Towards Fair Graph Generation Without Sensitive Attribute

Zichong Wang[1] **Zhipeng Yin**[1] **Wenbin Zhang**[1]*
[1] Florida International University, Miami, FL, United States
{zwang114, zyin007, wenbin.zhang}@fiu.edu

Abstract

Graph generation aims to produce new graphs from a distribution similar to that of observed data and has gained increasing attention as generated graphs are frequently used in high-stakes decision-making where fairness is critical. However, most existing fair graph generation methods assume full access to sensitive attributes, an assumption often violated in practice due to privacy concerns, regulatory constraints, and missing data. To this end, we propose to solve the problem from a new perspective, where sensitive proxy inference is reformulated as a component of the graph generation pipeline, enabling the inferred proxy to guide the generation process. We further model and mitigate the impact of proxy inference errors, and provide theoretical guarantees that quantify how such errors propagate to fairness outcomes in the generated graphs, offering practical guidance for interpreting fairness when sensitive attribute is missing. Experiments on benchmark datasets demonstrate that our method consistently improves fairness while maintaining competitive generation quality.

1 Introduction

Graph learning models have shown their strong ability in modeling graph structured data, and are widely used in a variety of applications, *e.g.*, hiring processes [1], healthcare analytics [2], and financial decision-making [3]. Among the important tasks in graph learning, graph generation has gained increasing attention, as it has emerged as a powerful tool for synthesizing realistic graphs that support a range of downstream tasks such as data augmentation [4], anomaly detection [5], and privacy-preserving data sharing [6]. Given that these synthetic graphs are increasingly used in high-stakes decision-making tasks that have substantial consequences for individuals and society, ensuring fairness in graph generation has become a crucial requirement driven by legal, ethical, and social considerations [7, 8, 9]. To this end, a growing body of work has been proposed to mitigate bias in graph generation models, aiming to ensure that the outputs of graph generation models are independent of sensitive attributes such as gender or race [10].

Despite their effectiveness, existing fair graph generation models mainly depend on full access to sensitive attribute to enforce their fairness constraints [9]. However, this assumption rarely holds in practice, as demographic attributes are often unavailable due to privacy restrictions [11], legal regulations [12], or the unwillingness of individuals to share sensitive information out of fear of discrimination [13]. For example, a financial institution may apply fair graph generation models to produce synthetic transaction graphs to share with external partners or for downstream credit scoring [14]. However, customers from minority racial or ethnic groups, may choose not to disclose their sensitive attribute, fearing that such disclosure could result in unfavorable lending decisions [15]. This scenario, where the sensitive attribute is not fully accessible, creates a critical gap between

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

current fair graph generation methods and real-world deployment, since these methods cannot operate without the complete sensitive attribute [16].

Despite the importance of fair graph generation without sensitive attributes and the growing attention this problem has received from the research community, achieving it in practice still faces several fundamental challenges: **i) Difficulty in establishing a shared representation pathway for proxy inference and graph generation.** Unlike classification settings where the sensitive proxy can be inferred from a separate encoder and then used as an external supervision signal, fair graph generation requires the proxy to be inferred from the same internal representations that the generator uses to generate graph. This shared dependency creates a fundamental challenge, since the representations need to simultaneously support accurate proxy inference and high-quality graph generation, and any compromise on one side directly degrades the other. The key challenge is therefore how to design a representation pathway that supports both tasks without sacrificing either. **ii) Difficulty in reducing the impact of inference error propagation.** In fair graph generation, inferred proxies are not only used to evaluate fairness, but also become part of the training signal that guides graph generation. As the generator iteratively refines the graph step by step, errors in the inferred proxies are carried into every step of the generation process, shaping how each node attribute and edge is sampled. Even small inference errors can therefore compound across steps and lead to systematic deviations in the generated graph. Hence, how to mitigate the impact of inference errors becomes a key challenge. **iii) Difficulty in providing theoretical fairness guarantees without sensitive attributes.** Most existing fair graph generation methods rely on empirical fairness constraints, *i.e.*, reducing statistical dependence between generated graph components and sensitive attribute. Although such constraints can improve fairness on observed or inferred groups, they cannot provide end users with any formal guarantee on the fairness of the generated graphs. Providing such a guarantee is difficult due to two compounding sources of uncertainty: the error introduced when sensitive attributes are inferred from the graph alone, and the way this error is carried through the generation process and reshapes the joint distribution of node attributes and edges.

To address above challenges, in this paper, *Certified Proxy-guided Fair Graph Generation* (**CertiFG**), a novel fairness-aware autoregressive graph generation framework that enables fair graph synthesis without sensitive attributes. *To the best of our knowledge, this is the first work to explore fair graph generation without sensitive attributes and to provide theoretical guarantees that quantify how errors in inferred sensitive proxies affect the fairness of the generated graphs, offering explicit guidance for end users to assess and interpret fair graph generation models in this setting.* Specifically, we first develop a sensitive proxy inference module that infers a sensitive proxy from the observed graph data. We then design a proxy-error-aware fairness regularizer that operates on the generator's representations to reduce bias in the generated graphs while reducing the impact of inaccurate proxy assignments. Finally, we provide theoretical guarantees on the fairness of the generated graphs by deriving a closed-form upper bound that explicitly quantifies how proxy inference errors propagate to the final fairness violation under the unobserved true sensitive attributes, offering end users a principled way to assess the trustworthiness of the generated graphs before deployment. Our contributions are as follows:

- We study a new and practical problem of fair graph generation without observed sensitive attributes, and propose CertiFG, a certified proxy-guided fair graph generation framework that integrates graph-aware proxy inference with proxy-error-aware fair graph generation.
- We provide theoretical guarantees that explicitly characterize how proxy inference errors affect the fairness of generated graphs. Our analysis connects measurable proxy-level fairness constraints with true fairness violations under unobserved sensitive attributes, offering end users clear guidance for interpreting fairness claims without sensitive attribute.
- We conducted extensive experiments on four real-world datasets, demonstrating the superior performance of CertiFG over other state-of-the-art methods.

2 Related Works

Graph Generation Models. Graph generation aims to learn the underlying distribution of observed graphs and produce new ones that preserve their structural and attribute characteristics [17], and a wide range of deep generative methods have been developed for this purpose [18]. Existing methods can be broadly grouped into one-shot and autoregressive models. One-shot models [19, 20, 21], often

built on variational autoencoders, generative adversarial networks, or diffusion-based backbones, generate the entire graph in a single pass. While straightforward to train, these models rely on independence assumptions across nodes and edges, which limits their ability to capture the complex local dependencies that characterize real-world graphs. To this end, autoregressive models [22, 23, 24] construct graphs step by step by adding nodes and edges in sequence, allowing each generation step to be conditioned on previously generated components, thereby enabling a more accurate modeling of the structural patterns observed in real-world graphs. Despite the great progress, these works do not consider fairness. As synthetic graphs are increasingly used in high-stakes decision-making [25], this gap leaves the generated graphs at risk of inheriting and even amplifying biases from the training data, motivating the need for graph generation methods that explicitly address fairness.

Fairness in Graph Generation Model. Despite their great success, most existing graph generation models focus only on utility performance while ignoring the fairness issue, which can lead to biased generation [7]. To address this, several efforts have been made toward fair graph generation [6, 26, 27]. Specifically, most existing fair graph generation works mainly focus on mitigating graph structure bias. For example, FAIRLP [28] adjusts the training graph to balance intra-group and inter-group link distributions, thereby enhancing representation fairness. More recently, a few works have begun to explore feature bias mitigation. For instance, FairGen [27] introduces fairness constraints to minimize subgroup reconstruction differences. However, these existing approaches rely on full access to sensitive attributes during training to enforce fairness constraints that mitigate graph structure bias and node feature bias, which significantly limits their applicability in real-world scenarios where sensitive attributes are often unavailable. In addition, existing work on fairness without sensitive attributes mainly focuses on classification tasks and cannot be easily extended to graph generation, since the inferred sensitive proxy directly guides the generation process and inference errors can therefore propagate into the generated graph [29].

Different from the above works, this paper takes a step toward fair graph generation without sensitive attributes, which has not been addressed by existing methods. Our framework infers a sensitive proxy from the observed graph and explicitly accounts for proxy inference errors during fair graph generation, supported by theoretical guarantees on the fairness of the generated graph. In addition, the proposed error-aware design is general and can be readily extended to other graph learning tasks without sensitive attributes.

3 Notation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be an undirected graph, where $\mathcal{V}$ is the node set with n nodes, $\mathcal{E}$ is the edge set with m edges. The graph topology is encoded by an adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$, where $\mathbf{A}_{u,k} = 1$ if an edge $e_{u,k} \in \mathcal{E}$ connects nodes u and k, and $\mathbf{A}_{u,k} = 0$ otherwise. The sensitive attribute of all nodes is collected in $S \in \{0,1\}^n$, with s_i denoting the value of node v_i. Based on S, the node set is partitioned into two subgroups, $\mathcal{S}_0 = \{v_i \mid s_i = 0\}$ and $\mathcal{S}_1 = \{v_i \mid s_i = 1\}$, which respectively correspond to the two sensitive groups. We let $\hat{S}$ denote the inferred sensitive proxy. For clarity of presentation, we describe the method and proofs under a binary sensitive attribute setting throughout the paper.

4 Methodology

4.1 Bias Analysis

This section presents the theoretical analysis that grounds the proposed framework, including the bias mechanisms in graph generation and the fairness guarantees on the generated graphs. Our framework follows the autoregressive graph generation paradigm, where the graph is constructed step by step and the underlying representations are learned through message passing over the generated graph. We adopt this paradigm because generating each node conditioned on previously generated nodes captures the rich local dependencies of real-world graphs, which is harder to achieve in one-shot models that produce the entire graph in a single pass under strong independence assumptions. Among autoregressive methods, we instantiate our framework on top of a representative graph diffusion model [24], which preserves the high generation quality of diffusion-based approaches while achieving significantly faster sampling than non-autoregressive graph diffusion models. We

begin our methodology by analyzing how bias arises in graph generation models, which lays the foundation for the proposed fairness regularization.

Given that the graph generator is trained to approximate the distribution of the observed graphs, any sensitive-related dependence in the training data is also learned and reproduced in the generated graphs. In graph data, this dependence appears through two main mechanisms. The first is graph structure bias, which arises when nodes sharing the same sensitive attribute are over-connected with one another, so a generator trained on graphs of this kind not only reproduces this connectivity pattern but tends to push it further, producing a topology in which sensitive groups are more structurally segregated than in the input data. The second is node feature bias, which arises from sensitive-related information embedded in node features and is amplified during feature reconstruction, increasing the gap between the feature distributions of different sensitive groups in the generated graph. These two mechanisms further reinforce each other through message passing, where a segregated topology pushes the generator to aggregate features mainly from same-group neighbors, and the resulting group-aligned representations are then translated back into close intra connections during generation.

This issue becomes more serious in diffusion-based graph generation. To formalize the above analysis, we let a diffusion-based graph generator defines a forward transition $q(x_t \mid x_{t-1})$ that gradually corrupts the observed graph $x_0 = (\mathbf{A}, \mathbf{X})$ into noise, and learns a reverse transition $p_\theta(x_{t-1} \mid x_t)$ that recovers node attributes and graph structures from noise. At denoising step t, we denote the hidden node representation produced by the denoising network as $\mathbf{H}^{(t)} = f_\theta(\mathbf{X}^{(t)}, \mathbf{A}^{(t)}, t)$, with $\mathbf{h}_i^{(t)}$ denoting the representation of node v_i. The generated edge probability between nodes v_i and v_j is then obtained as $\tilde{\mathbf{A}}_{ij}^{(t-1)} = d_\theta\left(\mathbf{h}_i^{(t)}, \mathbf{h}_j^{(t)}, t\right)$, where $d_\theta(\cdot)$ is the edge decoder, and the generated node attributes are decoded from the same node representations. Therefore, bias in graph generation first appears as group disparity in the node representations $\mathbf{h}_i^{(t)}$, and these representation differences are then translated into biased node attributes and biased connection probabilities.

To this end, we aim to measure the group disparity carried by the node representations at each denoising step. Specifically, we define the representation disparity of sensitive group $\mathcal{S}_k$ at step t as:

$$\delta_S^{(t)} = \left\| \frac{1}{N_S} \sum_{v_i \in \mathcal{S}_0} \mathbf{h}_i^{(t)} - \frac{1}{N - N_S} \sum_{v_i \notin \mathcal{S}_0} \mathbf{h}_i^{(t)} \right\|_2 \tag{1}$$

where N_S denotes the number of nodes in sensitive group $\mathcal{S}_0$, and $N - N_S$ denotes the number of nodes outside this group. A large $\delta_S^{(t)}$ indicates that the node representations at step t contain strong sensitive-related information, leading to group disparity in the representation space, which is in turn passed to the decoded node attributes and connection probabilities.

