Face to Fairness: Mitigating Influence Bias for Fair Graph Anomaly Detection in Dynamic Graphs

Anonymous Author(s)

Abstract

Graph anomaly detection plays a vital role in applications such as fraud detection, cybersecurity, and social network analysis. Despite recent advances in graph neural networks (GNNs) for dynamic graphs, existing methods often suffer from an overlooked issue: *influence bias* caused by nodes with high structural centrality. Such nodes dominate message passing and feature aggregation, leading to biased anomaly detection where anomalies around high-influence regions are overemphasized, while anomalies on low-centrality nodes remain under-detected. Severe class imbalance further exacerbates this challenge by limiting the model's ability to learn from scarce, structurally marginal anomalies. In this work, we propose FairGAD , a fairness-aware framework for dynamic graph anomaly detection that mitigates influence bias from a causal representation learning perspective. Our model integrates a *Graph Expert Mixture Encoder* that combines complementary structural experts and temporal modeling to produce robust node representations, a *Representation-Space Anomaly Generator* that learns the distribution of anomalous embeddings and synthesizes diverse tail anomalies to alleviate class imbalance, and an *Influence Debiasing Classifier* that employs adversarial training with a gradient reversal layer to learn influence-invariant representations by attenuating the spurious path from node influence to anomaly prediction. Extensive experiments on multiple real-world dynamic graph datasets demonstrate that FairGAD not only improves anomaly detection performance but also substantially reduces influence bias, yielding more equitable detection across nodes with varying influence levels. For instance, on the UCI and ALPHA datasets, our approach achieves relative F1-score improvements of 0.89% and 1.53% over strong baselines while narrowing performance gaps between high- and low-centrality nodes.

Keywords

Graph Anomaly Detection, Influence Bias, Graph Neural Networks, Dynamic Graphs, Wasserstein GAN, Adversarial Training, Temporal Modeling

ACM Reference Format:

Anonymous Author(s). 2026. Face to Fairness: Mitigating Influence Bias for Fair Graph Anomaly Detection in Dynamic Graphshttps://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

In many real-world networks, a small fraction of nodes exhibit high structural influence, characterized by their ability to disproportionately affect information propagation through the graph. For instance, such nodes include accounts with high degree or PageRank in financial transaction graphs, influential users with many followers in social or information networks, and central routers in communication infrastructures. These high-centrality nodes often play critical roles by directly or indirectly affecting a large portion of the graph. When such influential nodes become anomalous, their abnormal behaviors can propagate through the network and impact many otherwise normal nodes. Figure 1 shows this influence propagation effect: anomalous features originating from a structurally central node are amplified through message passing, whereas anomalies residing in peripheral regions remain weakly expressed. This asymmetric propagation leads to a systematic influence bias, where anomalies associated with high-influence nodes are overemphasized, while anomalies on low-centrality nodes are consistently under-detected.

Graph anomaly detection (GAD) aims to identify nodes, edges, or subgraphs whose behaviors deviate significantly from those of the majority, and thus often correspond to abnormal or malicious activities [1]. Recent advances in graph neural networks (GNNs) have enabled powerful representation learning for nodes on (dynamic) graphs, which has substantially improved the effectiveness of anomaly detection [2, 3]. Most GNN-based GAD methods adopt a message-passing paradigm: multi-layer GNNs aggregate information from local neighborhoods to obtain node embeddings, and then apply a classifier or scoring function over these embeddings to detect anomalies. However, under this paradigm, high-centrality nodes tend to have a disproportionate impact on representation learning. Because they connect to many neighbors and lie on numerous paths, their features are repeatedly aggregated and propagated across multiple layers, and thus implicitly injected into the embeddings of a large number of nodes. When such high-centrality nodes are anomalous, their abnormal features can be propagated and mixed into the representations of their neighbors, potentially contaminating many normal nodes. In contrast, anomalies residing in structurally marginal regions with low centrality are more likely to be overshadowed by dominant patterns, making them harder to detect. This leads to a systematic *influence bias*: anomalies associated with influential nodes are overemphasized, while anomalies on low-centrality nodes are under-detected [4–7].

From a causal perspective, node influence, quantified by structural attributes such as degree, PageRank, or other centrality measures, acts as a confounding factor in GNN-based anomaly detection. On the one hand, influence shapes node representations: high-centrality nodes participate in more message-passing and aggregation steps, so their information is more likely to dominate neighborhood embeddings, as qualitatively depicted in Figure 1. On the other hand, in many real-world datasets, supervision signals or pseudo-labels are more easily collected or inferred around structurally central regions, further entangling influence with anomaly labels. As a result, the prediction head can inadvertently exploit

Conference'17, Washington, DC, USA
© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

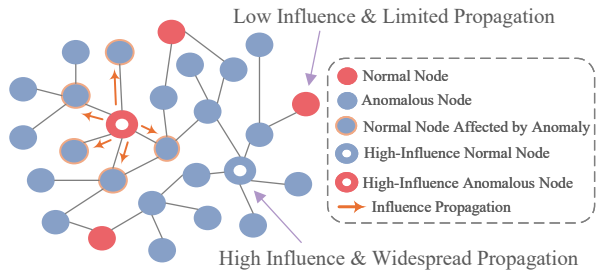


Figure 1: Illustration of influence bias in dynamic graph anomaly detection. A structurally central anomalous node propagates its influence through message passing, contaminating the representations of many neighbors, while anomalies in peripheral regions remain weakly represented. This creates a systematic bias that overemphasizes central areas and under-detects anomalies on low-influence nodes, reflecting a spurious pathway from node influence to anomaly prediction that our framework aims to mitigate.

influence-related cues instead of focusing on truly abnormal behaviors, forming a spurious causal pathway "Influence → Representation → Prediction". Along this path, normal neighbors of anomalous high-centrality nodes may be misclassified due to contaminated embeddings, while truly anomalous low-centrality nodes may be systematically underweighted. Therefore, merely increasing the complexity of GNN architectures is insufficient for fair and robust anomaly detection; it is crucial to explicitly disentangle and correct the bias introduced by node influence at the representation level.

In this work, we propose FairGAD , a fairness-aware framework for dynamic graph anomaly detection that mitigates influence bias from a causal disentanglement perspective. Our model comprises three key components. **First**, we design a Graph Expert Mixture Encoder that combines multiple GNN backbones (GCN, GAT, and GraphSAGE) with a learnable attention-based fusion mechanism and temporal encoding. This module can be viewed as a mixture-of-experts over structural views: different backbones capture complementary inductive biases, and the fusion module adaptively selects the most appropriate structural view for each node and time step, providing robust representations and reducing over-reliance on any single neighborhood aggregation scheme. **Second**, motivated by the long-tail nature of anomalies in real graphs, especially for low-centrality and structurally marginal nodes, we introduce a Representation-Space Anomaly Generator that learns the distribution of anomalous representations in the feature space and synthesizes diverse anomalous features. Rather than simply oversampling existing anomalies, this module enriches the tail of the anomaly distribution and effectively expands the representation space of rare and low-centrality anomalies, alleviating severe class imbalance and enhancing the model's sensitivity to hard-to-detect anomalies. **Third**, to causally disentangle influence from anomaly prediction, we design an influence debiasing classifier equipped with a Gradient Reversal Layer (GRL) [8]. Besides the main anomaly classifier, we attach an auxiliary influence predictor that attempts to recover influence-related attributes from the learned representations, while the GRL forces the feature extractor to make such recovery as difficult as possible. This adversarial training yields influence-invariant representations that retain predictive power for anomaly detection but suppress the spurious path from influence to prediction, thereby deconfounding influence bias in the learned decision rule.

