherwise specified, we denote set with copperplate uppercase letters (e.g., $\mathcal{A}$), matrices with bold uppercase letters (e.g., $\mathbf{A}$), and vectors with bold lower-case letters (e.g., $\mathbf{x}$). We denote a graph as $\mathcal{G} = \{\mathcal{V}, \mathbf{A}, \mathbf{H}\}$, where $\mathcal{V}$ is the set of n nodes in the graph, $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the adjacency matrix, and $\mathbf{H} \in \mathbb{R}^{n \times d}$ is the node feature matrix.

We use rules similar to NumbPy in Python for matrix and vector indexing. $\mathbf{A}[i, j]$ represents the entry of matrix $\mathbf{A}$ at the i-th row and the j-th column. $\mathbf{A}[i, :]$ and $\mathbf{A}[:, j]$ represent the i-th row and the j-th column of matrix $\mathbf{A}$, respectively. One column of $\mathbf{H}$ is the sensitive feature of nodes, we denote it as $\mathbf{H}[:, s]$. For the binary sensitive feature, $s \in \{0, 1\}$.

Notations	Definitions and Descriptions
$\mathcal{G}$	graph set
$\mathcal{V}$	node set
$\mathbf{A}$	adjacency matrix
$\mathbf{H}$	node feature matrix
n	number of nodes
d	dimension of feature vector
k	number of hops
t	number of eigenvectors
l	number of layers
s	sensitive feature

Table 2: Notations.

3.2 Fairness Evaluation Metrics

We present one definition of fairness for the binary label $y \in \{0, 1\}$ and the sensitive feature $s \in \{0, 1\}$, $\hat{y} \in \{0, 1\}$ denotes the class label of the prediction.

Statistical Parity [Dwork *et al.*, 2012] (i.e., Demographic Parity and Independence). Statistical parity requires the predictions to be independent of the sensitive features s. It could be formally written as:

$$\mathbb{P}(\hat{y}|s = 0) = \mathbb{P}(\hat{y}|s = 1). \tag{1}$$

When both the predicted labels and sensitive features are binary, to quantify the extent of statistical parity, the Δ_{SP} is defined as follows:

$$\Delta_{\text{SP}} = |\mathbb{P}(\hat{y} = 1|s = 0) - \mathbb{P}(\hat{y} = 1|s = 1)|. \tag{2}$$

The Δ_{SP} measures the acceptance rate difference between the two sensitive subgroups.

3.3 Transformer

The Transformer architecture consists of a composition of Transformer layers. Each Transformer layer has two parts: a self-attention module and a position-wise feed-forward network (FFN). Let $\mathbf{X} = [h_1^\top, ..., h_i^\top]^\top \in \mathbb{R}^{i \times d_m}$ denotes the input of self-attention module where d_m is the hidden dimension and $h_j \in \mathbb{R}^{1 times e s d_m}$ is the hidden representation at position j. The input $\mathbf{X}$ is projected by three matrices $W_Q \in \mathbb{R}^{d_m \times d_K}$, $W_K \in \mathbb{R}^{d_m \times d_K}$ and $W_V \in \mathbb{R}^{d_m \times d_V}$ to the corresponding representations Q, K, V. The self-attention is then calculated as:

$$Q = \mathbf{X}W_Q, K = \mathbf{X}W_K, V = \mathbf{X}W_V, \tag{3}$$

$$\text{Attn}(\mathbf{X}) = \text{softmax}(\frac{QK^\top}{\sqrt{d_K}})V. \tag{4}$$

For simplicity of illustration, we consider the self-attention and assume $d_K = d_V = d$. The extension to the multi-head attention is straightforward, and we omit bias terms for simplicity.

4 The Design of FairGT

In this section, we first present our theoretical findings that underpin FairGT, followed by the design of structural topology encoding and node feature encoding. The architecture of FairGT (see Figure 1) is then introduced, as well as its complexity analysis.

4.1 Theoretical Findings Underpinning FairGT

For a fairness-aware structural topology encoding, we analyze the similarity between the distribution of original sensitive features and the distribution of k-hop neighbor sensitive features.

Lemma 1 *The similarity between the distribution of original sensitive features and the distribution of k-hop neighbor*

sensitive features exhibits a pronounced correlation with the eigenvector corresponding to the largest magnitude eigenvalue of the adjacency matrix. Concurrently, the correlation with other eigenvectors diminishes exponentially.

