# Understanding the Influence of Extremely High-degree Nodes on Graph Anomaly Detection

Authors

Institutes

**Abstract.** Graph Anomaly Detection (GAD) has attracted considerable attention for its potential in detecting anomalies. However, an overlooked issue in prior research is the presence of extremely high-degree node, which can introduce noise into GAD, escalate computational costs, and intensify the problem of over-smoothing. To tackle this issue, this paper first presents a novel graph anomaly dataset, NFTGraph, characterized by a notable presence of extremely high-degree nodes. A series of experiments on this dataset sheds light on the influence of such nodes on GAD. Moreover, we introduce a novel model, the Super Node-Aware Graph Neural Network (SNGNN), designed to mitigate the noise emanating from extremely high-degree nodes. Experimental results demonstrate that SNGNN outperforms extant models, achieving an average improvement of over 2% in the Area Under the ROC Curve (AUROC), and effectively reducing noise.

**Keywords:** Graph Anomaly Detection · Extremely High-degree Nodes.

## 1 Introduction

Graph, a data structure with nodes and edges, has been widely used to model real-world scenarios, such as social networks [7], financial trading networks [19], and paper citing networks [5]. Since graph structures can capture relationships between entities, many anomaly detection methods are also based on graphs [1,23], aiming to identify anomalies that are distinct from the majority in the graph. Historically, numerous models for graph anomaly detection (GAD) have been put forth, such as CONAD [23] and PCGNN [12]. These models have contributed to the advancement of GAD.

However, a critical aspect overlooked by prior GAD studies pertains to the presence of extremely high-degree nodes, which can influence GAD models significantly. Firstly, since extreme high-degree nodes results in the formation of a tightly connected component within the graph, anomalies may inadvertently assimilate features from normal nodes through neighbor aggregation. This process complicates the delineation of anomalous nodes, introducing noise into the learning process. Moreover, the extensive connectivity associated with high-degree nodes could lead to elevated computational costs and exacerbate the issue of over-smoothing.

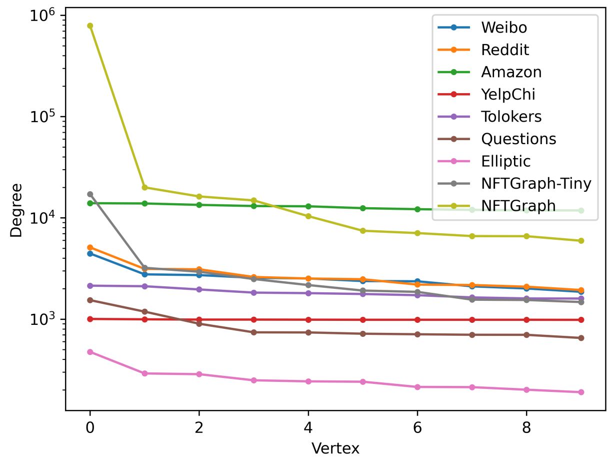



Fig. 1: The distribution of node degrees among the foremost ten nodes.

To elucidate the influence of extremely high-degree nodes, we first introduce a novel graph anomaly dataset termed NFTGraph. As depicted through the yellow and gray lines in Figure 1, both NFTGraph and its variant, NFTGraph-Tiny, manifest pronounced traits of extremely high-degree nodes. Additionally, utilizing these datasets, we investigate the influence of such nodes, including their role in introducing noise to GAD, amplifying computational costs, and intensifying the issue of over-smoothing. In response, we propose the Super Node-Aware Graph Neural Network (SNGNN), a novel GAD model that incorporates a Dummy Node and Link Predictor to mitigate the noise stemming from extremely high-degree nodes. Experimental results indicate that SNGNN surpasses current models, marked by an average increase of over 2% in the detection Area Under the ROC Curve (AUROC), alongside a reduction in noise. In essence, our contributions are multifaceted:

– We provide critical insights into the influence of extremely high-degree nodes, emphasizing their potential to disrupt GAD, amplify computational costs, and exacerbate the issue of over-smoothing.
– SNGNN is designed to mitigate the noise generated by extreme high-degree nodes. Experimental results indicate that SNGNN surpasses current methods across four datasets, registering an average enhancement of more than 2% in detection AUROC, while effectively reducing noise.

Moreover, we make the dataset and code publicly available on Github to facilitate further research.