Our contributions are summarized as follows:

- We identify and formalize the *influence bias* problem in GNN-based graph anomaly detection from a causal perspective, showing how structural centrality can act as a confounding factor that leads to unfair detection: anomalies near influential nodes are overemphasized, while low-centrality anomalies are under-detected.
- We propose a Representation-Space Anomaly Generator that learns the anomaly distribution in the representation space and synthesizes diverse anomalous features, particularly enriching the tail of rare and low-centrality anomalies to alleviate class imbalance.
- We design an influence debiasing classifier with adversarial training and a Gradient Reversal Layer to learn influence-invariant node representations, effectively mitigating influence bias while maintaining strong anomaly detection performance across nodes with varying influence levels.

2 Related Work

2.1 Fairness in Graph Representation Learning

The exploration of fairness in graph representation learning has garnered significant attention, with various approaches addressing the intrinsic biases present in graph-structured data. Spinelli et al. [5] introduce FairDrop, a biased edge dropout technique designed to enhance fairness by counteracting homophily effects, which often lead to biased representations. By selectively dropping edges that reinforce majority groups, FairDrop effectively reduces group-level disparity in node classification. However, this method focuses on static graphs and group fairness, and it does not explicitly consider temporal dynamics or anomaly detection tasks. Similarly, Khajehnejad et al. [6] propose CrossWalk, a straightforward yet effective method that enhances fairness across multiple graph algorithms, including influence maximization and node classification. CrossWalk reweights random walks to avoid overexposing majority groups, thereby improving fairness for any random-walk-based embedding technique. While CrossWalk is versatile and model-agnostic, it again targets group fairness in generic graph learning rather than node-level anomaly detection or dynamic settings. Köse et al. [9] focus on fairness-aware contrastive learning, employing adaptive feature masking and edge deletion to address fairness issues through graph augmentation. Their work emphasizes the importance of designing augmentations aligned with specific fairness notions, highlighting the role of data manipulation in reducing bias. Fan et al. [10] develop FairGAE, a fair graph auto-encoder that leverages tailored GCN layers to produce unbiased embeddings under sensitive-attribute constraints. These methods demonstrate that both data augmentation and architectural modifications can improve fairness, but they typically assume access to sensitive attributes and concentrate on static graph tasks.

Further advancing the field, Köse et al. [11, 12] introduce adaptive augmentation schemes based on fairness notions, with the latter

providing a theoretical analysis of bias sources in GNNs. These studies underscore the significance of understanding bias origins and employing adaptive strategies to mitigate them effectively. Laclau et al. [13] provide a comprehensive survey, discussing the unique challenges of ensuring fairness in graph mining, such as the non-IID nature of graph data and the difficulty of balancing accuracy with fairness. Their review highlights the complexity of defining and measuring fairness within relational data contexts.

Addressing the limitations of existing methods, Liu et al. [14] propose Graph-Fairness Mixture of Experts (G-Fame), a plug-and-play framework that enhances the learnability of unbiased, distinguishable representations, especially when attribute information is limited. Guo et al. [15] focus on distance correlation minimization under graph Laplacian regularization to learn fair representations that reduce group separation and intra-group clustering. These works are particularly relevant to our setting in that they emphasize representation-level debiasing and mixture-of-experts designs. From a causal perspective, many of these approaches can be interpreted as learning representations that are invariant to sensitive or group-related factors, partially deconfounding their effect on downstream predictions. Nonetheless, they primarily consider group or attribute-based fairness on static graphs, whereas our work focuses on *influence bias* at the node level in dynamic graph anomaly detection, without relying on sensitive attribute supervision.

2.2 Adversarial and Debiasing Methods in GNNs

The landscape of adversarial and debiasing methods in Graph Neural Networks (GNNs) has also garnered significant research attention, addressing both robustness against malicious perturbations and fairness concerns. Li et al. [16] introduce Spectral Adversarial Training (SAT), a method designed to enhance GNN robustness by regularizing spectral properties of the graph, thereby addressing the inherent difficulties posed by graph structure discreteness and interconnected relationships. Similarly, Pei et al. [17] propose HC-Ref, a hierarchical constrained refinement framework that improves adversarial robustness by separately strengthening the anti-perturbation capabilities of GNNs and their downstream classifiers. These methods show that carefully designed adversarial training can substantially improve robustness, but they mainly focus on worst-case perturbations rather than systematic biases induced by node influence.

In addition to these approaches, the integration of multi-view and ensemble learning strategies has been explored to bolster GNN resilience. Wu et al. [18] develop a multiview-ensemble-learning-based robust GCN (MV-RGCN), which constructs auxiliary views based on high-order structural and feature similarities from perturbed graphs, thereby enhancing robustness against adversarial attacks. Complementing this, Zhang et al. [19] leverage large language models (LLMs) to improve GNN robustness by inferring and correcting malicious graph structures, proposing the LLM4RGNN framework that distills GPT-based inference capabilities into local models for edge identification and recovery. These multi-view and LLM-enhanced approaches highlight the benefits of combining diverse structural views, but they are not explicitly designed to address fairness or influence-related bias.

Adversarial attacks targeting fairness in GNNs have also been scrutinized. Zhang et al. [20] investigate how adversarial perturbations can compromise the fairness of GNNs, despite their improvements in algorithmic fairness, and introduce G-FairAttack, a framework designed to attack fairness-aware GNNs with minimal impact on prediction utility. This line of work reveals that fairness-aware models remain vulnerable to carefully crafted perturbations, underscoring the dual challenge of defending against both structural and fairness-related adversarial threats. To counter such threats, Shen et al. [21] propose the Adversarial Contrastive Graph Masked AutoEncoder (ACGMAE), which jointly defends against structure and feature attacks through contrastive learning. Furthermore, Fang et al. [22] address structural bias in adversarial defenses, particularly focusing on low-degree nodes (tail nodes). They propose a mitigation strategy involving hetero-homo augmented graphs, kNN-based augmentation, and multi-view node-wise attention modules to reduce structural bias and improve defense effectiveness on less-connected nodes.

Collectively, these works illustrate that adversarial training, multi-view modeling, and contrastive learning are powerful tools for improving robustness and mitigating certain forms of bias in GNNs. However, most of them focus on adversarial perturbations crafted by an attacker or on group-level fairness, rather than on the *intrinsic influence bias* induced by high-centrality nodes in dynamic graphs. Moreover, they seldom consider generative augmentation of anomalies or causal disentanglement of influence-related factors. In contrast, our work combines generative modeling and adversarial debiasing to explicitly deconfound node influence from anomaly prediction in a dynamic graph setting, aiming to achieve both robust and fair anomaly detection across nodes with varying influence levels. In particular, our influence debiasing classifier with a GRL can be viewed as a causal representation learning module that adversarially removes influence-related information from node embeddings while preserving anomaly-relevant signals.