Proof. Assume $\mathbf{A} \in \mathbb{R}^{n\times n}$ is an adjacency matrix with real-valued entries. The k-hop neighbor feature matrix is $\mathbf{A}^k\mathbf{H}$, and the k-hop neighbor sensitive features are $\mathbf{A}^k\mathbf{H}[:, s]$. $|\lambda_1| > |\lambda_2| \geq ..... \geq |\lambda_n|$ are n real eigenvalues, and $\mathbf{p}_i$ ($i \in \{1, 2, ..., n\}$) are corresponding eigenvectors. We assume $\alpha_i = \mathbf{H}[:, s]^\top \mathbf{p}_i$. We use cosine similarity to measure the similarity. Thus, the following equation establishes:

$$cos(< \mathbf{p}_i, \mathbf{H}[:, s] >) = \frac{\alpha_i}{\sqrt{\sum_{j=1}^{n} \alpha_j^2}}.$$

Then,

$$\begin{aligned} &cos(< \mathbf{A}^k\mathbf{H}[:, s], \mathbf{H}[:, s] >) \\ =&\frac{\alpha_1^2 + \sum_{i=2}^{n} \alpha_i^2 (\frac{\lambda_i}{\lambda_1})^l}{\sqrt{\alpha_1^2 + \sum_{i=2}^{n} \alpha_i^2 (\frac{\lambda_i}{\lambda_1})^{2l}} \sqrt{\sum_{i=1}^{n} \alpha_i^2}}. \end{aligned}$$

Thus, the correlation between $\mathbf{p}_i$ and $cos(< \mathbf{A}^k\mathbf{H}[:, s], \mathbf{H}[:, s] >)$ is proportional to $(\frac{\lambda_i}{\lambda_1})^l$. Because of $|\lambda_1| > |\lambda_i|$, the correlation decays exponentially. Specifically, when $k \to \infty$, we have:

$$\lim_{k\to\infty} cos(< \mathbf{A}^k\mathbf{H}[:, s], \mathbf{H}[:, s] >) = cos(< \mathbf{p}_1, \mathbf{H}[:, s] >).$$

□

To ensure the node feature encoding to be independent with sensitive features, we divide the nodes of the origin graph $\mathcal{G}$ into two subgraphs $\mathcal{G}_0, \mathcal{G}_1$. Specifically, $\mathcal{G}_0 = \{v \in \mathcal{G} | v_{sen} = 0\}$ and $\mathcal{G}_1 = \{v \in \mathcal{G} | v_{sen} = 1\}$. Both $\mathcal{G}_0$ and $\mathcal{G}_1$ are complete subgraphs. We propose a sensitive feature complete graph, $\mathcal{G}_s = \mathcal{G}_0 \cup \mathcal{G}_1$. Given the adjacency matrix $\mathbf{A}_s$ of $\mathcal{G}_s$ and $\mathbf{H}$, multiplying $\mathbf{A}_s$ with $\mathbf{H}$ aggregates k-hop sensitive information. Applying this multiplication consecutively allows us to propagate information at larger distances. For example, we can access 2-hop sensitive information by $\mathbf{A}_s(\mathbf{A}_s\mathbf{H})$. Thereafter, the k-hop sensitive matrix can be described as:

$$\mathbf{H}^{(k)} = \mathbf{A}_s^k\mathbf{H}. \tag{5}$$

Lemma 2 *The sensitive feature distribution of $\mathbf{H}^{(k)}$ is the same as $\mathbf{H}$:*

$$\mathbf{H}^{(k)}[i, s] = q^k\mathbf{H}[i, s], \tag{6}$$

where $i \in \{1, 2, ..., n\}$, q denotes the number of nodes whose sensitive feature is 1.

Proof. We prove this lemma by mathematical induction. We firstly prove the based case (k=1). Then, we give the inductive hypothesis (k=r), and confirm the inductive step ($k = r + 1$). Combining the foundational step, the inductive hypothesis, and the inductive step, we show that this mathematical statement holds for all positive integer k.

□

The node feature encodings of FairGT effectively aggregate information from k-hop nodes while upholding independence with sensitive features.

4.2 Structural Topology Encoding

The structural information of nodes is a crucial feature for graph mining tasks. Spectral graph theory illustrates that the algebraic connectivity and spectral radius, representing the lowest and largest non-zero eigenvalues respectively, are intricately linked to the geometric properties of the graph [Kreuzer *et al.*, 2021].