## 2 Related Works

**Graph Anomaly Datasets:** Numerous graph anomaly datasets are widely employed in previous studies. For example, Weibo [10] and Reddit [10] are derived from social networks. Questions [16] is a question-answering dataset. Moreover, Amazon [15], Yelpchi [17], Tolokers [16] and Elliptic [21] are also famous GAD datasets. While these traditional datasets have played a pivotal role in advancing GAD, they fall short in accurately representing real-world networks by not encompassing the distinct attributes of extremely high-degree nodes. They obstruct a comprehensive understanding of the influence of extremely high-degree nodes on GAD.

**Degree-related GNNs:** Historically, several GNNs with a focus on degree-related considerations have been introduced to rectify node degree distribution biases. Notable examples include DEMO-Net [22] and SL-DSGCN [20], which implement degree-specific node transformations, and DegFairGNN [14] employs a function for generating debiasing contexts. Other GNNs addressing degree-related performance differences include Tail-GNN [13] and RawlsGCN [9], etc. However, these models have primarily been explored within the context of node or graph classification tasks. Thus far, a scarcity of research has addressed the ramifications of extremely high-degree nodes in anomaly detection tasks.

## 3 Data Collection and Properties

To understand the influence of extremely high-degree nodes, we initially gather data from NFT transactions on the blockchain and organize it into a graph structure. (1) **Raw data:** We extract certain fields of ERC-1155 NFT transaction on the Ethereum blockchain to compose the format of raw data. (Table 1). (2) **Graph Structure:** The *From* and *To* addresses, acting as the sending and receiving parties of a transaction, serve as the source and target nodes in the graph. An edge is established between the source and target nodes if tokens are transferred between them. Each node possesses 50-dimensional attributes. (3) **Labeling Suspicious Node:** We label nodes that exhibit interactions with the *ground-truth fraudulent nodes* (encompassing Ponzi schemes [4] and phishing scams [3]) exceeding a count of three instances as *suspicious nodes*. Suspicious nodes aim to alleviate the notable imbalance of ground-truth fraudulent nodes aligning with the NFTGraph's node set. (4) **Variant Dataset:** By extracting 20,000 of the most active nodes while excluding isolated nodes, we form NFTGraph-Tiny, leading to a substantial size reduction. This is executed with the recognition that certain GNNs may encounter challenges in handling extensive graphs within resource-constrained environments. More details are described in the supplemental materials.

Table 1: Format of raw NFT transaction data.

| *TxHash* | *From* | *To* | *Token* | *Timestamp* | *Amount* | *Value*($) | *TxFee*($) |
|---|---|---|---|---|---|---|---|
| 0xb5...b420 | 0x94...7293 | 0x6e...b7d3 | 0xd0...2430 | 20220730055230 | 1 | 78.52 | 2.23 |
| 0xa5...aeea | 0x00...0000 | 0xd8...ac95 | 0xd0...2430 | 20220730055230 | 14 | 0.0 | 0.98 |
| 0xa2...bdf1 | 0x5b...1abb | 0x4f...6580 | 0xd0...2430 | 20220730055138 | 1 | 0.0 | 0.33 |
| ... | ... | ... | ... | ... | ... | ... | ... |

**Properties:** Table 2 illustrates a comparison of statistical properties between NFTGraph and several other graph anomaly datasets [18]. The anomaly ratio of NFTGraph is only 0.39%, making it the lowest in the dataset. Moreover, in NFTGraph, the highest degree (No.1 deg) is 789,782, significantly surpassing No.2 deg. This pattern is consistent in NFTGraph-Tiny, but in other graph datasets, the discrepancy between No.1 deg and No.2 deg is less pronounced.

Table 2: Statistics of NFTGraph and some common graph anomaly datasets.