3 Framework

In this section, we present the overall architecture of our proposed model FairGAD (FairGAD) for fairness-aware dynamic graph anomaly detection. As shown in Figure 2, FairGAD consists of three tightly coupled components: (1) a **Graph Expert Mixture Encoder** that aggregates temporal and structural representations from multiple GNN backbones, (2) a **Representation-Space Anomaly Generator** that synthesizes realistic anomalous features to alleviate class imbalance, and (3) an **Influence Debiasing Classifier** that employs adversarial learning to remove influence-related information from the learned representations.

Formally, we denote a dynamic graph sequence by $\{G_t\}_{t=1}^{T}$, where each snapshot $G_t = (V_t, E_t, \mathbf{X}_t)$ consists of a node set V_t with $|V_t| = N_t$ nodes, an edge set E_t, and a node feature matrix $\mathbf{X}_t \in \mathbb{R}^{N_t \times d}$, where d is the input feature dimension. Let $\mathbf{A}_t \in \mathbb{R}^{N_t \times N_t}$ be the adjacency matrix of G_t. Our objective is to learn a function

$$f : \{G_t\}_{t=1}^{T} \rightarrow \{\mathbf{A}_t\}_{t=1}^{T}, \quad \mathbf{A}_t \in \mathbb{R}^{N_t}, \tag{1}$$

where $\mathbf{A}_t$ contains the anomaly scores for all nodes at time t. For brevity, we write $N = \max_t N_t$ when referring to the maximum number of nodes across time.

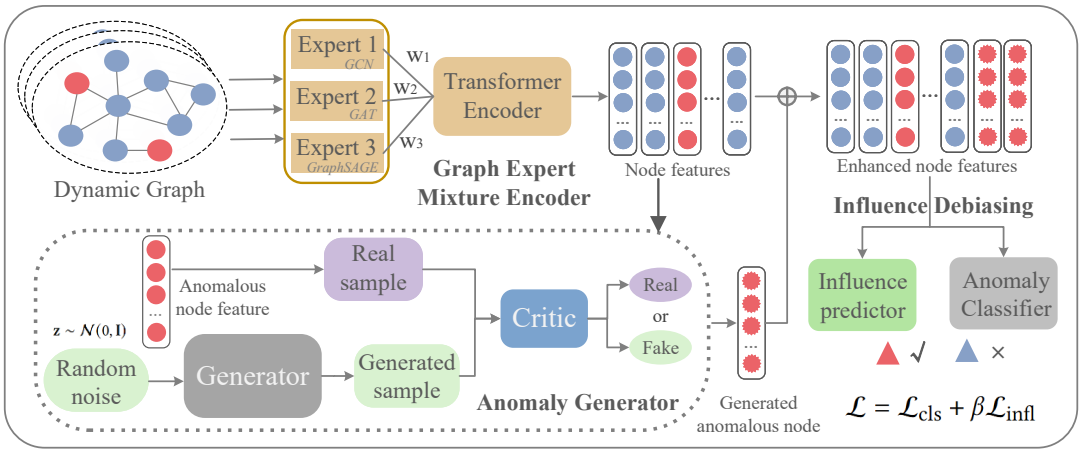


Figure 2: Architecture of the proposed FairGAD model, comprising a Graph Expert Mixture Encoder, a Representation-Space Anomaly Generator, and an Influence Debiasing Classifier for fairness-aware dynamic graph anomaly node detection.

OurMODEL follows a three-stage design:

(1) **Feature Extraction (Graph Expert Mixture Encoder):** Given $\{G_t\}_{t=1}^{T}$, we construct multi-branch structural experts at each time step and fuse their outputs via learnable expert weights and temporal Transformer encoding to obtain node representations $\mathbf{H}_t \in \mathbb{R}^{N_t \times d_h}$.

(2) **Anomaly Generation (Representation-Space Anomaly Generator):** In the learned representation space, a Wasserstein GAN-based module learns the distribution of anomalous embeddings and generates synthetic anomaly features $\tilde{\mathbf{h}}_g \in \mathbb{R}^{d_h}$ to enrich the tail of rare anomalies.

(3) **Influence Debiasing (Influence Debiasing Classifier):** A classifier equipped with a Gradient Reversal Layer (GRL) jointly predicts anomaly labels and suppresses influence-related information, yielding influence-invariant node representations and anomaly scores $\hat{\mathbf{A}}_t$.

In the following, we detail each component with explicit notation.

3.1 Graph Expert Mixture Encoder

The Graph Expert Mixture Encoder aims to learn expressive, temporally-aware node representations from the dynamic graph sequence by combining several complementary "structural experts". Each expert branch focuses on a different inductive bias: one emphasizes neighborhood smoothing, another captures attention-based neighbor importance, and a third models sampled neighborhood aggregation. In our implementation, these experts are instantiated using standard graph convolutional operators [23–25], but the framework is generic and can accommodate other choices. For each time step t, we denote:

- $\mathbf{X}_t \in \mathbb{R}^{N_t \times d}$: input node feature matrix,
- $\mathbf{A}_t \in \mathbb{R}^{N_t \times N_t}$: adjacency matrix,
- $\mathbf{H}_t^{(1)}, \mathbf{H}_t^{(2)}, \mathbf{H}_t^{(3)} \in \mathbb{R}^{N_t \times d'}$: branch-specific hidden representations for the three structural experts,
- $\mathbf{H}_t^{\text{fuse}} \in \mathbb{R}^{N_t \times d_h}$: fused representation at time t.

Adaptive Graph Convolution. For each snapshot G_t, we process $\mathbf{X}_t$ through three expert graph convolution branches that instantiate the above inductive biases.

(a) Expert 1 (smoothing-based aggregation) [23]. Given $\mathbf{A}_t$, we first construct the normalized adjacency matrix:

$$\tilde{\mathbf{A}}_t = \mathbf{A}_t + \mathbf{I}, \quad \tilde{\mathbf{D}}_t = \operatorname{diag}\left(\sum_j \tilde{A}_{t,ij}\right), \quad \hat{\mathbf{A}}_t = \tilde{\mathbf{D}}_t^{-\frac{1}{2}} \tilde{\mathbf{A}}_t \tilde{\mathbf{D}}_t^{-\frac{1}{2}}, \tag{2}$$

where $\mathbf{I}$ is the $N_t \times N_t$ identity matrix and $\tilde{\mathbf{D}}_t$ is the degree matrix of $\tilde{\mathbf{A}}_t$. The first expert branch updates node features as

$$\mathbf{H}_t^{(1)} = \sigma\left(\hat{\mathbf{A}}_t \mathbf{X}_t \mathbf{W}_1\right), \tag{3}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d \times d'}$ is a trainable weight matrix and $\sigma(\cdot)$ is a nonlinear activation function.

(b) Expert 2 (attention-based aggregation) [24]. This branch assigns dynamic attention weights to neighbors. For node i and neighbor $j \in \mathcal{N}(i)$, the unnormalized attention score is

$$e_{ij} = \text{LeakyReLU}\left(\mathbf{a}^\top \left[\mathbf{W}_2 \mathbf{x}_i \,\|\, \mathbf{W}_2 \mathbf{x}_j\right]\right), \tag{4}$$

where $\mathbf{W}_2 \in \mathbb{R}^{d \times d'}$ is a shared linear transformation, $\mathbf{a} \in \mathbb{R}^{2d'}$ is a learnable attention vector, and $[\cdot \,\|\, \cdot]$ denotes vector concatenation. The normalized attention coefficient is

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik})}, \tag{5}$$

and the updated features are aggregated as

$$\mathbf{H}_t^{(2)} = \sigma\left(\mathbf{A}_t^{\alpha} \mathbf{X}_t \mathbf{W}_2\right), \tag{6}$$

where $\mathbf{A}_t^{\alpha}$ is the attention-weighted adjacency matrix.