Furthermore, inspired by **Lemma 1**, we adopt the eigenvectors selection of adjacency matrix of the graph for capturing the structural information of nodes. Specifically, we select the eigenvectors corresponding to the t largest magnitude eigenvalues to construct the structure matrix $\mathbf{B} \in \mathbb{R}^{n\times t}$. Then we combine the original feature matrix $\mathbf{H}$ with the structure matrix $\mathbf{B}$ to preserve both node features and structural information:

$$\mathbf{H}' = \mathbf{H}||\mathbf{B}, \tag{7}$$

where $||$ indicates the concatenation operator and $\mathbf{H}' \in \mathbb{R}^{n\times(d+t)}$ denotes the fused feature matrix. In this structural topology, the similarity between the distribution of sensitive features and the sensitive features of k-hop neighbors is high, contributing to a fairer graph information encoding.

4.3 Node Feature Encoding

In addition to the structural topology of graph, node features also contain valuable information in a graph. To address fairness issues in GTs, we introduce a fairness-aware node feature encoding. This encoding method considers the k-hop information independent with sensitive features.

Specifically, for node v, let $\mathcal{V}_s^{(k)} = \{v^{(k)} \in \mathcal{V} | v_s^{(k)} = v_s^{(k-1)}, v^{(k)} \neq v^{(k-1)}\}$ be its k-hop information set. We define $\mathcal{V}_s^{(0)} = v$, i.e., the 0-hop is the node itself. In node feature encoding, we transform the k-hop information set $\mathcal{V}_s^{(k)}$ into a feature embedding $\mathbf{H'}_v^{(k)}$ with an operator Φ. The operator Φ serves as an aggregation operator that aggregates k-hop information from same sensitive feature neighbors. In this way, the k-hop representation of node v can be expressed as:

$$\mathbf{H'}_v^{(k)} = \Phi(v, \mathcal{V}_s^{(1)}, ..., \mathcal{V}_s^{(k)}). \tag{8}$$

By using Equation (8), we can calculate the information for variable hops of a certain node and further construct the corresponding sequence, i.e., $\mathbf{S}_v = (\mathbf{H'}_v^{(0)}, \mathbf{H'}_v^{(1)}, ..., \mathbf{H'}_v^{(k)})$. Assume $\mathbf{H'}_v^{(k)}$ is a d-dimensional vector, the sequence of all nodes in graph $\mathcal{G}$ will constitute a tensor $\mathcal{H} \in \mathbb{R}^{n\times(k+1)\times d}$. To better illustrate the implementation of node feature encoding, we decompose $\mathbf{H}'$ to a sequence $\mathbf{S} = (\mathbf{H'}^{(0)}, \mathbf{H'}^{(1)}, ..., \mathbf{H'}^{(k)})$, where $\mathbf{H'}^{(k)} \in \mathbb{R}^{n\times d}$ can be seen as the k-hop feature matrix. Here we define $\mathbf{H'}^{(0)}$ as the original feature matrix. In the experiments, we create a sensitive feature complete graph $\mathcal{G}_s$ (details are illustrated in Section 4.1). Given the adjacency matrix $\mathbf{A}_s$ of $\mathcal{G}_s$. The k-hop feature matrix is defined as following equation:

$$\mathbf{H'}^{(k)} = \mathbf{A}_s^k\mathbf{H}'. \tag{9}$$

According to **Lemma 2**, the encoding process of node features ensures the independence of sensitive features. Representing the k-hop information following this encoding proves

to be beneficial for capturing hop-wise correlations, which is a crucial aspect in fairness-aware graph transformers.

4.4 FairGT for Node Classification

Given an graph, we first concatenate a matrix constructed by eigendecomposition to the adjacency matrix, and encode structure topology of graph following Equation (7). Next, we assemble an aggregated neighborhood sequence as $\mathbf{S}_v = (\mathbf{H'}_v^{(0)}, \mathbf{H'}_v^{(1)}, ..., \mathbf{H'}_v^{(k)})$ by applying node feature encoding. Then we map $\mathbf{S}_v$ to the hidden dimension d_h of the Transformer with a learnable linear projection:

$$\mathbf{T}_v^{(0)} = [\mathbf{H'}_v^{(0)}\beta, \mathbf{H'}_v^{(1)}\beta, ..., \mathbf{H'}_v^{(k)}\beta], \tag{10}$$

where $\beta \in \mathbb{R}^{d \times d_h}$ and $\mathbf{T}_v^{(0)} \in \mathbb{R}^{(k+1) \times d_h}$.