Armed with this representation disparity measurement, we proceed in three steps. First, we analyze how the representation disparity is shaped by the message-passing-based denoising network and establish an upper bound that characterizes its propagation across denoising layers (Theorem 4.1). Second, we extend this upper bound to the setting where sensitive groups are not directly observed and the disparity has to be evaluated on inferred sensitive proxies $\hat{S}$, quantifying how proxy errors enlarge the upper bound on top of the message-passing term (Theorem 4.2). Third, we connect the bounded representation disparity to the statistical parity violation of the generated graph, showing that controlling $\delta_S^{(t)}$ across denoising steps is sufficient to bound the unfairness of the final generated graph (Theorem 4.3). Guided by these results, we then propose a fairness regularizer that explicitly accounts for proxy uncertainty and reduces both structural and feature bias during graph generation.

We begin with the first step of analyzing the representation disparity. In a message-passing-based denoising network, the representation of each node is updated by aggregating information from its neighbors, so any group-dependent connectivity in the current noisy graph is implicitly absorbed into the next-layer representations. Theorem 4.1 quantifies this effect by providing an upper bound on the representation disparity between a sensitive group $\mathcal{S}_d$ and the rest of the nodes at the l-th layer of the denoising network, revealing how the disparity accumulates across layers (proof in Appendix).

Theorem 4.1 (Representation Disparity under Attention-based Denoising)**.** *Consider the l-th layer of the denoising network at diffusion step t, where the two sensitive groups are denoted as $\mathcal{S}_0$ and $\mathcal{S}_1$, with $N_0 = |\mathcal{S}_0|$ and $N_1 = |\mathcal{S}_1|$. The representation disparity between the two sensitive groups after the l^{th} denoising layer can be upper bounded by:*

$$\delta_0^{(l+1,t)} \le L\Bigg(\delta\sqrt{d^{(l+1)}}\Bigg(\left|\frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_0} a_{iu}^{(l)} - \frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_0} a_{ju}^{(l)}\right| + \left|\frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_1} a_{iu}^{(l)} - \frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_1} a_{ju}^{(l)}\right|\Bigg) + 2\left\|\boldsymbol{\Delta}^{(l)}\right\|_2 + 2\sqrt{N}\Delta_z\Bigg) \tag{2}$$

where $a_{iu}^{(l)}$ denotes the attention weight from node v_u to node v_i in the l-th denoising layer, and L is the Lipschitz constant.

Armed with Theorem 4.1, we further consider how the proxy inference error affects the upper bound. In our setting, the ground-truth sensitive attribute S is unavailable during training, so the disparity in Theorem 4.1 cannot be evaluated directly. Instead, the two groups have to be identified using the inferred proxy $\hat{S}$, and the representation disparity is computed on the proxy-based partition $\hat{\mathcal{S}}_0, \hat{\mathcal{S}}_1$. Since the inferred proxy cannot guarantee an exact recovery of the ground-truth sensitive attribute, the disparity computed on $\hat{S}$ does not coincide with the one in Theorem 4.1, and the additional gap depends on how often $\hat{S}$ disagrees with S. The following theorem extends Theorem 4.1 by quantifying how this proxy inference error inflates the message-passing bound computed on the proxy partition.

Theorem 4.2 (Representation Disparity under Inferred Sensitive Proxies)**.** *Under the same setting as Theorem 4.1, but with the ground-truth groups $\mathcal{S}_0, \mathcal{S}_1$ replaced by the proxy-based partition $\hat{\mathcal{S}}_0, \hat{\mathcal{S}}_1$ obtained from the inferred sensitive proxy $\hat{S}$, and under a mild assumption on the proxy error (formalized in the Appendix), the representation disparity between the ground-truth sensitive groups after the l^{th} denoising layer can be upper bounded by:*

$$\delta_S^{(l+1,t)} \le \frac{L}{1-\eta_0-\eta_1}\Bigg(\delta\sqrt{d^{(l+1)}}\Bigg(\left|\frac{1}{\hat{N}_0}\sum_{v_i\in\hat{\mathcal{S}}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\hat{\mathcal{S}}_0} a_{iu}^{(l)} - \frac{1}{\hat{N}_1}\sum_{v_j\in\hat{\mathcal{S}}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\hat{\mathcal{S}}_0} a_{ju}^{(l)}\right| + \left|\frac{1}{\hat{N}_0}\sum_{v_i\in\hat{\mathcal{S}}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\hat{\mathcal{S}}_1} a_{iu}^{(l)} - \frac{1}{\hat{N}_1}\sum_{v_j\in\hat{\mathcal{S}}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\hat{\mathcal{S}}_1} a_{ju}^{(l)}\right|\Bigg) + 2\left\|\boldsymbol{\Delta}^{(l)}\right\|_2 + 2\sqrt{N}\Delta_z\Bigg) \tag{3}$$

where $\hat{N}_0 = |\hat{\mathcal{S}}_0|$, $\hat{N}_1 = |\hat{\mathcal{S}}_1|$, and $\eta_0 = \mathbb{P}(S = 1 \mid \hat{S} = 0)$, $\eta_1 = \mathbb{P}(S = 0 \mid \hat{S} = 1)$ are the impurities of the two proxy groups, determined by the proxy transition matrix $\boldsymbol{T}$ and the marginal group distributions $\boldsymbol{p}$, $\hat{\boldsymbol{p}}$ of S and $\hat{S}$, with $\eta_0 + \eta_1 < 1$.

Building on Theorem 4.2, which bounds the representation disparity at every denoising layer when the model is trained on the inferred sensitive proxy $\hat{S}$ instead of the ground-truth sensitive attribute S, the bound consists of two parts: i) a message-passing term computed on the proxy partition, which is fully controllable through the denoising network during training, and ii) an inflation factor $1/(1-\eta_0-\eta_1)$ that depends only on the error structure of the proxy and quantifies the price of not observing S. When the proxy is exact ($\eta_0 = \eta_1 = 0$), the factor equals one and the bound reduces to Theorem 4.1; as $\eta_0 + \eta_1$ approaches one, the proxy carries no information about S and the factor diverges, so a less accurate proxy loosens the guarantee on the true sensitive groups. Together, these two parts characterize how far the representation disparity at any denoising layer can be pushed under our setting. Building on this, we further translate the layer-wise disparity bound from Theorem 4.2 into an upper bound on the statistical parity violation of the generated graph.

Theorem 4.3 (Demographic Parity Bound under Inferred Sensitive Proxies)**.** *Under the same setting as Theorem 4.2, if the edge decoder is L_{edge}-Lipschitz with respect to the node representations, the statistical parity violation $\Delta_{\text{DP}}^{(l+1,t)}$ of the generated graph can be upper bounded by:*

$$\Delta_{\mathrm{DP}}^{(l+1,t)} \leq L_{\mathrm{edge}}\Bigg(\frac{L}{1-\eta_0-\eta_1}\Bigg(\delta\sqrt{d^{(l+1)}}\Bigg(\left|\frac{1}{\hat{N}_0}\sum_{v_i\in\hat{\mathcal{S}}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\hat{\mathcal{S}}_0}a_{iu}^{(l)}-\frac{1}{\hat{N}_1}\sum_{v_j\in\hat{\mathcal{S}}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\hat{\mathcal{S}}_0}a_{ju}^{(l)}\right| + \left|\frac{1}{\hat{N}_0}\sum_{v_i\in\hat{\mathcal{S}}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\hat{\mathcal{S}}_1}a_{iu}^{(l)}-\frac{1}{\hat{N}_1}\sum_{v_j\in\hat{\mathcal{S}}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\hat{\mathcal{S}}_1}a_{ju}^{(l)}\right|\Bigg) + 2\left\|\boldsymbol{\Delta}^{(l)}\right\|_2 + 2\sqrt{N}\Delta_z\Bigg) + r_0^{(l+1,t)} + r_1^{(l+1,t)}\Bigg) \tag{4}$$

where $r_0^{(l+1,t)}$ and $r_1^{(l+1,t)}$ are the within-group representation variations of the two ground-truth sensitive groups (defined in the Appendix).

Based on above theoretical results, we propose a fairness regularizer that directly minimizes the bound in Theorem 4.3. The key insight from our analysis is that the statistical parity violation of the generated graph is controlled by two quantities: i) the layer-wise representation disparity $\delta_S^{(l,t)}$ and ii) the proxy inference error, which enters Equation (4) through the inflation factor $1/(1-\eta_0-\eta_1)$. To reduce the first quantity, the node representations used for graph generation must not encode sensitive-related signals. To control the second quantity, however, the model still needs sufficient sensitive-related information to infer the missing proxy $\hat{S}$, since a more accurate proxy reduces $\eta_0+\eta_1$ and thereby tightens the inflation factor. These two objectives create a tension: removing all sensitive-related information from the representations harms proxy inference, while preserving it harms fairness.

To this end, we propose a mask-based representation fairness regularizer $\mathcal{L}_F$. The core idea is to keep the original node representation $\mathbf{h}_i^{(l,t)}$ for sensitive proxy inference, while applying fairness constraints only to a masked representation derived from $\mathbf{h}_i^{(l,t)}$. This design allows the model to preserve the sensitive-related information needed to infer the missing proxy $\hat{S}$, while ensuring that the representation that drives graph generation is aligned across sensitive groups, which directly reduces $\delta_S^{(l,t)}$ at every denoising layer. Specifically, let $\mathbf{m}^{(l,t)} \in [0,1]^{d_h}$ be a learnable dimension-wise retention mask at the l^{th} denoising layer of diffusion step t. A larger value of $\mathbf{m}_j^{(l,t)}$ indicates that the j^{th} representation dimension is retained, while a smaller value indicates that this dimension is suppressed. The masked representation is then defined as:

$$\widetilde{\mathbf{h}}_i^{(l,t)} = \mathbf{m}^{(l,t)} \odot \mathbf{h}_i^{(l,t)} \tag{5}$$

where $\odot$ denotes element-wise multiplication.

To penalize the dependence between the masked representation and the inferred sensitive proxy, we adopt a covariance-based fairness regularizer:

$$\mathcal{L}_{\mathrm{F}}^{(l,t)} = \left\| \frac{1}{N}\sum_{i=1}^{N}\left(\hat{s}_i-\bar{\hat{s}}\right)\left(\widetilde{\mathbf{h}}_i^{(l,t)}-\bar{\widetilde{\mathbf{h}}}^{(l,t)}\right)\right\|_1 \tag{6}$$

where $\hat{s}_i$ is the inferred sensitive proxy of node v_i, $\bar{\hat{s}}$ is its mean over all nodes, and $\bar{\widetilde{\mathbf{h}}}^{(l,t)}$ is the mean of $\widetilde{\mathbf{h}}_i^{(l,t)}$ across all nodes. By driving $\mathcal{L}_{\mathrm{F}}^{(l,t)}$ towards zero, the masked representation becomes uncorrelated with the inferred sensitive proxy, which reduces the node representation disparity and thereby enhances the fairness of the generated graph.

4.2 Error-Insight Demographic Identification

Building on the bias analysis above, we now turn to the question of how to obtain the sensitive proxy $\hat{S}$ required by the fairness regularizer in the absence of ground-truth sensitive attributes. A simple solution is to train an auxiliary classifier on the observed graph and feed its predictions directly into the fairness regularizer. However, treating $\hat{S}$ as if it were the true sensitive attribute would propagate

inference errors into the graph generation process. To this end, our error-insight demographic identification module is designed by two tasks: i) producing a proxy $\hat{S}$ from the observed graph, and ii) measuring the gap between $\hat{S}$ and the ground-truth sensitive attribute S, so that the fairness regularizer can be applied while the impact of inference errors is reduced.

For the first task, we infer the sensitive proxy $\hat{S}$ from the observed graph by reusing the node representations learned by the graph generator. By doing so, the proxy is inferred from the same representations that the generator uses to produce node attributes and edges, which capture intricate patterns in high-dimensional data and incorporate the influence of graph links and non-sensitive attributes. Specifically, we use the node representation $\mathbf{h}_i^{(L,t)} \in \mathbb{R}^d$ produced by the last layer of the GAT-based reverse denoising backbone, and attach a proxy prediction head $f_\psi(\cdot)$ that compresses $\mathbf{h}_i^{(L,t)}$ into a sensitive proxy. Mathematically, this can be represented as:

$$\hat{s}_i^{(t)} = f_\psi\left(\mathbf{h}_i^{(L,t)}\right) \tag{7}$$

where $f_\psi(\cdot)$ outputs a soft proxy distribution over the two sensitive groups, and $\hat{s}_i \in \{0, 1\}$ is the resulting hard proxy label.

For the second task, we treat $\hat{S}$ as a noisy observation of the latent sensitive attribute and explicitly estimate its error structure. Quantifying this gap allows the fairness regularizer to act on the authentic group information, which in turn reduces the inflation factor $1/(1-\eta_0-\eta_1)$ in Theorem 4.3. To this end, we introduce a proxy transition matrix, where each entry $\widehat{T}_{s,\hat{s}} \approx \Pr(\hat{S} = \hat{s} \mid S = s)$, measures the probability that a node from the latent sensitive group $S = s$ is assigned to the inferred proxy group $\hat{S} = \hat{s}$. By doing so, we first estimate the latent sensitive-group prior $\widehat{\boldsymbol{p}}$ and the empirical proxy-group prior $\widetilde{\boldsymbol{p}}$, where the former describes the unobserved distribution of S and the latter is computed directly from the inferred proxy labels. These three quantities together provide the minimal information required to translate any proxy-based statistic back to the authentic group information.