(c) Expert 3 (sampled neighborhood aggregation) [25]. This branch aggregates neighborhood features with a mean aggregator:

$$\mathbf{h}_{\mathcal{N}(i)} = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathbf{x}_j, \tag{7}$$

and updates each node by concatenating its own feature with aggregated neighbor features:

$$\mathbf{H}_t^{(3)} = \sigma\left([\mathbf{X}_t \,\|\, \mathbf{A}_t\mathbf{X}_t]\,\mathbf{W}_3\right), \tag{8}$$

where $\mathbf{W}_3 \in \mathbb{R}^{2d\times d'}$ is a projection matrix, and $(\mathbf{A}_t\mathbf{X}_t)_i = \mathbf{h}_{\mathcal{N}(i)}$.

Let $\boldsymbol{m} = [m_1, m_2, m_3]$ be a trainable parameter vector. We compute branch weights $\mathbf{w} = [w_1, w_2, w_3]$ via a softmax over $\boldsymbol{m}$ and form the weighted concatenation:

$$\mathbf{H}_t^{(w)} = \left[w_1\mathbf{H}_t^{(1)} \,\|\, w_2\mathbf{H}_t^{(2)} \,\|\, w_3\mathbf{H}_t^{(3)}\right], \tag{9}$$

which is further projected to obtain the fused representation

$$\mathbf{H}_t^{\text{fuse}} = \mathbf{W}_f\mathbf{H}_t^{(w)} + \mathbf{b}_f, \tag{10}$$

where $\mathbf{W}_f$ and $\mathbf{b}_f$ are trainable parameters and $\mathbf{H}_t^{\text{fuse}} \in \mathbb{R}^{N_t\times d_h}$.

Temporal Encoding. To explicitly model temporal dynamics in the evolving graph, we associate each discrete time step t with a learnable global temporal embedding $\mathbf{e}_t \in \mathbb{R}^d$. This embedding is shared across all nodes at the same time step and is added to the input node features to inject global temporal context:

$$\tilde{\mathbf{X}}_t = \mathbf{X}_t + \mathbf{e}_t, \tag{11}$$

After applying the graph expert mixture encoder, we obtain the fused node representations $\mathbf{H}_t^{\text{fuse}}$ at each time step. These representations are stacked along the temporal dimension to form a three-dimensional tensor:

$$\mathcal{H} = \left[\mathbf{H}_1^{\text{fuse}}; \mathbf{H}_2^{\text{fuse}}; \ldots; \mathbf{H}_T^{\text{fuse}}\right] \in \mathbb{R}^{T\times N_{\max}\times d_h}, \tag{12}$$

where T denotes the number of time steps and $N_{\max}$ is the maximum number of nodes across all snapshots. To capture long-range temporal dependencies, we apply a Transformer encoder along the temporal dimension, enabling each node to attend to its representations across different time steps:

$$\mathbf{H} = \text{TransformerEncoder}(\mathcal{H}) \in \mathbb{R}^{T\times N_{\max}\times d_h}. \tag{13}$$

These node representations are then used as input for downstream anomaly generation and debiasing modules.

3.2 Representation-Space Anomaly Generator

The Representation-Space Anomaly Generator operates in the latent feature space to mitigate class imbalance by enriching rare anomaly patterns. Let $\mathbb{P}_r$ denote the empirical distribution of anomalous embeddings (extracted from $\mathbf{H}$), and $\mathbb{P}_g$ the distribution induced by the generator.

Generator and Critic. The generator G maps a latent variable $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{z} \in \mathbb{R}^{d_z}$, to a synthetic anomalous representation $\tilde{\mathbf{h}}_g \in \mathbb{R}^{d_h}$:

$$G(\mathbf{z}) = \phi_3\left(\mathbf{W}_g^{(3)}\phi_2\left(\mathbf{W}_g^{(2)}\phi_1\left(\mathbf{W}_g^{(1)}\mathbf{z}\right)\right)\right), \tag{14}$$

where $\{\mathbf{W}_g^{(i)}\}$ and $\{\phi_i(\cdot)\}$ denote the learnable weight matrices and nonlinear activation functions, respectively. The critic (discriminator) D assigns a scalar score to a feature vector $\mathbf{h} \in \mathbb{R}^{d_h}$:

$$D(\mathbf{h}) = \mathbf{w}_d^\top\phi_2\left(\mathbf{W}_d^{(2)}\phi_1\left(\mathbf{W}_d^{(1)}\mathbf{h}\right)\right), \tag{15}$$

where $\{\mathbf{W}_d^{(1)}, \mathbf{W}_d^{(2)}, \mathbf{w}_d\}$ are learnable parameters.

Objective Function. Following the Wasserstein GAN (WGAN) framework, the critic is trained to maximize an empirical approximation of the Wasserstein-1 distance between $\mathbb{P}_r$ and $\mathbb{P}_g$:

$$\mathcal{L}_D = \mathbb{E}_{\mathbf{h}\sim\mathbb{P}_r}[D(\mathbf{h})] - \mathbb{E}_{\tilde{\mathbf{h}}\sim\mathbb{P}_g}[D(\tilde{\mathbf{h}})], \tag{16}$$

while the generator minimizes

$$\mathcal{L}_G = -\mathbb{E}_{\tilde{\mathbf{h}}\sim\mathbb{P}_g}[D(\tilde{\mathbf{h}})]. \tag{17}$$

To enforce the Lipschitz continuity constraint on the critic, we apply a weight clipping to its parameters: $\theta_D \leftarrow \text{clip}(\theta_D, -c, c)$, where $c > 0$ is a small constant.

Anomaly Feature Synthesis. After convergence, the generator produces synthetic anomalous features via

$$\tilde{\mathbf{h}}_g = G(\mathbf{z}), \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I}). \tag{18}$$

These generated features are combined with real anomalous and normal embeddings to construct an augmented training set for the debiasing classifier, effectively enriching the representation space around rare and low-centrality anomalies.

3.3 Influence Debiasing Classifier

From a causal perspective, we consider node influence S, node features X, latent representations H, and anomaly labels Y in a structural causal model where both S and X affect H, and H determines Y. In addition, S also shapes the graph structure and the supervision process, which further entangles S with Y. Our goal is to deconfound anomaly prediction by learning representations H that preserve the causal effect of behavior-related features X on Y, while suppressing the spurious pathway $S \to H \to Y$ induced by node influence.