Then, following the projection of the sequence representing structural topology, we proceed to input this transformed sequence into the Transformer architecture [Vaswani *et al.*, 2017]. The building blocks of the Transformer contain multi-head self-attention (MHA) and position-wise feed-forward network (FFN), wherein LayerNorm(LN) is applied before each block. The FFN consists of two linear layers with a GELU non-linearity:

$$\begin{aligned} \mathbf{T'}_v^{(l)} &= \mathbf{MHA}(\mathbf{LN}(\mathbf{T'}_v^{(l-1)})) + \mathbf{T'}_v^{(l-1)}, \\ \mathbf{T}_v^{(l)} &= \mathbf{FFN}(\mathbf{LN}(\mathbf{T'}_v^{(l)})) + \mathbf{T'}_v^{(l)}, \end{aligned} \tag{11}$$

where $l = 1, 2, ..., L$ implies the l-th layer of the Transformer.

In the end, we obtain the prediction by applying weighted summation to the output of the encoder using attention mechanisms. Through several Transformer layers, the corresponding output $\mathbf{T}_v^{(l)}$ contains the embeddings for all neighborhoods of node v. It requires a weighted summation to aggregate the information of k-hop node features into one. Conventional summation methods often overlook the significance of diverse neighborhoods. In addressing this limitation, we employ attention-based summation to learn the importance of various neighborhoods by computing the attention coefficients.

4.5 Complexity of FairGT

Existing GTs treat the nodes as independent tokens and construct a single sequence composed of all the node tokens to train the Transformer model, suffering from a quadratic complexity on the number of nodes for the self-attention calculation. Because FairGT encodes node features as one vector, the time complexity of FairGT is $O(n(k+1)^2 d)$ and the space complexity of FairGT is $O(n(k+1)^2 + n(k+1)d + d^2 l)$.

Furthermore, eigendecomposition causes cubic complexity in the number of nodes, which captures the node structural information, resulting in a computational cost of $O(n^3)$. In the eigenvector selection process of FairGT, we donnot need to rank the eigenvector after eigendecomposition. Within the Arnoldi Package algorithm [Lehoucq *et al.*, 1998], the time complexity of eigendecomposition in FairGT is $O(nt^2)$, and the space of complexity is $O(n)$.

5 Experiments

5.1 Datasets

In this study, node classification serves as the downstream task, employing five distinct real-world datasets: **NBA**, **Bail**, **German**, **Credit**, and **Income**. The statistics of five datasets are shown in Table 3.

- **NBA** [Dai and Wang, 2021]: The NBA dataset, sourced from Kaggle, encompasses player statistics from the 2016-2017 season, including additional player information and Twitter relationships. It categorizes nationality as U.S. or overseas, focusing on predicting whether a player's salary exceeds the median.
- **Bail** [Jordan and Freiburger, 2015]: This dataset represents defendants who got released on bail at the U.S. state courts during 1990-2009, connected by edges based on shared past criminal records and demographics. The goal is predicting a defendant's likelihood of committing either a violent or nonviolent crime post-release, with 'race' as the sensitive feature.
- **German** [Asuncion and Newman, 2007]: German is extracted from the Adult Data Set . The dataset is a credit graph which has 1,000 nodes representing clients in a German bank that are connected based on the similarity of their credit accounts. The objective is to categorize clients' credit risk as either high or low, with "gender" designated as the sensitive feature.
- **Credit** [I-cheng and Che-hui, 2009]: The Credit dataset uses personal next month. Comprising individuals connected based on similarities in spending and payment patterns. Age serves as the sensitive feature, while the label feature denotes defaulting on credit card payments.
- **Income** [Asuncion and Newman, 2007]: Income is extracted from the Adult Data Set. Each node represents an individual, with connections established based on criteria similar to [Agarwal *et al.*, 2021]. The sensitive feature in this dataset is race, and the task involves classifying whether an individual's salary exceeds $50,000$ annually.

Dataset	# Nodes	# Edges	Sensitive feature	Label
NBA	403	16,570	Nationality	Salary
German	1,000	22,242	Gender	Customer
Bail	18,876	321,308	Race	Recidivism
Credit	30,000	137,377	Age	Default
Income	14,821	100,483	Race	Income

Table 3: The statistics of the five real-world datasets.