| Dataset | #Nodes | #Edges | Anomaly | Avg deg | No.1 deg | No.2 deg | $q_1$ | $q_2$ |
|---|---|---|---|---|---|---|---|---|
| Weibo | 8,405 | 416,368 | 10.3% | 99.08 | 4,447 | 2,769 | 44.88 | 27.95 |
| Reddit | 10,984 | 168,016 | 3.3% | 30.59 | 5,112 | 3,134 | 167.10 | 102.44 |
| Amazon | 11,944 | 8,847,096 | 9.5% | 1481.43 | 13,964 | 13,874 | 9.43 | 9.37 |
| YelpChi | 45,954 | 7,739,912 | 14.5% | 336.85 | 1,004 | 996 | 2.98 | 2.96 |
| Tolokers | 11,758 | 530,758 | 21.8% | 90.28 | 2,140 | 2,113 | 23.70 | 23.40 |
| Questions | 48,921 | 202,461 | 3.0% | 8.28 | 1,541 | 1,186 | 186.18 | 143.29 |
| Elliptic | 203,769 | 438,124 | 9.8% | 4.30 | 475 | 291 | 110.46 | 67.67 |
| NFTGraph-Tiny | 20,000 | 245,221 | 1.30% | 24.52 | 18,104 | 1,330 | **738.27** | 54.24 |
| NFTGraph | 1,161,847 | 2,851,407 | 0.39% | 4.91 | 789,782 | 20,000 | **160904.05** | 4074.64 |

# 4 Exploring the Influence of Extremely High-degree Node

## 4.1 Definition of Extremely High-degree Node

In the past, scholars have proposed the concepts of *influential nodes* and *central nodes* [6,2], which differ from extremely high-degree nodes. While extremely high-degree nodes primarily focus on node degree, influential nodes and central nodes can be defined in various ways, such as through $K$-shell value [8] or betweenness centrality [2], among others.

Define a high-degree node in a graph as a node with a degree greater than or equal to the average degree (avg_deg). An extremely high-degree node is defined as a node whose degree/avg_deg$\geqslant q$ ($q\geqslant 1$), indicating that the node's degree exceeds the average degree by $q$ times. $q$ is a hyperparameter that varies depending on the dataset. Let $q_1$ denote the hyperparameter selectively elevating No. 1 node to an extremely high-degree node, while $q_2$ signifies the hyperparameter concurrently elevating both No. 1 and No. 2 to extremely high-degree nodes. As seen from Table 2, for NFTGraph, $q_1$ exceeds 16,0000, and $q_2$ also exceeds 4,000. Furthermore, there is a considerable discrepancy between the values of $q_1$ and $q_2$, whereas for graph datasets including Weibo, Reddit, and Questions, $q_1$ and $q_2$ are closer. This demonstrates significant characteristics of extremely high-degree nodes. NFTGraph-Tiny exhibits similar features. The threshold $q$ is determined through hyperparameter tuning on the dataset's validation set to optimize the detection AUROC. For simplicity and clarity in illustrating the influence of extremely high-degree nodes, we set $q$ equal to $q_1$ in subsequent discussions. Therefore, only the node with the highest degree is considered as the extremely high-degree node (abbreviated as SN).

### 4.2 Experimental Settings

**Datasets:** Due to the similarity properties between NFTGraph and NFTGraph-Tiny, and the challenges faced by certain GNNs in handling large graphs, the proposed NFTGraph-Tiny is chosen as the foundational dataset. To assess the influence of SN, a variant dataset is introduced by removing SN and the edges connected to it. These two graphs are respectively denoted as w/ SN and w/o SN. From Table 3, it can be observed that without SN, No.1 degree decreased from 18,104 to 1,330, bringing it closer to the degrees of its immediate neighbors. AnomalyAvgDeg represents the average degree of all abnormal nodes. The average abnormal node degree of NFTGraph-Tiny is 27.66, indicating that abnormal nodes generally have lower degrees than SN. Moreover, to demonstrate the advantage of the proposed dataset, several commonly used and well-known graph anomaly datasets, namely Weibo [10,11], Reddit [11,24], and Questions [16], are selected for comparison, as shown in Table 3.

Table 3: Datases for exploring the influence of SN.

| Datasets | #Nodes | #Edges | #Feature | #Anomaly | No.1-5 Deg | AnomalyAvgDeg |
|---|---|---|---|---|---|---|
| NFTGraph-Tiny w/ SN | 20,000 | 245,221 | 50 | 259 | [18104,1330,1212,1020,917] | 27.66 |
| NFTGraph-Tiny w/o SN | 19,999 | 227,118 | 50 | 259 | [1330,1211,1020,916,793] | 27.66 |
| Weibo w/ SN | 8,405 | 416,368 | 400 | 868 | [4447,2769,2723,2558,2523] | 54.82 |
| Weibo w/o SN | 8,404 | 411,922 | 400 | 868 | [2767,2721,2556,2521,2376] | 54.82 |
| Reddit w/ SN | 10,984 | 168,016 | 64 | 366 | [5112,3134,3106,2608,2518] | 24.75 |
| Reddit w/o SN | 10,983 | 162,905 | 64 | 366 | [3134,3106,2608,741,2476] | 24.75 |
| Questions w/ SN | 48,921 | 202,461 | 301 | 1460 | [1541,1186,901,741,739] | 20.93 |
| Questions w/o SN | 48,920 | 200,921 | 301 | 1460 | [1185,900,740,738,717] | 20.93 |