To this end, we design an **Influence Debiasing Classifier** that utilizes adversarial training to remove influence bias from learned node representations. Let $\mathcal{D}_{\text{real}}$ denote the set of real node embeddings (normal and limited anomalies), and $\mathcal{D}_{\text{gen}}$ the set of generated anomalous embeddings from Eq. (18). The combined training dataset is

$$\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{gen}}. \tag{19}$$

For each embedding $\mathbf{h}_i \in \mathbb{R}^{d_h}$ with anomaly label $y_i \in \{0, 1\}$ and influence attribute vector $\mathbf{s}_i \in \mathbb{R}^m$ (e.g., degree, betweenness, PageRank), the debiasing module consists of two branches:

- **Main classifier** $C : \mathbb{R}^{d_h} \to [0, 1]^2$ predicting anomaly labels:
$$\hat{y}_i = C(\mathbf{h}_i), \tag{20}$$
where $\hat{y}_i$ is the predicted probability over normal/anomalous classes.
- **Influence predictor** $I : \mathbb{R}^{d_h} \to \mathbb{R}^m$ estimating influence attributes via a Gradient Reversal Layer (GRL):
$$\hat{\mathbf{s}}_i = I\big(\text{GRL}(\mathbf{h}_i, \lambda)\big), \tag{21}$$
where $\lambda > 0$ controls the strength of gradient reversal.

The GRL [8] behaves as an identity function in the forward pass:

$$\text{GRL}(\mathbf{h}_i, \lambda) = \mathbf{h}_i, \tag{22}$$

but reverses and scales gradients during backpropagation:

$$\frac{\partial\,\text{GRL}}{\partial\mathbf{h}_i} = -\lambda\mathbf{I}, \tag{23}$$

forcing the shared encoder to produce embeddings from which $I(\cdot)$ cannot easily recover S.

The overall training objective combines a classification loss and an influence prediction loss:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \beta \mathcal{L}_{\text{infl}}, \tag{24}$$

where $\beta > 0$ balances detection accuracy and debiasing. Specifically,

$$\mathcal{L}_{\text{cls}} = \frac{1}{N} \sum_{i=1}^{N} \ell_{\text{CE}}\big(C(\mathbf{h}_i), y_i\big), \tag{25}$$

$$\mathcal{L}_{\text{infl}} = \frac{1}{N} \sum_{i=1}^{N} \big\| I\big(\text{GRL}(\mathbf{h}_i, \lambda)\big) - \mathbf{s}_i \big\|_2^2. \tag{26}$$

where ℓ_{CE} is the cross-entropy loss. Intuitively, $\mathcal{L}_{\text{cls}}$ encourages embeddings $\mathbf{h}_i$ to retain information predictive of anomalies Y, while the adversarial influence loss $\mathcal{L}_{\text{infl}}$ (via GRL) discourages $\mathbf{h}_i$ from encoding influence-related information S. This yields approximate influence-invariant representations that preserve anomaly-relevant signals while suppressing spurious influence effects, aligning with the goal of learning a causal, deconfounded representation for anomaly prediction.

Training Workflow. The training procedure alternates between feature encoding, anomaly generation, and influence debiasing:

(1) **Encode dynamic graphs:** For each snapshot G_t, apply the Graph Expert Mixture Encoder (Eqs. 3–13) to obtain node embeddings $\mathbf{H}_t^{\text{fuse}}$ and temporal representations $\mathbf{H}$.
(2) **Train generator and critic:** Sample real anomalous embeddings from $\mathbb{P}_r$, generate synthetic anomalies via Eq. (18), and update the critic and generator using Eqs. (16)–(17).
(3) **Debias classifier:** Construct $\mathcal{D}_{\text{train}}$ as in Eq. (19), compute anomaly predictions and influence predictions via Eqs. (20)–(21), and update the encoder, classifier, and influence predictor jointly by minimizing Eq. (24).

This iterative process continues until convergence, after which the generator and influence predictor are discarded at inference time, and the anomaly scores $\hat{\mathbf{A}}_t$ are obtained solely from the encoder and main classifier.

3.4 Time Complexity Analysis

We analyze the time complexity of the main components of FairGAD with respect to the number of nodes N, edges E, feature dimension d, and time steps T.

1. Graph Expert Mixture Encoder. For each graph snapshot at time t, the feature extractor processes node features through three graph neural network branches: GCN, GAT, and GraphSAGE. **GCN:** The core operation is a sparse matrix multiplication $\hat{\mathbf{A}}_t \mathbf{X}_t \mathbf{W}$, which costs roughly $O(|E|d + Ndd')$, where d' is the output feature dimension. **GAT:** Attention computation involves calculating scores for each edge and then aggregating, leading to a complexity on the same order as GCN up to a constant factor for attention heads. **GraphSAGE:** Aggregation requires summing neighbor features and applying linear projections, yielding a similar complexity to GCN. Since the three branches run in parallel, the overall encoder complexity per time step is dominated by the slowest branch and can be summarized as $O(|E|d + Ndd')$. Adding temporal embedding and a Transformer encoder over T time steps introduces an additional cost on the order of $O(TNd^2 + T^2Nd)$ due to self-attention over time.

2. Representation-Space Anomaly Generator. The generator and critic are implemented as multi-layer perceptrons with fixed input and output dimensions (d_z and d_x) independent of graph size. Each forward/backward pass therefore costs roughly $O(d_z d_x)$, and the total cost scales linearly with the number of iterations and batch size, which is generally negligible compared to graph operations.

3. Influence Debiasing Classifier. The classifier and influence predictor are also MLPs operating on embeddings of dimension d_h. For N nodes, the complexity per training epoch is approximately $O(Nd_h d_c)$, where d_c is the classifier hidden dimension. The Gradient Reversal Layer does not add computational cost beyond standard backpropagation.

Overall Complexity. Per time step, the most computationally expensive part is the multi-branch GNN feature extraction and temporal Transformer encoding. Over T time steps, the total complexity can be summarized as roughly $O\big(T(|E|d + Ndd') + TNd^2 + T^2Nd\big)$, which is manageable for sparse graphs where $|E| \approx O(N)$ and moderate T. The generator and classifier modules scale linearly with the number of samples and embedding dimensions and contribute only a small fraction of the total cost.

4 Experiments

4.1 Experimental Settings

This section outlines the experimental setup used to assess the proposed framework, covering datasets, baselines, evaluation metrics, and implementation details.

(a) Datasets. We evaluate our approach using five real-world dynamic graph datasets, each presenting distinct structural and temporal patterns that pose challenges for anomaly detection: **EU-Core1, EU-Core3** [26]: Temporal collaboration graphs constructed from European research communities and institutions. **OTC, ALPHA** [27]: Cryptocurrency transaction networks (Bitcoin), capturing rapidly changing financial exchanges. **UCI** [28]: A composite benchmark derived from multiple dynamic graphs in the UCI repository. All datasets are highly imbalanced, with anomalous nodes accounting for less than 5%, reflecting practical settings.

(b) Baselines. We benchmark against nine recent approaches for dynamic graph anomaly detection:

- **TGAT** [29]: Learns temporal-aware node embeddings via graph attention.
- **DOMINANT** [30]: Detects anomalies by jointly reconstructing graph structure and attributes.
- **DONE** [31]: Combines GCN and temporal autoencoders for sequential graph modeling.
- **CONAD** [32]: Utilizes contrastive learning with temporal perturbations.
- **AnomalyDAE** [33]: Employs deep autoencoders for dynamic anomaly detection.
- **SAD** [34]: A semi-supervised model that leverages limited labels to enhance detection.

Table 1: Overall performance comparison across five datasets.