Considering some datasets includes more than two classes of ground truth labels, we save the class of labels 0 and 1 and change the class of labels more than 1 to 1. Then, we randomly select 25% nodes as the validation set and 25% as the test set, ensuring a balanced ratio of ground truth labels. On **NBA**, we randomly select either 50% nodes or 50 nodes in each class of ground truth labels, depending on

Methods	NBA		Bail		German		Credit		Income	
	ACC ↑	Δ_{SP} ↓	ACC ↑	Δ_{SP} ↓	ACC ↑	Δ_{SP} ↓	ACC ↑	Δ_{SP} ↓	ACC ↑	Δ_{SP} ↓
GCN	71.70	9.18	84.56	7.35	73.44	35.17	73.87	12.86	73.87	25.93
GCNII	72.68	14.17	92.39	5.67	71.60	6.81	73.95	16.85	76.20	16.20
GAT	72.45	11.54	93.24	5.44	72.80	12.52	68.29	9.74	69.14	12.46
Specformer	73.42	11.59	95.46	6.32	72.40	6.90	74.06	8.37	80.06	7.26
FairGNN	69.72	1.32	82.94	6.90	69.68	3.49	68.29	9.74	69.14	12.46
NIFTY	70.16	3.26	83.43	4.75	69.92	5.73	66.81	13.59	70.87	24.43
BIND	72.14	4.37	89.72	7.77	71.60	3.46	68.64	11.65	71.76	14.75
Graphair	70.99	2.54	84.76	4.98	70.40	6.39	67.68	8.99	71.50	10.68
GraphTrans	73.81	9.01	93.56	7.03	74.40	8.27	71.35	8.59	79.57	8.89
SAN	73.41	29.02	74.91	6.97	73.20	9.77	70.01	8.39	79.44	7.41
NAGphormer	72.15	16.74	93.28	7.03	75.20	8.27	77.81	8.29	80.17	7.49
FairGT	**74.68**	**0.38**	**95.68**	**0.58**	**76.00**	**0.26**	**77.85**	**1.89**	**81.30**	**2.66**

Table 4: Comparison of performance (accuracy) and fairness (Δ_{SP}) in percentage (%). ↑ denotes the larger, the better; ↓ denotes the opposite. The best results are bold-faced.

which is a smaller number. On **Bail**, we change the number of nodes from 50 to 100. On **German**, **Credit**, and **Income**, we change 50 to 500. Such a splitting strategy is also followed by fairness-aware baselines [Agarwal *et al.*, 2021; Dai and Wang, 2023; Dong *et al.*, 2023b].

5.2 Baselines

To ensure diversity, we use three classes of baselines: GNNs, GTs, and fairness-aware GNNs.

- **GNNs**: We use two GNNs: GCN [Kipf and Welling, 2017] and GCNII [Chen *et al.*, 2020]. We also used two GNNs based on attention mechanism: GAT [Velickovic *et al.*, 2018] and Specformer [Bo *et al.*, 2023].
- **Fairness-aware GNNs**: We use four fairness-aware GNNS: FairGNN [Dai and Wang, 2023], NIFTY [Agarwal *et al.*, 2021], BIND [Dong *et al.*, 2023b], and Graphair [Ling *et al.*, 2023].
- **GTs**: We use three representative GTs: GraphTrans [Dwivedi and Bresson, 2020], SAN [Kreuzer *et al.*, 2021], and NAGphormer [Chen *et al.*, 2023].

5.3 Comparison Results

Table 4 provides a comprehensive overview of the fairness evaluation metrics, concerning our proposed FairGT method and various baseline models across five real-world datasets. We report the overall Accuracy and Δ_{SP} (average results of five-fold cross-validation) respectively. This table shows that FairGT consistently demonstrates outstanding fairness, proving the effectiveness of our method in the fairness-aware node classification. FairGT not only enhances fairness compared to GTs but also outperforms existing fairness-aware GNN methods in terms of fairness results. This underscores its efficacy in addressing fairness-related concerns within GTs. Furthermore, FairGT demonstrates improved performance in node classification across five real-world datasets.

5.4 Training Cost Comparison

FairGT, utilizing a subset of adjacency matrix eigenvectors and employing vectors as tokens, may exhibit efficiency compared to GTs. To validate the efficiency of FairGT, we compare its training time with GT baselines. For a fair comparison, we standardize key parameters across all methods, setting the number of hidden dimensions to 128, the number of layers to 1, the number of heads to 1, and the number of epochs to 500.

Dataset	FairGT	GraphTrans	SAN	NAGphormer
NBA	17.87	24.06	36.98	20.21
Bail	116.95	168.04	203.01	137.06
German	19.82	33.91	39.56	21.07
Credit	217.35	241.94	252.22	186.42
Income	124.99	118.29	140.55	119.90

Table 5: Runtime (s) of GTs over five real-world datasets.