**Task Description:** This section outlines a task aimed at identifying suspicious nodes. Formally, the objective is to train a model $f : f(u) \to \{0, 1\}$, where $\forall u \in \mathcal{V}$, $\mathcal{V}$ is node set, 1 denotes anomaly nodes and 0 denotes normal nodes.

**Models and Evaluation Metrics:** To comprehensively evaluate the influnce of SN, this section selects 34 anomaly detection models, including both supervised and unsupervised models, based on GNN and non-GNN models. Specifically, the unsupervised and non-GNN models [25] include OCSVM, LOF, CBLOF, COF, HBOS, SOD, COPOD, ECOD, LODA, and IForest; unsupervised and GNN-based models [11] include ANOMALOUS, ONE, OCGNN, CoLA, DONE, AnomalyDAE, CONAD, and DOMINANT; supervised and non-GNN models [19] include MLP, KNN, SVM, RF; supervised and GNN-based models [19] include GCN, SGC, GIN, GraphSAGE, GAT, GT, GAS, BernNet, AMNet, GHRN, GAT-Sep, PCGNN. Due to the severe class imbalance between suspicious and non-suspicious nodes, the Area Under the ROC Curve (AUROC) is chosen for evaluation. Other settings are in the supplemental materials.

Table 4: AUROC of anomaly detection models. Bold for significant change of AUROC (Higher AUROC).