However, given that S is not observable, we cannot directly calculate $\widehat{\boldsymbol{T}}$ and $\widehat{\boldsymbol{p}}$. To this end, we build on the observation that nodes in real-world graphs tend to connect with other nodes that share the same sensitive attribute. For example, in a social network, individuals of the same gender are more likely to interact with each other [30]; in a citation graph, papers from the same research community are more likely to cite one another [31]. In other words, for a given anchor node, its closely connected neighbors are likely to share its underlying sensitive attribute. Building on this, for each anchor node v_i, we sample two neighbors v_j and v_k, drawn respectively from its representation-nearest neighborhood. The three nodes in the triplet (v_i, v_j, v_k) are therefore likely to share the same sensitive attribute, while their inferred proxy labels $(\tilde{s}_i, \tilde{s}_j, \tilde{s}_k)$ effectively serve as three independent noisy observations of a single underlying S. The core idea behind this is that the joint statistics of three noisy observations sharing the same S are governed by the proxy-conditional probabilities $\Pr(\hat{S} \mid S)$ encoded in $\widehat{\boldsymbol{T}}$. As a result, the empirical co-occurrence patterns of $(\tilde{s}_i, \tilde{s}_j, \tilde{s}_k)$ aggregated across many triplets carry enough information to identify the entries of $\widehat{\boldsymbol{T}}$ together with the latent prior $\widehat{\boldsymbol{p}}$, without accessing S. Specifically, we recover $\widehat{\boldsymbol{T}}$ and $\widehat{\boldsymbol{p}}$ by matching the empirical frequencies of triplet label patterns to the moments of a binomial mixture. Conditional on the latent sensitive attribute shared by the triplet, the number of positive proxy labels among $(\tilde{s}_i, \tilde{s}_j, \tilde{s}_k)$ follows one of two binomial distributions whose parameters are exactly the columns of $\widehat{\boldsymbol{T}}$. Because three triplet observations provide three independent factorial moments, the binomial-mixture parameters and the latent prior are uniquely identified, and we recover $\widehat{\boldsymbol{T}}$ and $\widehat{\pi} = \Pr(S = 1)$ in closed form by solving a small system of moment equations [32]. The empirical proxy prior $\widetilde{\boldsymbol{p}}$ is computed directly from the inferred proxy labels without extra estimation.

Armed with $\widehat{\boldsymbol{T}}$, we train the proxy prediction head f_ψ in an error-aware manner that explicitly accounts for the gap between the latent sensitive attribute S and the inferred proxy $\hat{S}$. Specifically, we pass the soft output of $f_\psi(\mathbf{h}_i^{(L,t)})$ through the estimated transition matrix $\widehat{\boldsymbol{T}}$ to obtain the model-induced distribution over noisy proxy labels, and maximize the likelihood of the observed triplet labels under this distribution. Formally, the identification loss is defined as:

$$\mathcal{L}_{\mathrm{I}} = -\frac{1}{|\mathcal{V}_p|} \sum_{v_i \in \mathcal{V}_p} \log \left(\widehat{\boldsymbol{T}}^{\top} f_{\psi}(\mathbf{h}_i^{(L,t)}) \right) \tag{8}$$

where $\mathcal{V}_p \subseteq \mathcal{V}$ denotes the set of nodes that receive a weak proxy label $\tilde{s}_i$ through the triplet construction. By minimizing $\mathcal{L}_{\mathrm{id}}$, the proxy prediction head f_ψ is encouraged to recover the latent sensitive partition rather than to fit the systematic errors carried by $\tilde{s}_i$, so that the inferred proxy $\hat{S}$ becomes a tighter approximation of S, which directly reduces the inflation factor in Theorem 4.3.

4.3 Overall Optimization Objectives

We now describe the overall training objective of CertiFG, which combines a graph diffusion likelihood with the proposed proxy identification and fairness regularization. The graph diffusion model is trained in the standard forward–reverse manner [24]. In the forward process, an input graph G_0 is gradually corrupted by absorbing nodes and their incident edges according to a learned ordering σ sampled from $q_\phi(\sigma \mid G_0)$. In the reverse process, a denoising network $p_\theta(G_t \mid G_{t+1})$ reconstructs G_0 step by step. Following the variational lower bound under node permutation invariance, the denoising loss is defined as:

$$\mathcal{L}_P = -n\, \mathbb{E}_{\sigma_{1:n} \sim q_\phi(\sigma_{1:n}|G_0)}\, \mathbb{E}_{t \sim \mathcal{U}_n} \log p_\theta \left(O_{\sigma_t}^{\pi(>t)} \mid G_{t+1} \right) \tag{9}$$

where $\sigma_{1:n}$ is the node ordering sampled from $q_\phi(\sigma_{1:n} \mid G_0)$, t is drawn uniformly from $\{1, \ldots, n\}$, and $O_{\sigma_t}^{\pi(>t)}$ denotes the node v_{σ_t} together with its edges to the previously unmasked nodes $\{\sigma_{t+1}, \ldots, \sigma_n\}$. Minimizing $\mathcal{L}_P$ encourages the generator to recover both node attributes and graph structures from their noisy versions across all diffusion steps.

Building on this, we then jointly optimize $\mathcal{L}_P$ together with the proxy identification loss $\mathcal{L}_I$ and the calibrated fairness regularizer $\mathcal{L}_F$ to enforce fairness during graph generation. Therefore, the final training loss is defined as:

$$\mathcal{L} = \mathcal{L}_P + \lambda_I\, \mathcal{L}_I + \lambda_F\, \mathcal{L}_F \tag{10}$$

where $\lambda_I, \lambda_F > 0$ control the strength of error-aware proxy identification and calibrated fairness regularization, respectively.

5 Experiment

5.1 Experiment setting

Datasets and Baselines. We conduct experiments on four real-world graph datasets, including German [33], Cora [34], Pokec-z [35], and Photo [36], and compare our method with representative graph generation baselines, including GRAPHARM [24], GSDM [37], GraphVAE [38], GDSS [39], DiGress [40], and CatFlow [41]. We note that existing fair graph generation methods all require ground-truth sensitive attributes during training and therefore cannot be directly applied to our setting, so we compare CertiFG with state-of-the-art fairness-agnostic graph generation baselines. Detailed descriptions and statistics of the datasets and baselines are provided in Appendix D.1 and Appendix D.2, respectively.

Evaluation Metrics. We assess our method from two complementary aspects. **i) Graph Generation Quality:** Following [24], we adopt Maximum Mean Discrepancy (MMD) to measure the distributional difference between generated and real graphs. Following previous work [26], we evaluate graph generation quality using degree distribution (DD), clustering coefficient (Clus), and node feature distribution (NFea), and assess generation fairness using Fair Degree Distribution (Fair-DD), Fair Clustering Coefficient (Fair-Clus), and Fair Node Feature (Fair-NFea). All metrics are measured using Maximum Mean Discrepancy (MMD), where lower values indicate better generation quality or smaller subgroup disparities, respectively. **ii) Downstream task:** We evaluate the utility of generated graphs by training downstream task models and reporting AUC and F1 scores. We also measure

Table 1: Graph generation results on German and Cora datasets. Lower values indicate better performance. (the color cells show top rank)

Dataset	Method	DD	Clus	NFea	Fair-DD	Fair-Clus	Fair-NFea
German	GRAPHARM	0.337	0.148	0.242	0.087	0.074	0.057
	GSDM	0.366	0.177	0.289	0.083	0.071	0.052
	GraphVAE	0.381	0.183	0.302	0.074	0.061	0.049
	GDSS	0.392	0.190	0.318	0.088	0.073	0.043
	DIGRESS	0.377	0.176	0.291	0.080	0.066	0.050
	CatFlow	0.353	0.146	0.238	0.093	0.076	0.058
	CertiFG	0.364	0.172	0.286	0.048	0.034	0.023
Cora	GRAPHARM	0.238	0.135	0.312	0.077	0.083	0.049
	GSDM	0.241	0.128	0.289	0.064	0.069	0.051
	GraphVAE	0.263	0.169	0.335	0.072	0.078	0.042
	GDSS	0.275	0.173	0.348	0.073	0.075	0.051
	DIGRESS	0.259	0.167	0.332	0.063	0.060	0.047
	CatFlow	0.253	0.142	0.307	0.070	0.065	0.049
	CertiFG	0.263	0.156	0.318	0.036	0.030	0.025

classification fairness using Δ_{DP} [42] and Δ_{EO} [43], where lower values indicate fairer prediction outcomes.

5.2 Experiment Results

Comparison Study. We compare the graph generative performance and fairness of CertiFG with other baselines across all four datasets, and report the average performances along with standard deviations of each metric in Table 1 for the German and Cora datasets, with the results on the Pokec-z and Photo datasets provided in Appendix due to space limitations. From the results, we have the following key observations: i) fairness-agnostic baselines achieve competitive generation quality, but exhibit substantial fairness violations across all datasets. The reason is that these models are trained purely to recover the data distribution of the observed graphs, including the sensitive-related dependencies between node attributes and graph structures, so the bias in the training graphs is naturally inherited and even amplified during generation. ii) CertiFG consistently achieves the best fairness performance across all datasets, while maintaining generation quality comparable to the fairness-agnostic baselines. The reason is that CertiFG jointly considers sensitive proxy inference and the impact of inference errors during graph generation, which prevents the inferred proxy from injecting noise into the generation process. With this design, the regularizer can accurately remove the sensitive-related information from the learned representations while preserving the task-related information that drives graph generation. Hence, CertiFG achieves fairness improvements without significant degradation in generation quality.

Downstream Task. Beyond directly comparing the quality and fairness of the generated graphs, we further evaluate whether the synthetic graphs produced by CertiFG remain useful for downstream tasks while preserving fairness. Specifically, we use each method to generate a synthetic version of the training graph and then train a standard GNN classifier on the resulting graph for the downstream task. Using the Cora dataset as a case study, Table 2 reports the comparison between CertiFG and the baseline methods on this task. The results show that classifiers trained on graphs generated by CertiFG consistently achieve better fairness while maintaining classification AUC comparable to those trained on graphs from fairness-agnostic baselines. The reason is that CertiFG removes the sensitive-related dependencies during graph generation, so the generated graphs no longer carry strong demographic-related information into the downstream training, leading to fairer classifiers. These results confirm that CertiFG produces synthetic graphs that are not only intrinsically fair but also support fair downstream decision-making.

Ablation Study. To verify the effectiveness of our proposed components, we compare CertiFG with two ablated variants: i) **CertiFG-WI**, which removes the error-aware proxy identification loss $\mathcal{L}_I$; and ii) **CertiFG-WF**, which removes the calibrated fairness regularizer $\mathcal{L}_F$ and trains the generator with the standard denoising loss only. The results on the German and Cora datasets are illustrated in Figure 1, and additional results are provided in the Appendix. First, we observe that CertiFG consistently achieves the best fairness results across both datasets, confirming that

Table 2: Downstream task results on Cora dataset. (the color cells show top rank)

Dataset	Method	AUC (%)	F1-score (%)	Δ_{DP} (%)	Δ_{EO} (%)
Cora	Original-GCN	91.56 ± 1.86	83.52 ± 2.14	56.38 ± 2.07	13.90 ± 1.34
	GRAPHARM-GCN	87.34 ± 1.64	82.52 ± 2.18	33.21 ± 1.75	18.75 ± 1.06
	GSDM-GCN	85.75 ± 1.93	77.84 ± 1.87	24.45 ± 1.68	17.56 ± 1.23
	GraphVAE-GCN	80.82 ± 2.16	71.75 ± 1.92	35.92 ± 2.85	20.22 ± 2.31
	GDSS-GCN	86.86 ± 1.57	79.62 ± 2.24	28.65 ± 1.86	14.28 ± 1.42
	DIGRESS-GCN	84.22 ± 1.38	76.54 ± 1.73	33.08 ± 1.24	19.52 ± 1.17
	CatFlow-GCN	82.85 ± 1.26	74.02 ± 1.49	41.86 ± 1.92	21.83 ± 1.76
	CertiFG-GCN	85.68 ± 1.47	76.24 ± 1.35	18.64 ± 1.53	6.78 ± 0.89

combining the two proposed components leads to the most fair generation results. Second, removing the fairness regularizer $\mathcal{L}_F$ leads to the worst fairness results. Without $\mathcal{L}_F$, the generator has no mechanism to suppress the bias-inducing dependencies between sensitive-related information and graph components, and the bias in the training data is therefore directly inherited by the generated graphs. Third, removing the error-aware proxy identification loss also degrades fairness, since the proxy prediction head becomes more easily misled by the systematic errors in the noisy proxy labels, which then propagate into the fairness regularizer and weaken its effectiveness.