The results are summarized in Table 5, which demonstrate that our method attains superior performance and fairness in node classification without significantly escalating computational complexity.

5.5 Ablation Study

FairGT as an fairness-aware GT achieves its objectives through two key components: eigenvector selection and k-hop sensitive information. To comprehensively assess the contributions of these two elements, we conducted an ablation study. Specifically, we aim to examine the extent to which eigenvector selection or the incorporation of k-hop sensitive information contributes to improving prediction fairness and accuracy. In this analysis, we systematically remove each component independently to assess their individual impact.

We conduct a specific ablation analysis focused on the contribution of eigenvalue selection. Because some GTs use Laplacian matrix to encode structural topology, we also consider eigenvalue selection of Laplacian matrix, represented as **Lap ST**. The FairGT without eigenvalue selection is denoted as **w/o ST**. Notably, when comparing **Lap ST** and **w/o ST**, FairGT consistently outperforms them in terms of accuracy and statistical parity. The results shown in Table 6 reinforce

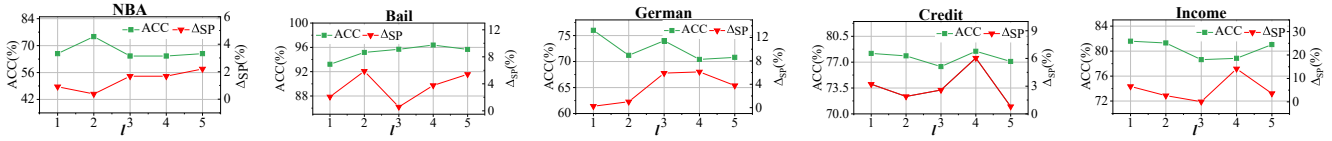

Figure 2: The accuracy and Δ_{SP} of FairGT w.r.t. different parameter l values.

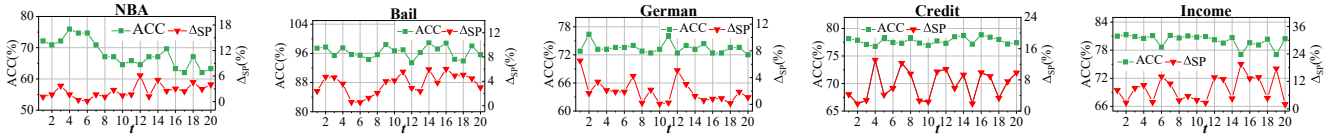

Figure 3: The accuracy and Δ_{SP} of FairGT w.r.t. different parameter t values.

the necessity of eigenvector selection within FairGT. We denote the configuration as **w/o NF** (FairGT without node feature encoding). We also consider k-hop neighborhood information based on adjacency matrix, represented as **Adj NF**. The results shown in Table 6 provide compelling evidence of the contributions of the component. When compared to the ablated versions, FairGT is consistently higher in terms of accuracy and lower in terms of Δ_{SP}. This observation underscores the pivotal role that k-hop sensitive information plays in concurrently improving the fairness of GTs' predictions and accuracy.

Dataset	Metric	FairGT	w/o ST	Lap ST	w/o NF	Adj NF
NBA	ACC ↑	**74.68**	58.23	64.55	63.29	69.62
	Δ_{SP} ↓	**0.38**	6.21	1.36	2.42	10.76
Bail	ACC ↑	**95.68**	93.96	93.81	90.76	92.73
	Δ_{SP} ↓	**0.58**	6.42	6.46	5.87	5.95
German	ACC ↑	**76.00**	75.19	75.20	70.80	74.40
	Δ_{SP} ↓	**0.26**	2.29	6.12	1.30	2.29
Credit	ACC ↑	**77.85**	75.20	75.20	75.20	76.80
	Δ_{SP} ↓	**1.89**	2.29	6.12	7.40	6.11
Income	ACC ↑	**81.30**	78.90	80.65	80.00	81.00
	Δ_{SP} ↓	**2.66**	5.52	4.28	4.37	5.42

Table 6: Ablation study on different components of FairGT.

In summary, the ablation study results collectively emphasize the integral role played by eigenvector selection and k-hop sensitive information for improving fairness and performance of GTs.