Dataset	Metric	TGAT	DOMINANT	DONE	CONAD	AnomalyDAE	SAD	GeneralDyG	MAMF	ABNet	FairGAD
UCI	ROC-AUC	0.9491	0.5233	0.4993	0.5225	0.4912	0.9223	0.7514	0.9735	0.9798	**0.9859**
	Precision	0.4971	0.1894	0.2000	0.1895	0.1895	0.4971	0.4459	0.9760	0.9410	**0.9835**
	F1_score	0.4985	0.0805	0.0850	0.0805	0.0805	0.4985	0.4583	0.9741	0.9392	**0.9830**
	AUPR	0.1258	0.1912	0.1848	0.1913	0.1835	0.1746	0.4158	0.9725	0.9755	**0.9910**
EU-Core1	ROC-AUC	0.4475	0.5282	0.5138	0.5255	0.4993	0.5361	0.5207	0.9573	0.9499	**0.9614**
	Precision	0.4972	0.2745	0.3137	0.2745	0.2745	0.4944	0.0606	0.9516	0.9104	**0.9600**
	F1_score	0.4986	0.1162	0.1328	0.1162	0.1162	0.4972	0.1017	0.9455	0.9092	**0.9551**
	AUPR	0.0057	0.2214	0.2247	0.2199	0.2170	0.0170	0.0767	0.9258	0.9535	**0.9766**
EU-Core3	ROC-AUC	0.5558	0.5856	0.5535	0.5868	0.5835	0.9080	0.2789	0.9403	**0.9498**	0.9419
	Precision	0.4944	0.2157	0.1569	0.5157	0.1961	0.4986	0.0116	0.9017	0.8822	**0.9416**
	F1_score	0.4972	0.0995	0.0724	0.0995	0.0905	0.4972	0.0164	0.8748	0.8748	**0.9343**
	AUPR	0.098	0.2123	0.2019	0.2128	0.2063	0.1555	0.0478	0.8381	0.8381	**0.9531**
OTC	ROC-AUC	0.6730	0.6227	0.6347	0.6250	0.3941	0.7173	0.7123	0.9415	0.9103	**0.9493**
	Precision	0.4560	0.6644	0.7153	0.6746	0.0915	0.4560	0.2242	0.9399	0.8912	**0.9431**
	F1_score	0.4770	0.2518	0.2710	0.2556	0.0347	0.477	0.3491	0.9284	0.8738	**0.9340**
	AUPR	0.1845	0.4129	0.435	0.4137	0.1694	0.1997	0.3155	0.9074	0.9419	**0.9670**
ALPHA	ROC-AUC	0.7542	0.6823	0.6843	0.6816	0.4962	0.7574	0.7528	0.9355	0.9389	**0.9548**
	Precision	0.4682	0.6579	0.6474	0.6737	0.0947	0.4682	0.2941	0.9400	0.8978	**0.9502**
	F1_score	0.4836	0.3289	0.3237	0.3368	0.0474	0.4836	0.3704	0.9293	0.8948	**0.9446**
	AUPR	0.1427	0.4152	0.4006	0.4107	0.1436	0.1414	0.4420	0.9320	0.9295	**0.9724**

- **GeneralDyG** [35]: Enhances anomaly detection on dynamic graphs by sampling temporal ego-graphs and sequentially extracting structural and temporal features.
- **MAMF** [36]: A multitask approach fusing diverse structural features.
- **ABNet** [37]: Designed to address anomaly scarcity in dynamic graphs by synthesizing diverse anomaly-like features for robust detection.

(c) Evaluation Metrics. We adopt four commonly used metrics: **ROC-AUC** (area under the ROC curve measuring anomaly differentiation ability), **Precision** (proportion of true anomalies among predicted anomalies), **F1-score** (harmonic mean of precision and recall), and **AUPR** (area under the precision-recall curve for imbalanced settings).

(d) Implementation Details. Each dataset is split into five temporal slices. Synthetic anomalies are injected in the first four slices, which are pooled and split into a training set and a test set for metric reporting. The fifth slice is kept clean for early stopping and light model selection. Our method injects anomalies at the node feature level, requiring a different train/validation/test split than conventional baselines: the validation slice is clean to optimize low false positives, and the test set contains concept-validation anomalies. The embedding dimension k is set to 128, the batch size to 100, and the initial learning rate to 0.00001. The hyperparameter β is fixed at 1. Experiments are performed on an NVIDIA RTX 4090 GPU, and the results are averaged over 20 independent runs for statistical reliability.

4.2 Performance Comparison

Table 1 reports the comparison results across five datasets. Overall, FairGAD consistently outperforms all baselines on most metrics, demonstrating its effectiveness in modeling temporal dynamics, handling class imbalance, and mitigating structural bias.

On the EU-Core1 and EU-Core3 datasets, our model surpasses the strongest baseline in both ROC-AUC and F1-score, showing the advantage of the *Graph Expert Mixture Encoder*, which fuses representations from GCN, GAT, and GraphSAGE to enhance robustness and reduce dependency on individual node influence. For transaction networks (OTC, ALPHA), FairGAD achieves the highest AUPR, benefiting from the *Representation-Space Anomaly Generator* that alleviates severe imbalance by synthesizing realistic anomalous samples. On the heterogeneous UCI dataset, our model attains near-perfect scores, validating the effectiveness of the *influence debiasing classifier* in ensuring fairness by disentangling anomaly cues from centrality-related bias.

These results collectively confirm that the integration of multi-branch fusion, anomaly generation, and influence debiasing enables FairGAD to achieve state-of-the-art performance in dynamic graph anomaly detection.

4.3 Ablation Studys

To further analyze the contribution of each component in FairGAD , we conduct ablation experiments by selectively removing or modifying key modules. Three variants are considered:

- **w/o IDC:** Removes the *Influence Debiasing Classifier*, and directly predicts anomaly scores from fused representations without adversarial debiasing. This variant evaluates the effect of influence removal on fairness and detection performance.
- **w/o GEME:** Replaces the *Graph Expert Mixture Encoder* with a single GCN backbone, removing attention-based fusion and temporal integration. This setup assesses the benefit of heterogeneous GNN fusion in representation quality.

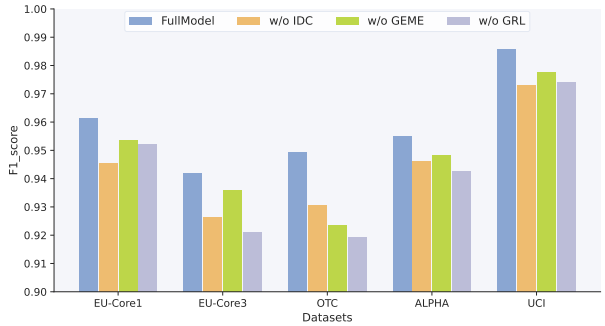


Figure 3: Ablation results for FairGAD. Variants are designed to examine the effect of influence debiasing classifier (IDC), graph expert mixture encoder (GEME), and adversarial learning via the Gradient Reversal Layer (GRL).

- **w/o GRL:** Keeps the influence classifier but disables the *Gradient Reversal Layer (GRL)*, so that the debiasing branch no longer exerts adversarial pressure on the shared encoder. This variant isolates the effect of adversarial learning in mitigating influence bias.