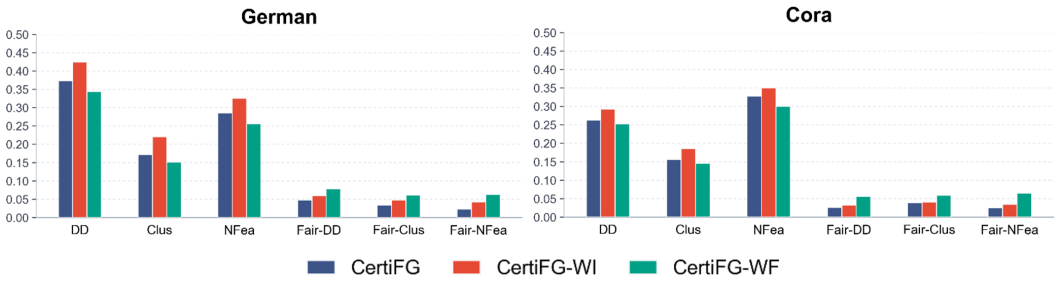


Figure 1: Ablation study for CertiFG, CertiFG-WI, and CertiFG-WF in German and Cora datasets.

6 Conclusion

Fair graph generation has emerged as an important research direction given the growing use of synthetic graphs in high-stakes decision-making applications. Despite recent progress, existing approaches assume full access to sensitive attributes, which is rarely available in real-world scenario and leaves the problem of fair graph generation without sensitive attributes largely open. This work takes a step toward filling this gap by theoretically analyzing how bias propagates through graph generation models when sensitive attributes are unobserved, and proposing CertiFG, a framework that infers a sensitive proxy while mitigating the impact of inference errors. Supported by theoretical guarantees, CertiFG demonstrates strong fairness improvements without sacrificing generation quality across extensive experiments on real-world datasets. Beyond these results, this work points to a broader research direction on fair graph generation in settings where sensitive information is complete missing.

Acknowledgements

This work was supported in part by the National Science Foundation (NSF) under Grant No. 2404039.

References

[1] Paolo Frazzetto, Muhammad Uzair Ul Haq, Flavia Fabris, and Alessandro Sperduti. Graph neural networks for candidate-job matching: An inductive learning approach: P. frazzetto et al. *Data Science and Engineering*, pages 1–18, 2025.

[2] Karan Bhanot, Miao Qi, John S Erickson, Isabelle Guyon, and Kristin P Bennett. The problem of fairness in synthetic healthcare data. *Entropy*, 23(9):1165, 2021.

[3] Zichong Wang, Giri Narasimhan, Xin Yao, and Wenbin Zhang. Mitigating multisource biases in graph neural networks via real counterfactual samples. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 638–647. IEEE, 2023.

[4] Tong Zhao, Wei Jin, Yozen Liu, Yingheng Wang, Gang Liu, Stephan Günnemann, Neil Shah, and Meng Jiang. Graph data augmentation for graph machine learning: A survey. *arXiv preprint arXiv:2202.08871*, 2022.

[5] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE transactions on knowledge and data engineering*, 35(12):12012–12038, 2021.

[6] O Deniz Kose and Yanning Shen. Fairwire: Fair graph generation. *arXiv preprint arXiv:2402.04383*, 2024.

[7] 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(10):10583–10602, 2023.

[8] April Chen, Ryan A Rossi, Namyong Park, Puja Trivedi, Yu Wang, Tong Yu, Sungchul Kim, Franck Dernoncourt, and Nesreen K Ahmed. Fairness-aware graph neural networks: A survey. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–23, 2024.

[9] Wenbin Zhang, Shuigeng Zhou, Toby Walsh, and Jeremy C Weiss. Fairness amidst non-iid graph data: A literature review. *AI Magazine*, 46(1):e12212, 2025.

[10] Zichong Wang and Wenbin Zhang. Fdgen: A fairness-aware graph generation model. In *Forty-second International Conference on Machine Learning*, 2025.

[11] Ying Zhao, Kai Zhang, Longxiang Gao, and Jinjun Chen. Privacy and fairness analysis in the post-processed differential privacy framework. *IEEE Transactions on Information Forensics and Security*, 20:2412–2423, 2025.

[12] Junyi Chai, Taeuk Jang, and Xiaoqian Wang. Fairness without demographics through knowledge distillation. *Advances in Neural Information Processing Systems*, 35:19152–19164, 2022.

[13] Zichong Wang, Jie Yang, Jun Zhuang, Puqing Jiang, Mingzhe Chen, Ye Hu, and Wenbin Zhang. Fair graph learning with limited sensitive attribute information. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 39423–39431, Mar. 2026.

[14] Erik Altman, Jovan Blanuša, Luc Von Niederhäusern, Béni Egressy, Andreea Anghel, and Kubilay Atasu. Realistic synthetic financial transactions for anti-money laundering models. *Advances in Neural Information Processing Systems*, 36:29851–29874, 2023.

[15] Zichong Wang, Zhipeng Yin, Roland HC Yap, and Wenbin Zhang. Ai fairness beyond complete demographics: Current achievements and future directions. In *ECAI 2025: 28th European Conference on Artificial Intelligence 25-30 October 2025, Bologna, Italy-Including 14th Conference on Prestigious Applications of Intelligent Systems (PAIS 2025) Proceedings*, pages 975–984. SAGE Publications, 1 Oliver's Yard, 55 City Road, London, EC1Y 1SP, 2025.

[16] Zichong Wang, Anqi Wu, Nuno Moniz, Shu Hu, Bart P Knijnenburg, Xingquan Zhu, and Wenbin Zhang. Towards fairness with limited demographics via disentangled learning. In *IJCAI*, pages 565–573, 2025.

[17] Xiaojie Guo and Liang Zhao. A systematic survey on deep generative models for graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5370–5390, 2022.

[18] Yanqiao Zhu, Yuanqi Du, Yinkai Wang, Yichen Xu, Jieyu Zhang, Qiang Liu, and Shu Wu. A survey on deep graph generation: Methods and applications. In *Learning on Graphs Conference*, pages 47–1. PMLR, 2022.

[19] Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

[20] Łukasz Maziarka, Agnieszka Pocha, Jan Kaczmarczyk, Krzysztof Rataj, Tomasz Danel, and Michał Warchoł. Mol-cyclegan: a generative model for molecular optimization. *Journal of Cheminformatics*, 12(1):2, 2020.

[21] Pengda Wang, Zhaowei Liu, Zhanyu Wang, Zongxing Zhao, Dong Yang, and Weiqing Yan. Graph generative adversarial networks with evolutionary algorithm. *Applied Soft Computing*, page 111981, 2024.

[22] Hanjun Dai, Azade Nazi, Yujia Li, Bo Dai, and Dale Schuurmans. Scalable deep generative modeling for sparse graphs. In *International conference on machine learning*, pages 2302–2312. PMLR, 2020.

[23] Chengyi Liu, Wenqi Fan, Yunqing Liu, Jiatong Li, Hang Li, Hui Liu, Jiliang Tang, and Qing Li. Generative diffusion models on graphs: Methods and applications. *arXiv preprint arXiv:2302.02591*, 2023.

[24] Lingkai Kong, Jiaming Cui, Haotian Sun, Yuchen Zhuang, B Aditya Prakash, and Chao Zhang. Autoregressive diffusion model for graph generation. In *International conference on machine learning*, pages 17391–17408. PMLR, 2023.

[25] Bukhoree Sahoh and Anant Choksuriwong. The role of explainable artificial intelligence in high-stakes decision-making systems: a systematic review. *Journal of Ambient Intelligence and Humanized Computing*, 14(6):7827–7843, 2023.

[26] Zichong Wang, Zhipeng Yin, and Wenbin Zhang. A unified framework for fair graph generation: Theoretical guarantees and empirical advances. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[27] Lecheng Zheng, Dawei Zhou, Hanghang Tong, Jiejun Xu, Yada Zhu, and Jingrui He. Fairgen: Towards fair graph generation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 2285–2297. IEEE, 2024.

[28] Yanying Li, Xiuling Wang, Yue Ning, and Hui Wang. Fairlp: Towards fair link prediction on social network graphs. In *Proceedings of the international AAAI conference on web and social media*, volume 16, pages 628–639, 2022.

[29] Patrik Joslin Kenfack, Samira Ebrahimi Kahou, and Ulrich Aïvodji. A survey on fairness without demographics. *Transactions on Machine Learning Research*, 2024.

[30] Jing Ma, Ruocheng Guo, Mengting Wan, Longqi Yang, Aidong Zhang, and Jundong Li. Learning fair node representations with graph counterfactual fairness. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 695–703, 2022.

[31] Shreya Chandrasekharan, Mariam Zaka, Stephen Gallo, Wenxi Zhao, Dmitriy Korobskiy, Tandy Warnow, and George Chacko. Finding scientific communities in citation graphs: Articles and authors. *Quantitative Science Studies*, 2(1):184–203, 2021.

[32] Zhaowei Zhu, Yiwen Song, and Yang Liu. Clusterability as an alternative to anchor points when learning with noisy labels. In *International Conference on Machine Learning*, pages 12912–12923. PMLR, 2021.

[33] Arthur Asuncion, David Newman, et al. Uci machine learning repository, 2007.

[34] 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.

[35] Lubos Takac and Michal Zabovsky. Data analysis in public social networks. In *International scientific conference and international workshop present day trends of innovations*, volume 1, 2012.

[36] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[37] Tianze Luo, Zhanfeng Mo, and Sinno Jialin Pan. Fast graph generation via spectral diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3496–3508, 2023.

[38] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*, pages 412–422. Springer, 2018.

[39] Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International Conference on Machine Learning*, 2022.

[40] Clément Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. In *International Conference on Learning Representations*, 2023.

[41] Floor Eijkelboom, Grigory Bartosh, Christian Andersson Naesseth, Max Welling, and Jan-Willem van de Meent. Variational flow matching for graph generation. In *Advances in Neural Information Processing Systems*, 2024.

[42] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[43] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

[44] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[45] Zichong Wang, Zhipeng Yin, Mo Sha, Xiaofeng Gao, Xiaoli Li, and Wenbin Zhang. Towards fair graph learning without demographic supervision. In Diego Calvanese, editor, *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-26*, pages 789–797. International Joint Conferences on Artificial Intelligence Organization, 8 2026. Main Track.

[46] Zichong Wang, Zhipeng Yin, Zhong Chen, Jack Yang, Jun Liu, and Wenbin Zhang. Learning counterfactual fairness from authentic generation. In Diego Calvanese, editor, *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-26*, pages 798–806. International Joint Conferences on Artificial Intelligence Organization, 8 2026. Main Track.

[47] Zhipeng Yin, Zichong Wang, Zhong Chen, Jack Yang, Xin Ning, and Wenbin Zhang. Disentangled graph-enhanced large language models for fair learning. In Diego Calvanese, editor, *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-26*, pages 844–852. International Joint Conferences on Artificial Intelligence Organization, 8 2026. Main Track.

[48] Zichong Wang, Tongliang Liu, and Wenbin Zhang. Guic: Certified graph unlearning with individual fairness guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 35820–35828, 2026.

[49] Zichong Wang, Nhat Hoang, Xingyu Zhang, Kevin Bello, Xiangliang Zhang, Sundararaja Sitharama Iyengar, and Wenbin Zhang. Towards fair graph learning without demographic information. In *AISTATS*, pages 2107–2115, 2025.

[50] Zichong Wang, Zhibo Chu, Thang Viet Doan, Shaowei Wang, Yongkai Wu, Vasile Palade, and Wenbin Zhang. Fair graph u-net: A fair graph learning framework integrating group and individual awareness. In *proceedings of the AAAI conference on artificial intelligence*, volume 39, pages 28485–28493, 2025.

[51] Zichong Wang, Zhipeng Yin, Liping Yang, Jun Zhuang, Rui Yu, Qingzhao Kong, and Wenbin Zhang. Fairness-aware graph representation learning with limited demographic information. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 354–371. Springer, 2025.

[52] Zichong Wang, Fang Liu, Shimei Pan, Jun Liu, Fahad Saeed, Meikang Qiu, and Wenbin Zhang. fairgnn-wod: Fair graph learning without complete demographics. International Joint Conferences on Artificial Intelligence Organization, 2025.