5.6 Parameter Analysis

To comprehensively evaluate FairGT's performance, we investigate the impact of two crucial parameters: the number of eigenvectors t and the number of Transformer layers l, conducting experiments across **NBA**, **Bail**, **German**, **Credit**, and **Income**.

With $l = 2$, we select the numbers of feature vector in $\{1, 2, 3, 4, 5, ..., 20\}$, respectively. Remarkably, t differs across datasets to achieve peak performance due to the varied structural topologies of different networks. The results are shown in Figure 2. This performance fluctuation with t increment suggests diverse effects of structural topology across different network types, influencing model performance diversely. After comparing both fairness and performance metrics, we choose the best result as our final selection for the parameter t. We select $t = 5$ for **NBA**, $t = 5$ for **Bail**, $t = 11$ **German**, $t = 2$ for **Credit**, and $t = 2$ for **Income**.

Besides, we fix the best value of t and change l from 1 to 5 on each dataset. The results are shown in Figure 3. We select the best result as our final selection for the parameter l, after evaluating both fairness and performance metrics. We set $l = 2$ for **NBA**, $l = 3$ for **Bail**, $l = 3$ for **German**, $l = 2$ for **Credit**, and $l = 1$ for **Income**.

The results clearly indicate pronounced fluctuations in the outcomes for both parameters. The selection of values for l and t is crucial for FairGT. Prudent choices of l and t are essential to obtain improved results in the experiment. It is noteworthy that our choice might represent only a locally optimal solution, and hence conducting a sufficient number of experiments would be necessary to explore the entire parameter space. This aspect also suggests a potential avenue for refining the parameters of FairGT.

6 Conclusion

In this work, we have established a vital connection between graph information encoding and sensitive features to address the fairness issues in GTs. We have proposed FairGT, an innovative approach dedicated to enhance the fairness of GTs. FairGT is built on two fairness-aware graph information encoding: structural topology encoding and node feature encoding. The designed components in FairGT collaborate to improve the fairness and performance of GTs. We also present theoretical analysis to prove the efficacy of the proposed approach. In diverse real-world datasets, FairGT outperforms state-of-the-art baselines. Despite achieving noteworthy results, our approach has limitations, as evidenced by pronounced fluctuations in outcomes when altering both parameters. This underscores the critical importance of selecting optimal values for l and t. In future work, we will further refine the efficiency of the FairGT approach and expand its applicability to situations with limited sensitive features. We aspire to contribute to the ongoing progression of fairness-aware GTs and foster their broader adoption in real-world applications.

References

[Agarwal *et al.*, 2021] Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. Towards a unified framework for fair and stable graph representation learning. In *UAI*, pages 2114–2124, 2021.

[Asuncion and Newman, 2007] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[Bo *et al.*, 2023] Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. In *ICLR*, 2023.

[Caton and Haas, 2023] Simon Caton and Christian Haas. Fairness in machine learning: A survey. *ACM Computing Surveys*, 2023.

[Chen *et al.*, 2020] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *ICML*, pages 1725–1735, 2020.

[Chen *et al.*, 2023] Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. NAGphormer: A tokenized graph transformer for node classifcation in large graphs. In *ICLR*, 2023.

[Cheng *et al.*, 2024] Debo Cheng, Jiuyong Li, Lin Liu, Jixue Liu, and Thuc Duy Le. Data-driven causal effect estimation based on graphical causal modelling: A survey. *ACM Computing Surveys*, 56(5):1–37, 2024.

[Cong *et al.*, 2023] Zicun Cong, Baoxu Shi, Shan Li, Jaewon Yang, Qi He, and Pei Jian. FairSample: Training fair and accurate graph convolutional neural networks efficiently. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–14, 2023.

[Dai and Wang, 2021] Enyan Dai and Suhang Wang. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In *WSDM*, pages 680–688, 2021.

[Dai and Wang, 2023] Enyan Dai and Suhang Wang. Learning fair graph neural networks with limited and private sensitive attribute information. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):7103–7117, 2023.

[Dong *et al.*, 2023a] Yushun Dong, Jing Ma, Song Wang, Chen Chen, and Jundong Li. Fairness in graph mining: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(01):10583–10602, 2023.

[Dong *et al.*, 2023b] Yushun Dong, Song Wang, Jing Ma, Ninghao Liu, and Jundong Li. Interpreting unfairness in graph neural networks via training node attribution. In *AAAI*, pages 7441–7449, 2023.

[Dwivedi and Bresson, 2020] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

[Dwork *et al.*, 2012] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. In *ITCS*, pages 214–226, 2012.