From the results, removing the influence debiasing module (w/o IDC) leads to a significant drop in fairness-related metrics and a moderate decline in overall ROC-AUC, confirming that structural influence indeed biases anomaly prediction. Eliminating the multi-branch extractor (w/o GEME) yields the largest decrease in detection accuracy, highlighting the importance of fusing heterogeneous GNN representations for robust temporal encoding. Finally, the variant w/o GRL shows a smaller but consistent performance degradation, verifying that adversarial gradient reversal is essential to fully disentangle influence-related signals.

Overall, these findings demonstrate that each component, multi-branch fusion, WGAN-based augmentation, and adversarial debiasing, plays a complementary role in achieving both high detection accuracy and fairness.

4.4 Parameter Sensitivity Analysis

We further analyze the sensitivity of FairGAD to the hyperparameter β, which controls the trade-off between the anomaly detection loss and the influence debiasing loss in the total objective. Specifically, β adjusts the relative weight of the fairness constraint introduced by the adversarial branch.

As shown in Figure 4, we vary β from 0.1 to 0.9 with an interval of 0.1, and report the ROC-AUC results across five datasets. The performance remains overall stable, exhibiting only minor fluctuations within this range. This indicates that FairGAD is not overly sensitive to the choice of β, and can maintain consistent performance across a broad range of settings.

These results suggest that the proposed debiasing mechanism is robust, effectively improving fairness and detection performance without requiring precise hyperparameter tuning.

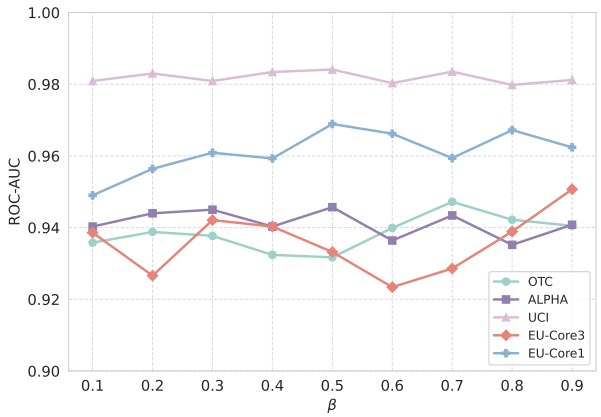


Figure 4: Sensitivity analysis of the hyperparameter β, which balances the anomaly detection loss and the influence debiasing loss.

4.5 Feature Distribution Visualization via t-SNE

To further examine the impact of the proposed influence debiasing mechanism, we visualize the learned node embeddings using three-dimensional t-SNE projections. Figure 5 compares the feature distributions *before* and *after* applying debiasing on two representative datasets (UCI and EU-Core1).

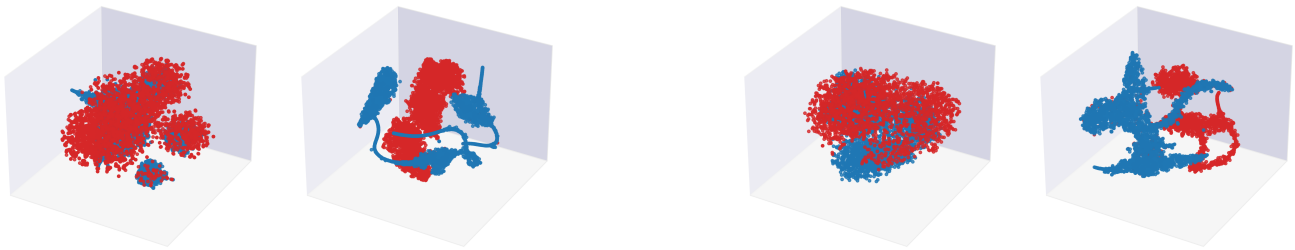
Before debiasing, normal and anomalous nodes exhibit substantial overlap, indicating that the learned representations are strongly affected by high-influence nodes and structural bias. After applying the debiasing module, the two categories become more separable, showing clearer cluster boundaries and reduced entanglement.

These observations demonstrate that the proposed debiasing mechanism effectively mitigates influence-related distortions in node embeddings, producing more discriminative and fairness-aware representations for anomaly detection.

5 Conclusion and Future Work

In this paper, we have proposed FairGAD , a fairness-aware framework for dynamic graph anomaly detection that explicitly addresses influence bias from a causal perspective. FairGAD integrates a Graph Expert Mixture Encoder that aggregates complementary structural and temporal signals, a Representation-Space Anomaly Generator that enriches the long-tail distribution of anomalies, and an Influence Debiasing Classifier that learns influence-invariant node representations via adversarial training. Extensive experiments on real-world dynamic graph datasets demonstrate that FairGAD achieves superior anomaly detection performance while improving fairness across nodes with varying influence levels, and ablation studies validate the contribution of each key component.

Despite its effectiveness, FairGAD has several limitations. First, the current encoder is instantiated with a fixed set of GNN backbones and a Transformer-based temporal module, which may incur non-trivial computational overhead on extremely large or dense dynamic graphs. Second, the Representation-Space Anomaly Generator relies on reasonably representative anomalous embeddings



(a) Before Debiasing **(b) After Debiasing**

Dataset: UCI

(a) Before Debiasing **(b) After Debiasing**

Dataset: EU-Core1

Figure 5: t-SNE visualization of node embeddings before and after applying the proposed influence debiasing mechanism. Red: anomalies; Blue: normal.

to learn the anomaly distribution; when labeled anomalies are extremely scarce or heavily noisy, the quality of generated features may degrade. Third, our influence debiasing mechanism focuses on node-level influence attributes derived from structural centrality, and does not yet consider other potential confounders such as attribute-level biases or temporal sampling artifacts. Finally, our empirical evaluation is conducted on a limited number of benchmark datasets, and further validation on broader application domains is still needed.

In future work, we plan to extend FairGAD along several directions. (1) *Model generalization:* We aim to adapt the Graph Expert Mixture Encoder to heterogeneous and attributed graphs with multiple node and edge types, and to explore lighter-weight temporal models for large-scale streaming settings. (2) *Advanced temporal and causal modeling:* We plan to investigate more advanced spatiotemporal architectures and causal discovery techniques to better capture evolving influence patterns and to further disentangle causal and spurious factors. (3) *Explainability and diagnostics:* We intend to integrate explanation modules that provide human-interpretable rationales for anomaly and fairness decisions, enabling practitioners to better understand and trust the debiasing process. (4) *Online and continual learning:* Finally, we will explore online learning strategies and continual debiasing mechanisms that allow FairGAD to adapt in real time to non-stationary, evolving graph environments. Overall, FairGAD provides a solid foundation for fairness-aware anomaly detection in dynamic graphs and opens avenues for future research at the intersection of graph learning, fairness, and causal representation learning.

References

[1] Hezhe Qiao, Hanghang Tong, Bo An, Irwin King, Charu Aggarwal, and Guansong Pang. 2025. Deep Graph Anomaly Detection: A Survey and New Perspectives. *IEEE Transactions on Knowledge & Data Engineering* 01 (2025), 1–20.

[2] Arturo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, and Charles E Leiserson. 2020. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5363–5370.

[3] Emanuele Rossi, Ben Chambers, Rex Ying, Michael Bronstein, and Petar Velickovic. 2020. Temporal Graph Networks for Deep Learning on Dynamic Graphs. *arXiv preprint arXiv:2006.10637* (2020).