A Proof of Theorem 4.1

Here, we prove Theorem 4.1. Without loss of generality, we focus on the l^{th} GNN layer in the reverse denoising network at a fixed denoising step t. For notation simplicity, we omit the superscript t in the proof when there is no ambiguity. The input node representations are denoted by $\mathbf{h}^l$, and the output node representations are denoted by $\mathbf{h}^{l+1}$. In here, we use Graph Attention Networks (GAT) [44] as a representative example. Hence, the layer-wise message passing can be written as:

$$\mathbf{h}_{i,\overline{L}}^{(l+1)} = \sum_{u \in \widetilde{\mathcal{N}}(i)} a_{iu}^{(l)} \, \mathbf{h}_{u,\overline{L}}^{(l)}, \qquad a_{iu}^{(l)} = \frac{\exp(e_{iu}^{(l)})}{\sum_{v \in \widetilde{\mathcal{N}}(i)} \exp(e_{iv}^{(l)})} \tag{11}$$

where $\widetilde{\mathcal{N}}(i)$ denotes the denoised neighborhood of node v_i at step t, and $a_{iu}^{(l)}$ is the attention weight from node v_u to node v_i. Since the attention weights are normalized, we have:

$$\sum_{u \in \widetilde{\mathcal{N}}(i)} a_{iu}^{(l)} = 1, \qquad a_{iu}^{(l)} \geq 0. \tag{12}$$

For clarity, we first prove the result for binary sensitive groups $\mathcal{S}_0$ and $\mathcal{S}_1$. The extension to the general case $\mathcal{S}_k$ versus $\mathcal{V} \setminus \mathcal{S}_k$ follows by replacing $\mathcal{S}_0$ with $\mathcal{S}_k$ and $\mathcal{S}_1$ with $\mathcal{V} \setminus \mathcal{S}_k$. Let $N_S = |\mathcal{S}_0|$ and $N_{\neg S} = N - N_S$. The representation disparity after the l^{th} message-passing layer is:

$$\delta_k^{l+1,t} = \left\| \frac{1}{N_k} \sum_{v_i \in \mathcal{S}_k} \mathbf{h}_i^{l+1} - \frac{1}{N_{\neg k}} \sum_{v_i \notin \mathcal{S}_k} \mathbf{h}_i^{l+1} \right\|_2 \tag{13}$$

Hence, we can have the group-wise mean representations at layer l are defined as:

$$\boldsymbol{\mu}_0^{(l)} = \frac{1}{N_0} \sum_{i \in \mathcal{S}_0} \mathbf{h}_i^{(l)}, \qquad \boldsymbol{\mu}_1^{(l)} = \frac{1}{N_1} \sum_{j \in \mathcal{S}_1} \mathbf{h}_j^{(l)} \tag{14}$$

Thus, the representation disparity before the l^{th} message-passing layer is:

$$\delta_0^{(l,t)} = \left\| \boldsymbol{\mu}_0^{(l)} - \boldsymbol{\mu}_1^{(l)} \right\|_2 \tag{15}$$

We now decompose the GAT aggregation according to sensitive groups. For a node $v_i \in \mathcal{S}_0$, define its cross-group attention mass as:

$$\beta_i^{(l)} = \sum_{u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_1} a_{iu}^{(l)} \tag{16}$$

Then the attention mass assigned to nodes in the same group $\mathcal{S}_0$ is:

$$\sum_{u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_0} a_{iu}^{(l)} = 1 - \beta_i^{(l)} \tag{17}$$

Similarly, for a node $v_j \in \mathcal{S}_1$, we can define it as:

$$\beta_j^{(l)} = \sum_{u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_0} a_{ju}^{(l)} \tag{18}$$

which measures how much attention node v_j assigns to the opposite sensitive group.

Next, for every node $v_u \in \mathcal{S}_0$, write its layer-l representation as:

$$\mathbf{h}_u^{(l)} = \boldsymbol{\mu}_0^{(l)} + \boldsymbol{\epsilon}_u^{(l,0)} \tag{19}$$

where $\boldsymbol{\epsilon}_u^{(l,0)}$ is the within-group deviation from the mean of $\mathcal{S}_0$.

Similarly, for every node $v_u \in \mathcal{S}_1$, we have:

$$\mathbf{h}_u^{(l)} = \boldsymbol{\mu}_1^{(l)} + \boldsymbol{\epsilon}_u^{(l,1)} \tag{20}$$

To measure the possible variation of node representations inside each group, we define the coordinate-wise maximal deviation from the group mean. Hence, we have:

$$\boldsymbol{\Delta}_0^{(l)} = \max_{v_u \in \mathcal{S}_0} \left| \mathbf{h}_u^{(l)} - \boldsymbol{\mu}_0^{(l)} \right|, \quad \boldsymbol{\Delta}_1^{(l)} = \max_{v_u \in \mathcal{S}_1} \left| \mathbf{h}_u^{(l)} - \boldsymbol{\mu}_1^{(l)} \right| \tag{21}$$

Building on this, for any node $(v_u \in \mathcal{S}_0$, we have:

$$\mathbf{h}_u^{(l)} \in \left[\boldsymbol{\mu}_0^{(l)} \pm \boldsymbol{\Delta}_0^{(l)} \right] \tag{22}$$

Similar, for any node $v_u \in \mathcal{S}_1$, we have:

$$\mathbf{h}_u^{(l)} \in \left[\boldsymbol{\mu}_1^{(l)} \pm \boldsymbol{\Delta}_1^{(l)} \right] \tag{23}$$

Therefore, for a node $v_i \in \mathcal{S}_0$, the GAT aggregation can be decomposed into the contribution from nodes with the same sensitive attribute and the contribution from nodes with a different sensitive attribute:

$$\mathbf{h}_i^{(l+1)} = \sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_0} a_{iu}^{(l)} \mathbf{h}_u^{(l)} + \sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_1} a_{iu}^{(l)} \mathbf{h}_u^{(l)} \tag{24}$$

Since $a_{iu}^{(l)} \geq 0$, substituting the above interval bounds gives:

$$\begin{aligned} \mathbf{h}_i^{(l+1)} \in \Bigg[& \left(\sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_0} a_{iu}^{(l)} \right) \boldsymbol{\mu}_0^{(l)} + \left(\sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_1} a_{iu}^{(l)} \right) \boldsymbol{\mu}_1^{(l)} \\ & \pm \left(\left(\sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_0} a_{iu}^{(l)} \right) \boldsymbol{\Delta}_0^{(l)} + \left(\sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_1} a_{iu}^{(l)} \right) \boldsymbol{\Delta}_1^{(l)} \right) \Bigg] \end{aligned} \tag{25}$$

Similarly, for a node $v_j \in \mathcal{S}_1$, we have:

$$\mathbf{h}_j^{(l+1)} = \sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_1} a_{ju}^{(l)} \mathbf{h}_u^{(l)} + \sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_0} a_{ju}^{(l)} \mathbf{h}_u^{(l)} \tag{26}$$

and therefore we can also obtain:

$$\begin{aligned} \mathbf{h}_j^{(l+1)} \in \Bigg[& \left(\sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_1} a_{ju}^{(l)} \right) \boldsymbol{\mu}_1^{(l)} + \left(\sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_0} a_{ju}^{(l)} \right) \boldsymbol{\mu}_0^{(l)} \\ & \pm \left(\left(\sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_1} a_{ju}^{(l)} \right) \boldsymbol{\Delta}_1^{(l)} + \left(\sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_0} a_{ju}^{(l)} \right) \boldsymbol{\Delta}_0^{(l)} \right) \Bigg] \end{aligned} \tag{27}$$

Next, we take the average over all nodes in $\mathcal{S}_0$. Hence, we have

$$\frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\mathbf{h}_{i,\overline{L}}^{(l+1)} \in \Bigg[\boldsymbol{\mu}_0^{(l)} + \frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_1} a_{iu}^{(l)}\left(\boldsymbol{\mu}_1^{(l)} - \boldsymbol{\mu}_0^{(l)}\right) \\ \pm \frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\left(\left(\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_0} a_{iu}^{(l)}\right)\boldsymbol{\Delta}_0^{(l)} + \left(\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_1} a_{iu}^{(l)}\right)\boldsymbol{\Delta}_1^{(l)}\right)\Bigg] \tag{28}$$

Similarly, we have:

$$\frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\mathbf{h}_{j,\overline{L}}^{(l+1)} \in \Bigg[\boldsymbol{\mu}_1^{(l)} + \frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_0} a_{ju}^{(l)}\left(\boldsymbol{\mu}_0^{(l)} - \boldsymbol{\mu}_1^{(l)}\right) \\ \pm \frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\left(\left(\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_1} a_{ju}^{(l)}\right)\boldsymbol{\Delta}_1^{(l)} + \left(\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_0} a_{ju}^{(l)}\right)\boldsymbol{\Delta}_0^{(l)}\right)\Bigg] \tag{29}$$

Building on this, we have:

$$\begin{aligned}
&\frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\mathbf{h}_i^{(l+1)} - \frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\mathbf{h}_j^{(l+1)} \\
\in &\Bigg[\left(\boldsymbol{\mu}_0^{(l)} - \boldsymbol{\mu}_1^{(l)}\right) + \frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\left(\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_1} a_{iu}^{(l)}\right)\left(\boldsymbol{\mu}_1^{(l)} - \boldsymbol{\mu}_0^{(l)}\right) \\
&- \frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\left(\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_0} a_{ju}^{(l)}\right)\left(\boldsymbol{\mu}_0^{(l)} - \boldsymbol{\mu}_1^{(l)}\right) \\
&\pm\Bigg(\frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\left(\left(\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_0} a_{iu}^{(l)}\right)\boldsymbol{\Delta}_0^{(l)} + \left(\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_1} a_{iu}^{(l)}\right)\boldsymbol{\Delta}_1^{(l)}\right) \\
&\quad + \frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\left(\left(\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_1} a_{ju}^{(l)}\right)\boldsymbol{\Delta}_1^{(l)} + \left(\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_0} a_{ju}^{(l)}\right)\boldsymbol{\Delta}_0^{(l)}\right)\Bigg)\Bigg] \\
= &\Bigg[\left(1 - \frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_1} a_{iu}^{(l)} - \frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_0} a_{ju}^{(l)}\right)\left(\boldsymbol{\mu}_0^{(l)} - \boldsymbol{\mu}_1^{(l)}\right) \\
&\pm\Bigg(\frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\left(\left(\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_0} a_{iu}^{(l)}\right)\boldsymbol{\Delta}_0^{(l)} + \left(\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_1} a_{iu}^{(l)}\right)\boldsymbol{\Delta}_1^{(l)}\right) \\
&\quad + \frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\left(\left(\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_1} a_{ju}^{(l)}\right)\boldsymbol{\Delta}_1^{(l)} + \left(\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_0} a_{ju}^{(l)}\right)\boldsymbol{\Delta}_0^{(l)}\right)\Bigg)\Bigg]
\end{aligned} \tag{30}$$

Taking the l_2-norm on both sides and applying the L-Lipschitz property of $\sigma(\cdot)$, we obtain

$$
\begin{aligned}
\delta_0^{(l+1,t)} &= \left\| \frac{1}{N_0} \sum_{v_i \in \mathcal{S}_0} \mathbf{h}_i^{(l+1)} - \frac{1}{N_1} \sum_{v_j \in \mathcal{S}_1} \mathbf{h}_j^{(l+1)} \right\|_2 \\
&\leq L \Bigg(\Bigg\| \left(\frac{1}{N_0} \sum_{v_i \in \mathcal{S}_0} \sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_0} a_{iu}^{(l)} - \frac{1}{N_1} \sum_{v_j \in \mathcal{S}_1} \sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_0} a_{ju}^{(l)} \right) \boldsymbol{\mu}_0^{(l)} \\
&\quad + \left(\frac{1}{N_0} \sum_{v_i \in \mathcal{S}_0} \sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_1} a_{iu}^{(l)} - \frac{1}{N_1} \sum_{v_j \in \mathcal{S}_1} \sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_1} a_{ju}^{(l)} \right) \boldsymbol{\mu}_1^{(l)} \Bigg\|_2 \\
&\quad + \Bigg\| \frac{1}{N_0} \sum_{v_i \in \mathcal{S}_0} \left(\left(\sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_0} a_{iu}^{(l)} \right) \boldsymbol{\Delta}_0^{(l)} + \left(\sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_1} a_{iu}^{(l)} \right) \boldsymbol{\Delta}_1^{(l)} \right) \\
&\quad + \frac{1}{N_1} \sum_{v_j \in \mathcal{S}_1} \left(\left(\sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_1} a_{ju}^{(l)} \right) \boldsymbol{\Delta}_1^{(l)} + \left(\sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_0} a_{ju}^{(l)} \right) \boldsymbol{\Delta}_0^{(l)} \right) \Bigg\|_2 + 2\sqrt{N} \Delta_z \Bigg)
\end{aligned} \tag{31}
$$

By triangle inequality, Equation (31) gives

$$
\begin{aligned}
\delta_0^{(l+1,t)} &\leq L \Bigg(\left| \frac{1}{N_0} \sum_{v_i \in \mathcal{S}_0} \sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_0} a_{iu}^{(l)} - \frac{1}{N_1} \sum_{v_j \in \mathcal{S}_1} \sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_0} a_{ju}^{(l)} \right| \left\| \boldsymbol{\mu}_0^{(l)} \right\|_2 \\
&\quad + \left| \frac{1}{N_0} \sum_{v_i \in \mathcal{S}_0} \sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_1} a_{iu}^{(l)} - \frac{1}{N_1} \sum_{v_j \in \mathcal{S}_1} \sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_1} a_{ju}^{(l)} \right| \left\| \boldsymbol{\mu}_1^{(l)} \right\|_2 \\
&\quad + \Bigg\| \frac{1}{N_0} \sum_{v_i \in \mathcal{S}_0} \left(\left(\sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_0} a_{iu}^{(l)} \right) \boldsymbol{\Delta}_0^{(l)} + \left(\sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \mathcal{S}_1} a_{iu}^{(l)} \right) \boldsymbol{\Delta}_1^{(l)} \right) \\
&\quad + \frac{1}{N_1} \sum_{v_j \in \mathcal{S}_1} \left(\left(\sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_1} a_{ju}^{(l)} \right) \boldsymbol{\Delta}_1^{(l)} + \left(\sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \mathcal{S}_0} a_{ju}^{(l)} \right) \boldsymbol{\Delta}_0^{(l)} \right) \Bigg\|_2 + 2\sqrt{N} \Delta_z \Bigg)
\end{aligned} \tag{32}
$$

Then the group mean representations are also bounded. Since $\|\mathbf{h}_u^{(l)}\|_\infty \leq \delta_h$ for all $v_u \in \mathcal{V}$, each coordinate of the group mean is bounded by δ_h. Thus, we have

$$
\left\| \boldsymbol{\mu}_0^{(l)} \right\|_2 \leq \delta_h \sqrt{d_l}, \qquad \left\| \boldsymbol{\mu}_1^{(l)} \right\|_2 \leq \delta_h \sqrt{d_l} \tag{33}
$$

where d_l is the dimension of the node representation $\mathbf{h}_u^{(l)}$.