| Models | Datasets | NFTGraph-Tiny | | Weibo | | Reddit | | Questions | |
|---|---|---|---|---|---|---|---|---|---|
| | | w/ SN | w/o SN | w/ SN | w/o SN | w/ SN | w/o SN | w/ SN | w/o SN |
| Unsupervised & | OCSVM | 0.4763 | **0.5018** | 0.8001 | 0.8017 | 0.5702 | 0.5703 | 0.5995 | 0.5995 |
| non-GNN-based | LOF | **0.5658** | 0.5332 | 0.5756 | 0.5756 | 0.5369 | 0.5372 | 0.5680 | 0.5679 |
| | CBLOF | 0.5134 | 0.5106 | 0.8003 | 0.8084 | 0.5809 | 0.5827 | 0.6016 | 0.6003 |
| | COF | **0.5662** | 0.5430 | 0.4877 | 0.4885 | 0.5755 | 0.5756 | 0.5591 | 0.5591 |
| | HBOS | 0.4998 | 0.5041 | 0.4038 | 0.4034 | 0.5338 | 0.5338 | 0.5951 | 0.5951 |
| | SOD | 0.6590 | 0.6405 | 0.4258 | 0.4249 | 0.5495 | 0.5402 | 0.5526 | 0.5553 |
| | COPOD | 0.5977 | 0.5977 | 0.4736 | 0.4738 | 0.4974 | 0.4975 | 0.6059 | 0.6059 |
| | ECOD | 0.5240 | 0.5241 | 0.4774 | 0.4775 | 0.4999 | 0.4999 | 0.6015 | 0.6015 |
| | LODA | **0.5749** | 0.5412 | 0.7139 | 0.7096 | 0.5630 | 0.5633 | 0.5745 | 0.5745 |
| | IForest | 0.6016 | 0.6009 | 0.5500 | 0.5502 | **0.5942** | 0.5514 | 0.6057 | 0.6020 |
| Unsupervised | ANOMALOUS | 0.6159 | **0.6818** | 0.9876 | 0.9876 | 0.5688 | 0.5629 | 0.5527 | 0.5530 |
| GNN-based | ONE | **0.5445** | 0.4992 | 0.6637 | 0.6518 | **0.5356** | 0.5157 | 0.4867 | **0.5102** |
| | OCGNN | **0.6327** | 0.5389 | 0.8251 | 0.8257 | **0.6308** | 0.6139 | 0.5590 | **0.5745** |
| | CoLA | **0.4943** | 0.4021 | 0.4254 | **0.4464** | 0.4963 | **0.5409** | 0.5306 | 0.5465 |
| | DONE | 0.5734 | 0.5858 | 0.5536 | **0.6569** | 0.5518 | 0.5556 | 0.6644 | 0.6639 |
| | AnomalyDAE | 0.5555 | **0.5803** | 0.8256 | 0.8268 | 0.5805 | 0.5709 | 0.4771 | **0.4995** |
| | CONAD | 0.5382 | 0.5424 | 0.6311 | **0.7050** | 0.4680 | **0.5174** | 0.6019 | 0.6021 |
| | DOMINANT | 0.6026 | **0.6251** | **0.7015** | 0.6290 | 0.5129 | 0.5138 | 0.6036 | 0.6028 |
| Supervised | MLP | 0.5645 | **0.6730** | 0.9738 | 0.9669 | 0.6771 | 0.6765 | 0.6753 | 0.6785 |
| non-GNN-based | KNN | 0.5994 | **0.6204** | 0.9672 | 0.9674 | 0.6067 | **0.6301** | 0.6760 | 0.6789 |
| | SVM | 0.5756 | 0.5773 | 0.9536 | 0.9539 | 0.6622 | 0.6659 | 0.6359 | 0.6410 |
| | RF | **0.6539** | 0.6314 | 0.9864 | 0.9865 | 0.6290 | 0.6312 | 0.5621 | 0.5512 |
| Supervised | GCN | 0.6580 | 0.6401 | 0.9830 | 0.9867 | 0.7172 | 0.7122 | 0.7018 | 0.7011 |
| GNN-based | SGC | 0.5968 | **0.6179** | 0.9892 | 0.9893 | 0.6842 | 0.6885 | 0.6911 | 0.6921 |
| | GIN | **0.6688** | 0.6164 | 0.9881 | 0.9901 | **0.7028** | 0.6574 | 0.7185 | 0.7185 |
| | GraphSAGE | 0.5777 | **0.6437** | 0.9934 | 0.9932 | 0.6949 | 0.7130 | 0.7197 | 0.7179 |
| | GAT | 0.6510 | 0.6405 | 0.9800 | 0.9816 | 0.6866 | 0.6724 | 0.7037 | 0.7093 |
| | GT | 0.6163 | **0.6518** | 0.9899 | 0.9897 | 0.6444 | **0.6682** | 0.6949 | 0.7134 |
| | GAS | 0.6663 | 0.6636 | 0.9828 | 0.9824 | **0.6858** | 0.6627 | **0.7118** | 0.6913 |
| | BernNet | 0.6230 | **0.6628** | 0.9783 | 0.9853 | 0.6868 | 0.6763 | 0.6951 | 0.7095 |
| | AMNet | **0.6970** | 0.6601 | 0.9808 | 0.9858 | 0.6445 | 0.6371 | 0.6990 | 0.6989 |
| | GHRN | 0.6734 | 0.6656 | 0.9792 | 0.9892 | 0.6894 | **0.7180** | 0.7204 | 0.7210 |
| | GAT-Sep | **0.6775** | 0.6534 | 0.9846 | 0.9863 | 0.6665 | 0.6739 | 0.6913 | 0.6892 |
| | PCGNN | **0.6895** | 0.6377 | 0.9848 | 0.9846 | 0.6779 | 0.6785 | **0.6929** | 0.6692 |

### 4.3 Influence of SN on GNN-based and non-GNN-based Models

Table 4 presents the AUROC of models on NFTGraph-Tiny, Weibo, Reddit, and Questions datasets, along with their corresponding graphs without SN. Refining Table 4, the **significant change rate** is defined as the proportion of models with AUROC changes exceeding 2% (±2%) after removing SN, while the **positive significant change rate** indicates an augmentation in AUROC (+2%) after SN removal.

Table 5 illustrates the significant change rates for both non-GNN and GNN models. Remarkably, the significant change rate for NFTGraph-Tiny surpasses that of Weibo, Reddit, and Questions, with datasets such as Weibo and Questions showing a minimal 0% significant change rate. This highlights the distinct advantage of employing NFTGraph-Tiny for exploring the influence of extremely high-degree nodes, thereby suggesting the limited utility of other datasets in this context.