[Guo *et al.*, 2023a] Dongliang Guo, Zhixuan Chu, and Sheng Li. Fair attribute completion on graph with missing attributes. In *ICLR*, 2023.

[Guo *et al.*, 2023b] Xiaojun Guo, Yifei Wang, Tianqi Du, and Yisen Wang. Contranorm: A contrastive learning perspective on oversmoothing and beyond. In *ICLR*, 2023.

[He *et al.*, 2023] Xiaoxin He, Bryan Hooi, Thomas Laurent, Adam Perold, Yann LeCun, and Xavier Bresson. A generalization of vit/mlp-mixer to graphs. In *ICML*, pages 12724–12745, 2023.

[I-cheng and Che-hui, 2009] Yeh I-cheng and Lien Che-hui. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480, 2009.

[Jordan and Freiburger, 2015] Kareem L Jordan and Tina L Freiburger. The effect of race/ethnicity on sentencing: Examining sentence type, jail length, and prison length. *Journal of Ethnicity in Criminal Justice*, 13(3):179–196, 2015.

[Kipf and Welling, 2017] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Kreuzer *et al.*, 2021] Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Letourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *NeurIPS*, pages 21618–21629, 2021.

[Lehoucq *et al.*, 1998] Richard B. Lehoucq, Danny C. Sorensen, and Chao Yang. *ARPACK users' guide - solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. Software, environments, tools. SIAM, 1998.

[Ling *et al.*, 2023] Hongyi Ling, Zhimeng Jiang, Youzhi Luo, Shuiwang Ji, and Na Zou. Learning fair graph representations via automated data augmentations. In *ICLR*, 2023.

[Liu *et al.*, 2023] Chuang Liu, Yibing Zhan, Xueqi Ma, Liang Ding, Dapeng Tao, Jia Wu, and Wenbin Hu. Gapformer: Graph transformer with graph pooling for node classification. In *IJCAI*, 2023.

[Mehrabi *et al.*, 2021] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54(6):1–35, 2021.

[Ren *et al.*, 2023] Jing Ren, Feng Xia, Ivan Lee, Azadeh Noori Hoshyar, and Charu Aggarwal. Graph learning for anomaly analytics: Algorithms, applications, and challenges. *ACM Transactions on Intelligent Systems and Technology*, 14(2):1–29, 2023.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and

Yoshua Bengio. Graph attention networks. In *NeurIPS*, page 4, 2018.

[Wang *et al.*, 2022] Lei Wang, Shuo Yu, Falih Gozi Febrinanto, Fayez Alqahtani, and Tarek E. EI-Tobely. Fairness-aware predictive graph learning in social networks. *Mathematics*, 10(15):2696, 2022.

[Wu *et al.*, 2021] Zhanhao Wu, Paras Jain, Mattew A. Wright, Azalia Mirhoseini, Joseph E. Gonzalez, and Ion Stoica. Representing long-range context for graph neural networks with global attention. In *NeurIPS*, pages 13266–13279, 2021.

[Wu *et al.*, 2023] Yi Wu, Yangyang Xu, Wenhao Zhu, Guojie Song, Zhouchen Lin, Liang Wang, and Shaoguo Liu. KDLGT: A linear graph transformer framework via kernel decomposition approach. In *IJCAI*, 2023.

[Xia *et al.*, 2021] Feng Xia, Ke Sun, Shuo Yu, Abdul Aziz, Liangtian Wan, Shirui Pan, and Huan Liu. Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*, 2(2):109–127, 2021.

[Yin and Zhong, 2023] Shuo Yin and Guoqiang Zhong. LGI-GT:graph Transformers with local and global operators interleaving. In *IJCAI*, 2023.

[Ying *et al.*, 2021] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *NeurIPS*, pages 28877–28888, 2021.

[Zhang *et al.*, 2022] Zaixi Zhang, Qi Liu, Qingyong Hu, and Chee-Kong Lee. Hierarchical graph transformer with adaptive node sampling. In *NeurIPS*, pages 21171–21183, 2022.

[Zhang *et al.*, 2024] Guixian Zhang, Debo Cheng, Guan Yuan, and Shichao Zhang. Learning fair representations via rebalancing graph structure. *Information Processing & Management*, 61(1), 2024.

[Zhu *et al.*, 2023] Wenhao Zhu, Tianyu Wen, Guojie Song, Xiaojun Ma, and Liang Wang. Hierarchical transformer for scalable graph learning. In *IJCAI*, 2023.