Substituting the above inequalities into Equation (32), we have

$$\begin{aligned}\delta_0^{(l+1,t)} \leq L\Bigg(\delta\sqrt{d^{(l+1)}}\Bigg(&\left|\frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_0}a_{iu}^{(l)}-\frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_0}a_{ju}^{(l)}\right|\\&+\left|\frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_1}a_{iu}^{(l)}-\frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_1}a_{ju}^{(l)}\right|\Bigg)\\&+\Bigg\|\frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\left(\left(\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_0}a_{iu}^{(l)}\right)\boldsymbol{\Delta}_0^{(l)}+\left(\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_1}a_{iu}^{(l)}\right)\boldsymbol{\Delta}_1^{(l)}\right)\\&+\frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\left(\left(\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_1}a_{ju}^{(l)}\right)\boldsymbol{\Delta}_1^{(l)}+\left(\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_0}a_{ju}^{(l)}\right)\boldsymbol{\Delta}_0^{(l)}\right)\Bigg\|_2+2\sqrt{N}\Delta_z\Bigg)\end{aligned} \tag{34}$$

where Δ_z denotes the maximal within-group deviation of the aggregated pre-activation representations.

We let $\boldsymbol{\Delta}^{(l)} = max < \boldsymbol{\Delta}_0^{(l)}, \boldsymbol{\Delta}_1^{(l)} >$, then because the GAT attention weights are normalized for each center node, we have

$$\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_0}a_{iu}^{(l)}+\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_1}a_{iu}^{(l)}=1\quad v_i\in\mathcal{S}_0, \tag{35}$$

and

$$\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_1}a_{ju}^{(l)}+\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_0}a_{ju}^{(l)}=1,\quad v_j\in\mathcal{S}_1 \tag{36}$$

Therefore, we have:

$$\begin{aligned}&\Bigg\|\frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\left(\left(\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_0}a_{iu}^{(l)}\right)\boldsymbol{\Delta}^{(l)}+\left(\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_1}a_{iu}^{(l)}\right)\boldsymbol{\Delta}^{(l)}\right)\\&\quad+\frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\left(\left(\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_1}a_{ju}^{(l)}\right)\boldsymbol{\Delta}^{(l)}+\left(\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_0}a_{ju}^{(l)}\right)\boldsymbol{\Delta}^{(l)}\right)\Bigg\|_2=2\left\|\boldsymbol{\Delta}^{(l)}\right\|_2\end{aligned} \tag{37}$$

Thus, Equation (34) can be simplified as:

$$\begin{aligned}\delta_0^{(l+1,t)} \leq L\Bigg(\delta\sqrt{d^{(l+1)}}\Bigg(&\left|\frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_0}a_{iu}^{(l)}-\frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_0}a_{ju}^{(l)}\right|\\&+\left|\frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_1}a_{iu}^{(l)}-\frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_1}a_{ju}^{(l)}\right|\Bigg)+2\left\|\boldsymbol{\Delta}^{(l)}\right\|_2+2\sqrt{N}\Delta_z\Bigg)\end{aligned} \tag{38}$$

which completes the proof. □

Here, $a_{iu}^{(l)}$ denotes the attention weight from node v_u to node v_i, and $\widetilde{\mathcal{N}}(i)$ is the denoised neighborhood of v_i. Let $\mathbf{z}_i^{(l+1,t)}$ denote the aggregated representation before the nonlinear activation, i.e., $\mathbf{h}_i^{(l+1,t)} = \sigma(\mathbf{z}_i^{(l+1,t)})$. We define $\Delta_z = \max(\Delta_z^{(0)}, \Delta_z^{(1)})$, where $\|\mathbf{z}_i^{(l+1,t)} - \bar{\mathbf{z}}_s^{(l+1,t)}\|_\infty \leq \Delta_z^{(s)}$ for

all $v_i \in \mathcal{S}_s$, with $\bar{\mathbf{z}}_s^{(l+1,t)} = \frac{1}{|\mathcal{S}_s|}\sum_{v_i \in \mathcal{S}_s} \mathbf{z}_i^{(l+1,t)}$. The term Δ_z measures the maximal within-group deviation of the pre-activation aggregated representations.

Then, the disparity between the hidden representations after the nonlinear activation can be bounded by the disparity between the aggregated representations before activation. Substituting the upper bound of the aggregated representation disparity in Equation (34), we obtain:

$$\delta_0^{(l+1,t)} \leq L\Bigg(\delta\sqrt{d^{(l+1)}}\Bigg(\left|\frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_0} a_{iu}^{(l)} - \frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_0} a_{ju}^{(l)}\right| + \left|\frac{1}{N_0}\sum_{v_i\in\mathcal{S}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\mathcal{S}_1} a_{iu}^{(l)} - \frac{1}{N_1}\sum_{v_j\in\mathcal{S}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\mathcal{S}_1} a_{ju}^{(l)}\right|\Bigg) + 2\left\|\boldsymbol{\Delta}^{(l)}\right\|_2 + 2\sqrt{N}\Delta_z\Bigg) \tag{39}$$

B Proof of Theorem 4.2

Here, we prove Theorem 4.2. In this paper, the ground-truth sensitive attribute S is unavailable during training. Therefore, the group partition used in the fairness regularizer is constructed by the inferred proxy $\hat{S}$, leading to two proxy groups $\hat{\mathcal{S}}_0$ and $\hat{\mathcal{S}}_1$. The goal of this proof is to quantify how the representation disparity computed on the proxy groups differs from the representation disparity computed on the true sensitive groups.

Let the true sensitive groups be $\mathcal{S}_0$ and $\mathcal{S}_1$, with group priors as follow:

$$\boldsymbol{p} = \begin{bmatrix}\mathbb{P}(S=0)\\ \mathbb{P}(S=1)\end{bmatrix} \tag{40}$$

and let the proxy group priors be

$$\hat{\boldsymbol{p}} = \begin{bmatrix}\mathbb{P}(\hat{S}=0)\\ \mathbb{P}(\hat{S}=1)\end{bmatrix} \tag{41}$$

Hence, we can define the proxy transition matrix as:

$$\boldsymbol{T} = \begin{bmatrix}\mathbb{P}(\hat{S}=0 \mid S=0) & \mathbb{P}(\hat{S}=1 \mid S=0)\\ \mathbb{P}(\hat{S}=0 \mid S=1) & \mathbb{P}(\hat{S}=1 \mid S=1)\end{bmatrix} = \begin{bmatrix}1-e_0 & e_0\\ e_1 & 1-e_1\end{bmatrix} \tag{42}$$

where $e_0 = \mathbb{P}(\hat{S}=1 \mid S=0)$ and $e_1 = \mathbb{P}(\hat{S}=0 \mid S=1)$ denote the proxy inference error rates. Building on this, the proxy group prior satisfies as follow:

$$\hat{\boldsymbol{p}} = \boldsymbol{T}^\top \boldsymbol{p} \tag{43}$$

Let $\boldsymbol{\mu}_0^{(l+1,t)}$ and $\boldsymbol{\mu}_1^{(l+1,t)}$ denote the mean node representations of the two true sensitive groups after the l-th denoising layer at diffusion step t:

$$\boldsymbol{\mu}_0^{(l+1,t)} = \mathbb{E}\left[\mathbf{h}^{(l+1,t)} \mid S=0\right], \qquad \boldsymbol{\mu}_1^{(l+1,t)} = \mathbb{E}\left[\mathbf{h}^{(l+1,t)} \mid S=1\right] \tag{44}$$

Similarly, let $\hat{\boldsymbol{\mu}}_0^{(l+1,t)}$ and $\hat{\boldsymbol{\mu}}_1^{(l+1,t)}$ denote the mean node representations of the two proxy groups:

$$\hat{\boldsymbol{\mu}}_0^{(l+1,t)} = \mathbb{E}\left[\mathbf{h}^{(l+1,t)} \mid \hat{S}=0\right], \qquad \hat{\boldsymbol{\mu}}_1^{(l+1,t)} = \mathbb{E}\left[\mathbf{h}^{(l+1,t)} \mid \hat{S}=1\right] \tag{45}$$

For compactness, collect the true and proxy group means into two matrices:

$$\boldsymbol{U} = \begin{bmatrix} \left(\boldsymbol{\mu}_0^{(l+1,t)}\right)^\top \\ \left(\boldsymbol{\mu}_1^{(l+1,t)}\right)^\top \end{bmatrix}, \qquad \hat{\boldsymbol{U}} = \begin{bmatrix} \left(\hat{\boldsymbol{\mu}}_0^{(l+1,t)}\right)^\top \\ \left(\hat{\boldsymbol{\mu}}_1^{(l+1,t)}\right)^\top \end{bmatrix} \tag{46}$$

We use the standard proxy-noise assumption that, once the true sensitive group is fixed, the proxy inference error does not further depend on the node representation:

$$\mathbb{P}(\hat{S} = \hat{s} \mid \mathbf{h}^{(l+1,t)}, S = s) = \mathbb{P}(\hat{S} = \hat{s} \mid S = s) \tag{47}$$

Under this assumption, the proxy group mean is a normalized mixture of the true group means. Specifically, for $\hat{s} \in \{0, 1\}$, we have:

$$\mathbb{P}(\hat{S} = \hat{s})\, \mathbb{E}\left[\mathbf{h}^{(l+1,t)} \mid \hat{S} = \hat{s}\right] = \sum_{s \in \{0,1\}} \mathbb{P}(\hat{S} = \hat{s} \mid S = s)\mathbb{P}(S = s)\mathbb{E}\left[\mathbf{h}^{(l+1,t)} \mid S = s\right] \tag{48}$$

Therefore, in matrix form, we have:

$$\mathrm{diag}(\hat{\boldsymbol{p}})\hat{\boldsymbol{U}} = \boldsymbol{T}^\top \mathrm{diag}(\boldsymbol{p})\boldsymbol{U} \tag{49}$$

Hence, we are able to rewrite it as:

$$\hat{\boldsymbol{U}} = \boldsymbol{M}\boldsymbol{U}, \qquad \boldsymbol{M} = \mathrm{diag}(\hat{\boldsymbol{p}})^{-1}\boldsymbol{T}^\top \mathrm{diag}(\boldsymbol{p}) \tag{50}$$

By Bayes' rule, the entries of $\boldsymbol{M}$ are the posterior probabilities of the true sensitive group given the proxy group, *i.e.*, $\boldsymbol{M}_{\hat{s},s} = \mathbb{P}(S = s \mid \hat{S} = \hat{s})$. Writing $\eta_0 = \mathbb{P}(S = 1 \mid \hat{S} = 0) = e_1 p_1 / \hat{p}_0$ and $\eta_1 = \mathbb{P}(S = 0 \mid \hat{S} = 1) = e_0 p_0 / \hat{p}_1$ for the impurity of the two proxy groups, we have:

$$\boldsymbol{M} = \begin{bmatrix} 1 - \eta_0 & \eta_0 \\ \eta_1 & 1 - \eta_1 \end{bmatrix} \tag{51}$$

Therefore, the two proxy group means are mixtures of the true group means:

$$\hat{\boldsymbol{\mu}}_0^{(l+1,t)} = (1 - \eta_0)\, \boldsymbol{\mu}_0^{(l+1,t)} + \eta_0\, \boldsymbol{\mu}_1^{(l+1,t)}, \qquad \hat{\boldsymbol{\mu}}_1^{(l+1,t)} = \eta_1\, \boldsymbol{\mu}_0^{(l+1,t)} + (1 - \eta_1)\, \boldsymbol{\mu}_1^{(l+1,t)} \tag{52}$$

Subtracting the second equation from the first gives:

$$\hat{\boldsymbol{\mu}}_0^{(l+1,t)} - \hat{\boldsymbol{\mu}}_1^{(l+1,t)} = (1 - \eta_0 - \eta_1)\left(\boldsymbol{\mu}_0^{(l+1,t)} - \boldsymbol{\mu}_1^{(l+1,t)}\right) \tag{53}$$

Taking the ℓ_2-norm on both sides, we have:

$$\delta_{\hat{S}}^{(l+1,t)} = |1 - \eta_0 - \eta_1|\ \delta_S^{(l+1,t)} \tag{54}$$