Table 5: Significant change rate for non-GNN and GNN models.

| | NFTGraph-Tiny | Weibo | Reddit | Questions |
|---|---|---|---|---|
| non-GNN-based | **50.00%** | 0.00% | 14.29% | 0.00% |
| GNN-based | **70.00%** | 20.00% | 40.00% | 25.00% |
| non-GNN-based w/o SN + | 42.86% | - | 50.00% | - |
| GNN-based w/o SN + | 50.00% | 75.00% | 50.00% | 60.00% |

Across the four datasets, the significant change rates of GNN-based models are substantially higher than those of non-GNN-based models. Specifically, within the NFTGraph-Tiny dataset, GNN-based models show a remarkable significant change rate of 70.00%, in contrast to the 50.00% observed for non-GNN-based models. Furthermore, Table 5 highlights that the positive significant change rate for GNN-based models surpasses 50%. This outcome appears counterintuitive since the inclusion of SN is theoretically expected to enhance the informational content, suggesting that graphs incorporating SN should uniformly exhibit superior AUROC performance. Nonetheless, the empirical data reveals that over half of the GNN-based models achieve higher AUROC scores upon the removal of SN, with some models registering a remarkable AUROC increase of up to 7% (e.g., ANOMALOUS). This phenomenon indicates that SN and its connected edges might introduce noise that, via neighbor aggregation, obscures the distinction between normal and anomalous nodes, complicating their differentiation.

### 4.4 Impact of SN on Unsupervised and Supervised GADs

Given that GNN-based models generally exhibit higher significant change rates compared to non-GNN-based models, the experimental results of GNN-based models are further analyzed to assess the influence of SN on unsupervised and supervised settings.

From Table 6, it can be observed that, regardless of the supervised or unsupervised setting, the significant change rate of NFTGraph-Tiny is not lower than that of the other three datasets, which also shows the advantages of NFTGraph-Tiny. Additionally, the positive significant change rates of unsupervised GNN models are higher than those of supervised GNN models across the four datasets. This suggests that, after removing SN, unsupervised GNN models achieve a higher proportion of models with increased AUROC. This phenomenon may be attributed to the absence of training labels in unsupervised GNN models, making the noise introduced by SN edges more impactful for anomaly detection.

Consequently, after removing SN, unsupervised GNN models may learn better, highlighting the importance of considering the noise introduced by SN.

Table 6: Significant change rate for unsupervised and supervised GADs.

| | NFTGraph-Tiny | Weibo | Reddit | Questions |
|---|---|---|---|---|
| Unsupervised GAD | **50.00%** | 50.00% | 50.00% | 37.50% |
| Supervised GAD | **66.67%** | 0.00% | 33.33% | 16.67% |
| Unsupervised GAD w/o SN + | 75.00% | 75.00% | 50.00% | 100.00% |
| Supervised GAD w/o SN + | 50.00% | - | 50.00% | 0.00% |

### 4.5 Computational Cost

Considering the highlighted benefits of NFTGraph-Tiny, it will be the primary dataset employed for further investigation in subsequent sections. Additionally, to more effectively demonstrate the impact on computational costs, this subsection will also incorporate NFTGraph to provide statistics on the average number of node neighbors and the execution time for both 1-layer and 2-layer Graph Attention Networks (GAT).

Table 7 shows the average number of 1-hop and 2-hop neighbors of GAT in NFTGraph-Tiny is 11.26 and 2386.39. Upon removing SN, the average number of 1-hop neighbors remains relatively unchanged, while the average number of 2-hop neighbors sharply decreases to 123.88. This is due to the fact that the degree of SN in NFTGraph-Tiny is 18,104, indicating that the majority of the whole 20,000 nodes in the graph are connected to SN. More pronounced disparities are observed in NFTGraph and NFTGraph w/o SN. Table 8 demonstrates how the execution time of GAT varies with different numbers of layers. Notably, on NFTGraph-Tiny and NFTGraph, removing SN leads to a nonlinear decrease in execution time, with a more significant reduction observed for 2-layer GAT compared to a single layer. The presence of SN significantly impacts the computational cost, leading to a substantial increase in both the average number of node neighbors and the execution time.

Table 7: Average number of node neighbors for GAT at different hops.