[4] Meng-Jiun Chiou, Henghui Ding, Hanshu Yan, Changhu Wang, Roger Zimmermann, and Jiashi Feng. 2021. Recovering the unbiased scene graphs from the biased ones. In *Proceedings of the 29th ACM international conference on multimedia*. 1581–1590.

[5] Indro Spinelli, Simone Scardapane, Amir Hussain, and Aurelio Uncini. 2021. Fairdrop: Biased edge dropout for enhancing fairness in graph representation learning. *IEEE Transactions on Artificial Intelligence* 3, 3 (2021), 344–354.

[6] Ahmad Khajehnejad, Moein Khajehnejad, Mahmoudreza Babaei, Krishna P Gummadi, Adrian Weller, and Baharan Mirzasoleiman. 2022. Crosswalk: Fairness-enhanced node representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11963–11970.

[7] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. 2022. Rethinking Graph Neural Networks for Anomaly Detection. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 21076–21089. https://proceedings.mlr.press/v162/tang22b.html

[8] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of machine learning research* 17, 59 (2016), 1–35.

[9] Öykü Deniz Köse and Yanning Shen. 2021. Fairness-aware node representation learning. *arXiv preprint arXiv:2106.05391* (2021).

[10] Wei Fan, Kunpeng Liu, Rui Xie, Hao Liu, Hui Xiong, and Yanjie Fu. 2021. Fair graph auto-encoder for unbiased graph representations with wasserstein distance. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1054–1059.

[11] Oyku Deniz Kose and Yanning Shen. 2022. Fair contrastive learning on graphs. *IEEE Transactions on Signal and Information Processing over Networks* 8 (2022), 475–488.

[12] O Deniz Kose and Yanning Shen. 2022. Fair node representation learning via adaptive data augmentation. *arXiv preprint arXiv:2201.08549* (2022).

[13] Charlotte Laclau, Christine Largeron, and Manvi Choudhary. 2022. A survey on fairness for machine learning on graphs. *arXiv preprint arXiv:2205.05396* (2022).

[14] Zheyuan Liu, Chunhui Zhang, Yijun Tian, Erchi Zhang, Chao Huang, Yanfang Ye, and Chuxu Zhang. 2023. Fair graph representation learning via diverse mixture-of-experts. In *Proceedings of the ACM web conference 2023*. 28–38.

[15] Dandan Guo, Chaojie Wang, Baoxiang Wang, and Hongyuan Zha. 2022. Learning fair representations via distance correlation minimization. *IEEE Transactions on Neural Networks and Learning Systems* 35, 2 (2022), 2139–2152.

[16] Jintang Li, Jiaying Peng, Liang Chen, Zibin Zheng, Tingting Liang, and Qing Ling. 2022. Spectral adversarial training for robust graph neural network. *IEEE Transactions on Knowledge and Data Engineering* 35, 9 (2022), 9240–9253.

[17] Xiaobing Pei, Haoran Yang, and Gang Shen. 2023. HC-Ref: Hierarchical Constrained Refinement for Robust Adversarial Training of GNNs. *arXiv preprint arXiv:2312.04879* (2023).

[18] Tao Wu, Junhui Luo, Shaojie Qiao, Chao Wang, Lin Yuan, Xiao Pu, and Xingping Xian. 2024. Multiview-ensemble-learning-based robust graph convolutional networks against adversarial attacks. *IEEE Internet of Things Journal* 11, 16 (2024), 27700–27714.

[19] Zhongjian Zhang, Xiao Wang, Huichi Zhou, Yue Yu, Mengmei Zhang, Cheng Yang, and Chuan Shi. 2025. Can Large Language Models Improve the Adversarial Robustness of Graph Neural Networks?. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*. 2008–2019.

[20] Binchi Zhang, Yushun Dong, Chen Chen, Yada Zhu, Minnan Luo, and Jundong Li. 2023. Adversarial attacks on fairness of graph neural networks. *arXiv preprint arXiv:2310.13822* (2023).

[21] Weixuan Shen, Xiaobo Shen, and Shirui Pan. 2025. Adversarial Contrastive Graph Masked AutoEncoder Against Graph Structure and Feature Dual Attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 12515–12523.

[22] Junyuan Fang, Huimin Liu, Han Yang, Jiajing Wu, Zibin Zheng, and Chi K Tse. 2025. Mitigating the Structural Bias in Graph Adversarial Defenses. *arXiv e-prints* (2025), arXiv–2504.

[23] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 7370–7377.

[24] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. 2017. Graph attention networks. *stat* 1050, 20 (2017), 10–48550.

[25] Maryam Sanisales, Ali Vefghi, Zahed Rahmati, and Ali Darvishi Boloorani. 2024. Dust Source Emission Forecasting using GraphSAGE and Delaunay Graph. In *2024 10th International Conference on Signal Processing and Intelligent Systems (ICSPIS)*. IEEE, 173–177.

[26] Xingzhi Guo, Baojian Zhou, and Steven Skiena. 2022. Subset node anomaly tracking over large dynamic graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 475–485.

[27] Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. 2016. Edge weight prediction in weighted signed networks. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 221–230.

[28] Li Zheng, Zhenpeng Li, Jian Li, Zhao Li, and Jun Gao. 2019. AddGraph: Anomaly Detection in Dynamic Graph Using Attention-based Temporal GCN.. In *IJCAI*, Vol. 3. 7.

[29] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962* (2020).

[30] Kaize Ding, Jundong Wang, Kai Shu, and Huan Liu. 2019. Deep Anomaly Detection on Attributed Networks. In *Proceedings of the 2019 SIAM International Conference on Data Mining (SDM)*. SIAM, 594–602.

[31] Sambaran Bandyopadhyay, Lokesh N, Saley Vishal Vivek, and M Narasimha Murty. 2020. Outlier resistant unsupervised deep architectures for attributed network embedding. In *Proceedings of the 13th international conference on web search and data mining*. 25–33.

[32] Zhiming Xu, Xiao Huang, Yue Zhao, Yushun Dong, and Jundong Li. 2022. Contrastive attributed network anomaly detection with data augmentation. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 444–457.

[33] Haoyi Fan, Fengbin Zhang, and Zuoyong Li. 2020. Anomalydae: Dual autoencoder for anomaly detection on attributed networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5685–5689.

[34] Sheng Tian, Jihai Dong, Jintang Li, Wenlong Zhao, Xiaolong Xu, Baokun Wang, Bowen Song, Changhua Meng, Tianyi Zhang, and Liang Chen. 2023. SAD: semi-supervised anomaly detection on dynamic graphs. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. 2306–2314.

[35] Xiao Yang, Xuejiao Zhao, and Zhiqi Shen. 2025. A generalizable anomaly detection method in dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 22001–22009.

[36] Yifan Hong, Chuanqi Shi, Junyang Chen, Huan Wang, and Di Wang. 2024. Multitask asynchronous metalearning for few-shot anomalous node detection in dynamic networks. *IEEE Transactions on Computational Social Systems* (2024).

[37] Yifan Hong, Muhammad Asif Ali, Huan Wang, Junyang Chen, and Di Wang. 2025. ABNet: Mitigating Sample Imbalance in Anomaly Detection Within Dynamic Graphs. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*. ijcai.org, 2910–2918. doi:10.24963/IJCAI.2025/324