Since $\eta_0 + \eta_1 < 1$, we obtain:

$$\delta_S^{(l+1,t)} = \frac{\delta_{\hat{S}}^{(l+1,t)}}{1 - \eta_0 - \eta_1} \tag{55}$$

It remains to bound the proxy-based representation disparity $\delta_{\hat{S}}^{(l+1,t)}$. Since $\hat{\mathcal{S}}_0$ and $\hat{\mathcal{S}}_1$ form the group partition used during training, Theorem 4.1 can be applied to this proxy-based partition. Therefore,

$$\delta_{\hat{S}}^{(l+1,t)} \leq \widehat{\mathcal{B}}_{\mathrm{mp}}^{(l,t)} \tag{56}$$

where $\widehat{\mathcal{B}}_{\mathrm{mp}}^{(l,t)}$ is the right-hand side of Theorem 4.1 computed on the proxy groups $\hat{\mathcal{S}}_0$ and $\hat{\mathcal{S}}_1$. Substituting it into Equation (55), the representation disparity between the true sensitive groups can be upper bounded by:

$$\delta_S^{(l+1,t)} \leq \frac{L}{1-\eta_0-\eta_1}\left(\delta\sqrt{d^{(l+1)}}\left(\left|\frac{1}{\hat{N}_0}\sum_{v_i\in\hat{\mathcal{S}}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\hat{\mathcal{S}}_0} a_{iu}^{(l)} - \frac{1}{\hat{N}_1}\sum_{v_j\in\hat{\mathcal{S}}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\hat{\mathcal{S}}_0} a_{ju}^{(l)}\right|\right.\right.$$
$$\left.\left. + \left|\frac{1}{\hat{N}_0}\sum_{v_i\in\hat{\mathcal{S}}_0}\sum_{v_u\in\widetilde{\mathcal{N}}(i)\cap\hat{\mathcal{S}}_1} a_{iu}^{(l)} - \frac{1}{\hat{N}_1}\sum_{v_j\in\hat{\mathcal{S}}_1}\sum_{v_u\in\widetilde{\mathcal{N}}(j)\cap\hat{\mathcal{S}}_1} a_{ju}^{(l)}\right|\right) + 2\left\|\boldsymbol{\Delta}^{(l)}\right\|_2 + 2\sqrt{N}\Delta_z\right) \tag{57}$$

where $\hat{N}_0 = |\hat{\mathcal{S}}_0|$ and $\hat{N}_1 = |\hat{\mathcal{S}}_1|$, which completes the proof. □

C Proof of Theorem 4.3

Here, we prove Theorem 4.3. The goal is to translate the representation disparity bound in Theorem 4.2 into an upper bound on the demographic parity violation of the generated graph.

We start from the statistical parity metric for edge generation. In our reverse generative process, the decoder predicts the edge type between two nodes through a mixture of multinomial distributions. Let $g_\theta^{(t)}(v_i, v_j)$ denote the probability that an edge is generated between v_i and v_j at denoising step t. Equivalently, if 0 denotes the no-edge type, then:

$$g_\theta^{(t)}(v_i, v_j) = 1 - \mathbb{P}_\theta\left(A_{ij}^{(t)} = 0 \mid G'_{t+1}\right) \tag{58}$$

For a specific edge type r, the same proof applies by setting $g_{\theta,r}^{(t)}(v_i, v_j) = \mathbb{P}_\theta\left(A_{ij}^{(t)} = r \mid G'_{t+1}\right)$. The demographic parity violation of the generated graph is defined as:

$$\Delta_{\mathrm{DP}}^{(l+1,t)} = \left|\mathbb{E}\left[g_\theta^{(t)}(v_i, v_j) \mid S_i = S_j\right] - \mathbb{E}\left[g_\theta^{(t)}(v_i, v_j) \mid S_i \neq S_j\right]\right| \tag{59}$$

Since the generated adjacency entry is sampled according to $g_\theta^{(t)}(v_i, v_j)$, bounding Equation (59) also bounds the expected statistical parity violation of the generated graph.

Hence, we define the mean node representations under the true sensitive partition as

$$\boldsymbol{\mu}_{0,S}^{(l+1,t)} = \mathbb{E}\left[\mathbf{h}^{(l+1,t)} \mid S = 0\right], \qquad \boldsymbol{\mu}_{1,S}^{(l+1,t)} = \mathbb{E}\left[\mathbf{h}^{(l+1,t)} \mid S = 1\right] \tag{60}$$

The true-group representation disparity is then:

$$\delta_S^{(l+1,t)} = \left\|\boldsymbol{\mu}_{0,S}^{(l+1,t)} - \boldsymbol{\mu}_{1,S}^{(l+1,t)}\right\|_2 \tag{61}$$

For each sensitive group $s \in \{0, 1\}$, define the within-group representation variation as:

$$r_s^{(l+1,t)} = \mathbb{E}\left[\left\|\mathbf{h}^{(l+1,t)} - \boldsymbol{\mu}_{s,S}^{(l+1,t)}\right\|_2 \mid S = s\right] \tag{62}$$

For simplicity, define the group-pair expected edge-generation probability as:

$$G_{ab} = \mathbb{E}\left[g_\theta^{(t)}(v_i, v_j) \mid S_i = a, S_j = b\right], \qquad a, b \in \{0, 1\} \tag{63}$$

For an undirected generated graph, the edge decoder is symmetric with respect to the two endpoints, and therefore $G_{01} = G_{10}$. Let $\pi_0 = \mathbb{P}(S = 0)$ and $\pi_1 = \mathbb{P}(S = 1)$. Then we have:

$$\mathbb{E}\left[g_\theta^{(t)}(v_i, v_j) \mid S_i = S_j\right] = \omega_0 G_{00} + \omega_1 G_{11} \tag{64}$$

where $\omega_0 = \frac{\pi_0^2}{\pi_0^2+\pi_1^2}$, $\omega_1 = \frac{\pi_1^2}{\pi_0^2+\pi_1^2}$, and $\omega_0 + \omega_1 = 1$.

Similarly, we also have:

$$\mathbb{E}\left[g_\theta^{(t)}(v_i, v_j) \mid S_i \neq S_j\right] = G_{01} \tag{65}$$

Substituting the above equations into Equation (59), we obtain:

$$\begin{aligned}\Delta_{\mathrm{DP}}^{(l+1,t)} &= |\omega_0 G_{00} + \omega_1 G_{11} - G_{01}| \\ &\leq \omega_0 |G_{00} - G_{01}| + \omega_1 |G_{11} - G_{01}|\end{aligned} \tag{66}$$

We now bound the two terms in Equation (66). Given that the edge decoder $g_\theta^{(t)}$ is L_{edge}-Lipschitz with respect to the node representations, namely,

$$\left|g_\theta^{(t)}(\mathbf{h}_i, \mathbf{h}_j) - g_\theta^{(t)}(\mathbf{h}'_i, \mathbf{h}'_j)\right| \leq L_{\mathrm{edge}} \left(\|\mathbf{h}_i - \mathbf{h}'_i\|_2 + \left\|\mathbf{h}_j - \mathbf{h}'_j\right\|_2\right) \tag{67}$$

Let $\mathbf{h}_0$ and $\mathbf{h}'_0$ be two independent node representations sampled from group $S = 0$, and let $\mathbf{h}_1$ be sampled from group $S = 1$. Then

$$\begin{aligned}|G_{00} - G_{01}| &= \left|\mathbb{E}\left[g_\theta^{(t)}(\mathbf{h}_0, \mathbf{h}'_0) - g_\theta^{(t)}(\mathbf{h}_0, \mathbf{h}_1)\right]\right| \\ &\leq \mathbb{E}\left[\left|g_\theta^{(t)}(\mathbf{h}_0, \mathbf{h}'_0) - g_\theta^{(t)}(\mathbf{h}_0, \mathbf{h}_1)\right|\right] \\ &\leq L_{\mathrm{edge}} \mathbb{E}\left[\|\mathbf{h}'_0 - \mathbf{h}_1\|_2\right] \\ &\leq L_{\mathrm{edge}} \mathbb{E}\left[\left\|\mathbf{h}'_0 - \boldsymbol{\mu}_{0,S}^{(l+1,t)}\right\|_2 + \left\|\boldsymbol{\mu}_{0,S}^{(l+1,t)} - \boldsymbol{\mu}_{1,S}^{(l+1,t)}\right\|_2 + \left\|\mathbf{h}_1 - \boldsymbol{\mu}_{1,S}^{(l+1,t)}\right\|_2\right] \\ &= L_{\mathrm{edge}} \left(r_0^{(l+1,t)} + \delta_S^{(l+1,t)} + r_1^{(l+1,t)}\right)\end{aligned} \tag{68}$$

Similarly, let $\mathbf{h}_1$ and $\mathbf{h}'_1$ be two independent node representations sampled from group $S = 1$, and let $\mathbf{h}_0$ be sampled from group $S = 0$. By the same argument, we have:

$$|G_{11} - G_{01}| \leq L_{\mathrm{edge}} \left(r_0^{(l+1,t)} + \delta_S^{(l+1,t)} + r_1^{(l+1,t)}\right) \tag{69}$$

Substituting Equation (68) and Equation (69) into Equation (66), and using $\omega_0 + \omega_1 = 1$, gives

$$\Delta_{\mathrm{DP}}^{(l+1,t)} \leq L_{\mathrm{edge}} \left(\delta_S^{(l+1,t)} + r_0^{(l+1,t)} + r_1^{(l+1,t)}\right) \tag{70}$$

Next, by Theorem 4.2, when the denoising model is trained using the inferred sensitive proxy $\hat{S}$, the true-group representation disparity satisfies:

$$\begin{aligned}\delta_S^{(l+1,t)} \leq \frac{L}{1-\eta_0-\eta_1} \Bigg(\delta \sqrt{d^{(l+1)}} \Bigg(&\left| \frac{1}{\hat{N}_0} \sum_{v_i \in \hat{\mathcal{S}}_0} \sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \hat{\mathcal{S}}_0} a_{iu}^{(l)} - \frac{1}{\hat{N}_1} \sum_{v_j \in \hat{\mathcal{S}}_1} \sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \hat{\mathcal{S}}_0} a_{ju}^{(l)} \right| \\ &+ \left| \frac{1}{\hat{N}_0} \sum_{v_i \in \hat{\mathcal{S}}_0} \sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \hat{\mathcal{S}}_1} a_{iu}^{(l)} - \frac{1}{\hat{N}_1} \sum_{v_j \in \hat{\mathcal{S}}_1} \sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \hat{\mathcal{S}}_1} a_{ju}^{(l)} \right| \Bigg) + 2 \left\| \boldsymbol{\Delta}^{(l)} \right\|_2 + 2\sqrt{N} \Delta_z \Bigg)\end{aligned} \tag{71}$$

where $\eta_0 = \mathbb{P}(S = 1 \mid \hat{S} = 0)$ and $\eta_1 = \mathbb{P}(S = 0 \mid \hat{S} = 1)$ are the impurity of the two proxy groups defined in Theorem 4.2, and L is the Lipschitz constant of the activation function in the denoising network. Substituting Equation (71) into Equation (70), we obtain:

$$\Delta_{\mathrm{DP}}^{(l+1,t)} \leq L_{\mathrm{edge}} \Bigg(\frac{L}{1-\eta_0-\eta_1} \Bigg(\delta \sqrt{d^{(l+1)}} \Bigg(\left| \frac{1}{\hat{N}_0} \sum_{v_i \in \hat{\mathcal{S}}_0} \sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \hat{\mathcal{S}}_0} a_{iu}^{(l)} - \frac{1}{\hat{N}_1} \sum_{v_j \in \hat{\mathcal{S}}_1} \sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \hat{\mathcal{S}}_0} a_{ju}^{(l)} \right| \\ + \left| \frac{1}{\hat{N}_0} \sum_{v_i \in \hat{\mathcal{S}}_0} \sum_{v_u \in \widetilde{\mathcal{N}}(i) \cap \hat{\mathcal{S}}_1} a_{iu}^{(l)} - \frac{1}{\hat{N}_1} \sum_{v_j \in \hat{\mathcal{S}}_1} \sum_{v_u \in \widetilde{\mathcal{N}}(j) \cap \hat{\mathcal{S}}_1} a_{ju}^{(l)} \right| \Bigg) + 2 \left\| \mathbf{\Delta}^{(l)} \right\|_2 + 2\sqrt{N} \Delta_z \Bigg) \\ + r_0^{(l+1,t)} + r_1^{(l+1,t)} \Bigg) \tag{72}$$

which completes the proof. □

D Experimental Setup

D.1 Datasets

In our experiments, four real-world datasets, namely German [33], Cora [34], Pokec-z [35], and Photo [36], are used. We provide a brief description of these datasets as follows:

- **German** [33]: This dataset is derived from the German Credit dataset, where nodes represent credit applicants with demographic and financial attributes. Node features include information such as account status, credit history, employment status, and personal characteristics. The prediction label indicates the credit risk or creditworthiness of each applicant.
- **Cora** [34]: This dataset is a widely used citation network, where nodes represent academic papers and edges denote citation relationships between papers. Node features are bag-of-words representations extracted from paper text. Cora is commonly used to evaluate graph learning models on citation-based relational structures.
- **Pokec-z** [35]: This dataset is sampled from Pokec, a large-scale social networking platform in Slovakia. Nodes represent users, and edges denote friendship relations between users. Each node is associated with profile attributes such as gender, age, region, and personal interests. The dataset is widely used in fair graph learning because social connections and user attributes can encode demographic or regional disparities.
- **Photo** [36]: This dataset is a subset of the Amazon co-purchase network, where nodes represent photo-related products and edges indicate frequent co-purchase relationships. Node features are bag-of-words representations extracted from product reviews.