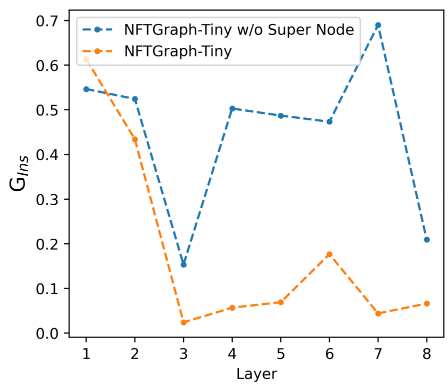
| Dataset / Hops | 1-hop | 2-hop |
|---|---|---|
| NFTGraph-Tiny | 11.26 | **2386.39** |
| NFTGraph-Tiny w/o SN | 10.36 | 123.88 |
| NFTGraph | 2.45 | **27647.54** |
| NFTGraph w/o SN | 1.86 | 20.42 |

Table 8: Execution time (s) of GAT with different numbers of layers.

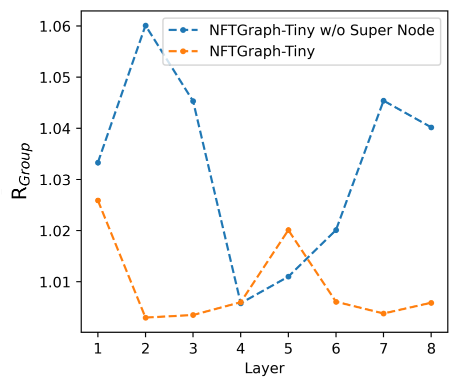
| Dataset / Layer Number | 1-layer | 2-layer |
|---|---|---|
| NFTGraph-Tiny | 4.89 | **5.84** |
| NFTGraph-Tiny w/o SN | 4.58 | 5.31 |
| NFTGraph | 49.26 | **91.06** |
| NFTGraph w/o SN | 20.03 | 33.73 |

### 4.6 Over-smoothing

To investigate the influence of extremely high-degree nodes on the issue of over-smoothing, we compute two over-smoothing metrics [26]: Instance Information Gain ($G_{Ins}$) and Group Distance Ratio ($R_{Group}$). These calculations are performed across different layer numbers of GAT applied to both the NFTGraph-Tiny dataset and its variant (w/o SN). Generally, lower values of these metrics indicate a higher level of over-smoothing.




(a) Over-smoothing metric $G_{Ins}$




(b) Over-smoothing metric $R_{Group}$

Fig. 2: Over-smoothing metrics $G_{Ins}$ and $R_{Group}$ for different layer numbers of GAT on NFTGraph-Tiny and NFTGraph-Tiny w/o SN.

Figure 2 illustrates the changes in the over-smoothing metrics $G_{Ins}$ and $R_{Group}$ for different layer numbers of GAT. It is evident that, in the majority of cases, as the number GAT's layers increases, the metrics decrease, indicating a progressive over-smoothing of node representations. Upon removal of SN, both $G_{Ins}$ and $R_{Group}$ metrics exhibit an increase compared to the original, thereby alleviating the over-smoothing phenomenon. Therefore, the results demonstrate that with the same layer number of GNN, the presence of extremely high-degree nodes increases the likelihood of over-smoothing.

## 5 Method and Experiments

### 5.1 SNGNN

In this section, we introduce a novel graph anomaly detection model, named Super Node-Aware Graph Neural Network (SNGNN), aimed at mitigating the noise generated by nodes with extremely high degrees. The conceptual framework of SNGNN is depicted in Figure 3.

In SNGNN, we first introduce a dummy node (DN) as a new node, and establish edges between DN and all anomaly nodes in $G_0$, as follows:

$$\mathcal{V} = \mathcal{V} \cup \{DN\}, \mathcal{E} = \mathcal{E} \cup \{e'\}, \tag{1}$$

where,

$$e' = (DN, s), \forall s \in \mathcal{N}_{anomaly}, \tag{2}$$

where $\mathcal{V}$ represents the set of nodes, and $\mathcal{E}$ denotes the set of all edges. Edge $e = (u, v)$ connects $u$ and $v$ ($\{u, v\} \in \mathcal{V}$). $\mathcal{N}_{anomaly}$ denotes the set of anomaly nodes.