For each dataset, we use 80% of the graphs for training and 20% for testing. To simulate realistic scenarios where demographic information is unavailable during model development, we mask all demographic attributes in the training and validation sets.

D.2 Baselines

We compare our method with several representative graph generation baselines covering variational, score-based, diffusion-based, autoregressive, spectral, and flow-matching paradigms. **GRAPHARM** [24] is an autoregressive diffusion-based model that sequentially masks nodes and edges, employing a learned node-ordering strategy for efficient and accurate discrete graph generation. **GSDM** [37] is a spectral diffusion-based graph generation framework that uses low-rank stochastic differential equations over the graph spectrum space to improve graph topology generation while reducing computational cost. **GraphVAE** [38] formulates graph generation as a variational autoencoding problem and directly decodes a probabilistic fully connected graph with a predefined maximum size, avoiding explicit graph sequence linearization. **GDSS** [39] is a score-based graph generative model that jointly models node features and adjacency matrices through a system of

stochastic differential equations, enabling the generation process to capture dependencies between nodes and edges. **DiGress** [40] is a discrete denoising diffusion model for graphs with categorical node and edge attributes, where a graph transformer learns to reverse a discrete noise process that edits edges and attribute categories. **CatFlow** [41] develops a variational flow matching framework for categorical graph data, providing an efficient flow-based generative model with strong performance on graph generation tasks. These baselines provide a comprehensive comparison across the graph generation task.

D.3 Implementation Details

For all experiments, we use the following implementation settings. Models are trained for up to 500 epochs on German, Cora, and Photo, and up to 200 epochs on Pokec-z, with early stopping based on the validation loss. We use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, a learning rate of 1×10^{-3}, and a weight decay of 1×10^{-5}. Training is stopped early if the validation loss does not improve for 30 consecutive epochs, and the model parameters from the best validation epoch are restored. Hyperparameter configurations are selected based on validation performance using the same search space across all methods to ensure a fair comparison. During the evaluation, we generate the same number of synthetic subgraphs as the test set and report the average results over five random seeds. All experiments are conducted on a single NVIDIA A100 GPU. We implement all models using PyTorch.

E Additional Experimental Results

E.1 Additional Comparison Study

We provide additional evaluation of the graph generative performance on the Pokec-z and Photo datasets, as Table 3. The results show a consistent trend with those observed on German and Cora. Fairness-agnostic graph generative baselines can achieve competitive generation quality, indicating that they are effective at capturing the overall structural and attribute distributions of the original graphs. However, they also preserve noticeable fairness gaps, since their generation objectives do not distinguish task-relevant graph patterns from sensitive-related dependencies. In contrast, CertiFG achieves substantially better fairness performance on both Pokec-z and Photo while maintaining comparable generation quality. This suggests that the proposed proxy-aware fairness regularization can effectively suppress sensitive-related information during graph generation without disrupting the structural patterns needed to generate realistic graphs. These results demonstrate that CertiFG generalizes well across datasets with different graph properties and consistently produces synthetic graphs that are both high-quality and fairness-preserving.

Table 3: Graph generation results on Pokec-z and Photo datasets. Lower values indicate better performance.

Dataset	Method	DD	Clus	NFea	Fair-DD	Fair-Clus	Fair-NFea
Pokec-z	GRAPHARM	0.348	0.150	0.253	0.078	0.067	0.038
	GSDM	0.331	0.142	0.231	0.073	0.060	0.057
	GraphVAE	0.378	0.174	0.291	0.062	0.049	0.039
	GDSS	0.386	0.181	0.308	0.068	0.054	0.047
	DIGRESS	0.372	0.173	0.285	0.077	0.061	0.051
	CatFlow	0.347	0.158	0.248	0.072	0.068	0.058
	CertiFG	0.355	0.168	0.272	0.043	0.029	0.025
Photo	GRAPHARM	0.317	0.235	0.327	0.063	0.084	0.068
	GSDM	0.293	0.229	0.314	0.055	0.079	0.063
	GraphVAE	0.354	0.258	0.349	0.046	0.065	0.054
	GDSS	0.375	0.291	0.362	0.041	0.061	0.057
	DIGRESS	0.342	0.284	0.351	0.054	0.071	0.061
	CatFlow	0.319	0.227	0.334	0.059	0.076	0.068
	CertiFG	0.320	0.257	0.345	0.023	0.040	0.041

E.2 Additional Downstream Task Results

We provide additional downstream task results on the German, Pokec-z, and Photo datasets to further validate that the synthetic graphs generated by CertiFG remain useful for downstream tasks while preserving fairness. Following the same setup as in the main text, we use each method to generate a synthetic version of the training graph, and then train a standard GNN classifier on the generated graph for the downstream task. The results are reported in Table 4. Consistent with the observations on the Cora dataset, classifiers trained on graphs generated by CertiFG consistently achieve better fairness across all three datasets, while maintaining utility comparable to those trained on graphs from fairness-agnostic baselines. These results confirm that CertiFG produces synthetic graphs that not only achieve better fairness on their own structure, but also lead to fairer downstream classifiers across diverse graph datasets, further supporting that the fairness improvements achieved at the generation stage successfully transfer to downstream tasks.

Table 4: Downstream task results on German, Pokec-z and Photo datasets.

Dataset	Method	AUC (%)	F1-score (%)	Δ_{DP} (%)	Δ_{EO} (%)
German	Original-GCN	95.14 ± 1.32	90.31 ± 1.18	28.35 ± 1.77	5.25 ± 0.79
	GRAPHARM-GCN	88.86 ± 1.43	86.29 ± 1.51	30.69 ± 1.83	5.37 ± 1.07
	GSDM-GCN	86.50 ± 1.95	83.99 ± 1.69	31.71 ± 1.56	6.18 ± 1.19
	GraphVAE-GCN	83.69 ± 2.08	77.75 ± 1.47	33.38 ± 1.91	8.62 ± 1.36
	GDSS-GCN	86.96 ± 1.75	82.23 ± 1.84	37.46 ± 2.46	9.41 ± 1.61
	DIGRESS-GCN	87.87 ± 1.23	82.19 ± 1.66	29.85 ± 1.74	7.86 ± 1.32
	CatFlow-GCN	85.55 ± 1.64	79.28 ± 1.58	26.98 ± 1.48	5.94 ± 1.16
	CertiFG-GCN	84.82 ± 1.46	82.01 ± 1.29	21.64 ± 0.98	4.07 ± 0.74
Pokec-z	Original-GCN	88.28 ± 0.86	81.23 ± 0.93	35.15 ± 1.12	14.14 ± 0.81
	GRAPHARM-GCN	85.71 ± 0.64	79.75 ± 0.70	32.22 ± 0.95	13.72 ± 0.67
	GSDM-GCN	83.77 ± 0.89	77.94 ± 0.92	41.43 ± 1.26	21.54 ± 0.94
	GraphVAE-GCN	80.76 ± 1.23	74.82 ± 1.11	43.23 ± 1.48	24.87 ± 1.06
	GDSS-GCN	82.21 ± 0.97	77.02 ± 1.07	44.29 ± 1.51	22.45 ± 0.91
	DIGRESS-GCN	81.75 ± 0.81	76.75 ± 0.88	41.07 ± 1.31	23.16 ± 1.14
	CatFlow-GCN	78.32 ± 0.76	73.06 ± 0.95	33.75 ± 1.07	14.83 ± 0.84
	CertiFG-GCN	83.13 ± 1.11	74.03 ± 0.96	29.91 ± 0.72	11.95 ± 0.68
Photo	Original-GCN	93.15 ± 1.91	90.72 ± 2.64	55.91 ± 2.10	9.11 ± 1.01
	GRAPHARM-GCN	87.08 ± 1.69	85.45 ± 1.72	49.85 ± 1.86	13.64 ± 1.14
	GSDM-GCN	89.45 ± 1.57	82.15 ± 2.12	51.64 ± 1.95	15.82 ± 1.27
	GraphVAE-GCN	85.41 ± 2.05	78.32 ± 1.81	54.02 ± 3.05	17.39 ± 1.93
	GDSS-GCN	88.62 ± 1.73	80.11 ± 1.88	53.46 ± 2.17	15.73 ± 1.48
	DIGRESS-GCN	87.92 ± 1.44	83.71 ± 1.60	60.29 ± 1.38	18.21 ± 1.18
	CatFlow-GCN	84.15 ± 1.27	79.42 ± 1.39	48.72 ± 1.86	12.98 ± 1.24
	CertiFG-GCN	85.75 ± 1.68	84.36 ± 1.57	44.27 ± 1.89	7.81 ± 1.09

E.3 Additional Ablation Study

We provide additional ablation study results for the Pokec-z and Photo datasets to further validate the importance of each component in CertiFG. Following the same setup as in the main text, we compare CertiFG with two ablated variants, CertiFG-WI and CertiFG-WF, where the proxy identification loss $\mathcal{L}_I$ and the calibrated fairness regularizer $\mathcal{L}_F$ are removed, respectively. The results are reported in Figure 2. Consistent with the observations on the German and Cora datasets, removing either component leads to a worst fairness result. Specifically, removing $\mathcal{L}_F$ causes the largest fairness drop, since the generator no longer has any guide to suppress the bias-inducing dependencies in the training data, while removing $\mathcal{L}_I$ also degrades fairness because the proxy prediction head becomes more easily misled by the systematic errors in the noisy proxy labels. These results further confirm that both $\mathcal{L}_I$ and $\mathcal{L}_F$ are necessary.

E.4 Visualization of Synthetic Graphs

To provide a more intuitive understanding of how CertiFG mitigates bias during graph generation, we visualize the synthetic graphs produced by CertiFG and a representative fairness-agnostic baseline *e.g.,* GRAPHARM, on the German dataset, where red edges denote inter-group connections and green edges denote intra-group connections. As shown in Figure 3, the graph generated by GRAPHARM exhibits a clear concentration of green intra-group edges, revealing a strong segregation between the two sensitive groups, where members of the same group are heavily connected with one another while

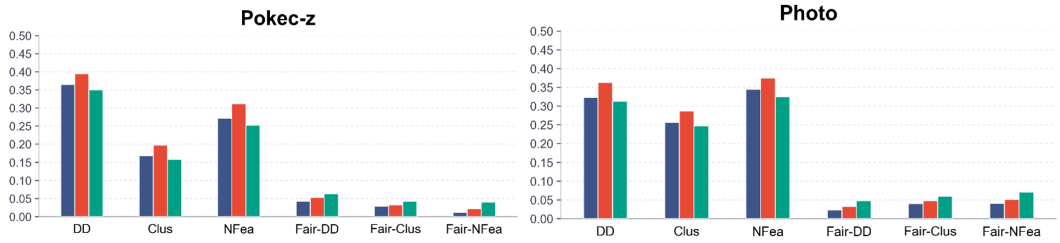


Figure 2: Ablation study results for CertiFG, CertiFG-WI, and CertiFG-WF in Pokec-z and Photo datasets.

remaining largely isolated from the other group, mirroring and even amplifying the demographic separation present in the training data. In contrast, the graph generated by CertiFG (right) shows a noticeably larger proportion of red inter-group edges that span across the two sensitive groups, with intra-group and inter-group edges interleaved more uniformly across the graph. This visual difference confirms that CertiFG effectively breaks the segregation between sensitive groups in the generated topology, which aligns with our quantitative fairness improvements and validates that the proposed proxy-error-aware fairness regularizer indeed produces fairer graph structures.

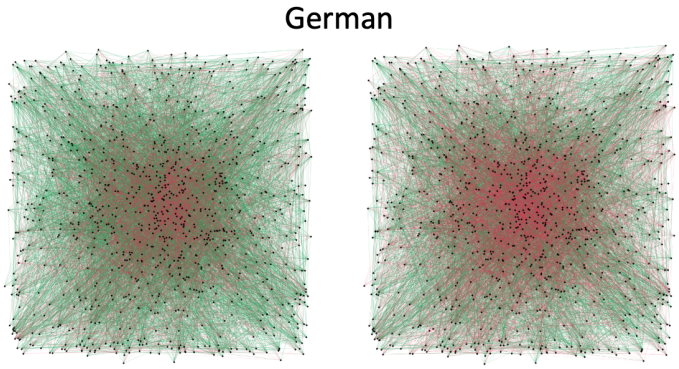


Figure 3: Comparison of intra-edge (green) and inter-edge (red) distributions in synthetic German graphs generated by GRAPHARM (left) and CertiFG (right).

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: **[TODO]**

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: **[TODO]**

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: **[TODO]**

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: **[TODO]**

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: **[TODO]**

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: **[TODO]**

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: **[TODO]**

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: **[TODO]**

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.