Subsequently, we undertake the task of link prediction for the SN. This component inputs the SN and its neighboring nodes into a Link Predictor (LP), yielding a probability vector ($\boldsymbol{c}$) that represents the likelihood of an edge existing between SN and its neighbors. Here, we employ a straightforward dot product as the LP, which is articulated as follows:

$$c_n =< h_{SN}, h_n >, \forall n \in \mathcal{N}_{neigbhour} \tag{3}$$

where $h_n$ is the hidden vector of node $n$, $\mathcal{N}_{neighbour}$ is the set of the original neighbors of SN, and $<, >$ indicates the dot product between two vectors. Then, the $p_1$-quantile ($c_{p_1}$) and $p_2$-quantile ($c_{p_2}$) are derived from the probability vector $\mathbf{c}$. Nodes with probabilities below $c_{p_1}$ are severed from SN, whereas those with probabilities above $c_{p_2}$ are linked to SN. Formally:

$$\mathbf{A}_{SN,n} = \begin{cases} 0, c_n \leq c_{p_1} \\ 1, c_n \geq c_{p_2} \end{cases}, \forall n \in \mathcal{N}_{neigbhour}, \tag{4}$$

where $\mathbf{A}_{SN,n} = 0$ signifies the absence of an edge between the SN and node $n$ in the adjacency matrix $\mathbf{A}$.

Finally, we update the node representations by aggregating neighbor information according to the updated graph topology at each iteration. Utilizing the node representations from the final layer, we then ascertain whether nodes are anomalous.

The rationale behind the development of SNGNN encompasses several key aspects: Firstly, considering the typically lower degree of anomalous nodes themselves (as indicated by AnomalyAvgDeg in Table 3), the integration of a dummy node (DN) connected to all identified anomalous nodes serves to mitigate the imbalance between low-degree anomalous nodes and extremely high-degree nodes to some extent. Moreover, the inclusion of DN facilitates the acquisition of a more "pure" representation of anomalous nodes during the propagation process. This is crucial as the connection of anomalous nodes to their normal counterparts results in the amalgamation of information from normal nodes during the neighbor aggregation phase by GNN, which is counterproductive for accurately modeling the representations of anomalous nodes.

Secondly, the implementation of thresholds $p_1$ and $p_2$ allows for the disconnection of edges with probabilities below $p_1$ and the maintenance or addition

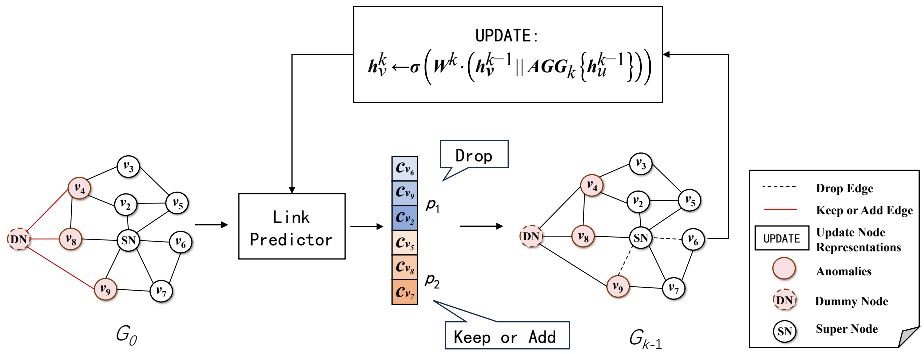



Fig. 3: Schema of SNGNN.

of edges with probabilities above $p_2$. This approach is designed to evaluate the reliability of edges associated with the SN. By learning from anomaly labels, the model is capable of autonomously determining the optimal thresholds to either facilitate or inhibit message transmission, thereby diminishing noise in the connections to SN and enhancing the model's resilience.

### 5.2 Experiments

**Setup:** NFTGraph-Tiny, Weibo, Reddit, and Questions datasets are used, and the task is anomaly detection. The baseline models include three basic GNNs: GCN, GAT, and GraphSAGE, as well as three GNNs that achieve better performance in Table 4: PCGNN, GAS, and GIN. Settings are different from Section 4.2, which can be found in the supplemental materials.

**Results:** Table 9 shows the comparison of AUROC between SNGNN and other GADs. Across all four datasets, SNGNN consistently achieves the highest AUROC, with an average increase of over 2% compared to other models. Notably, on the NFTGraph-Tiny dataset, SNGNN achieves an AUROC of 0.6980, surpassing GHRN by 5%. Similarly, on the Weibo dataset, SNGNN's AUROC is 0.9926, higher than the second-best model GraphSAGE. Likewise, on Reddit and Questions, SNGNN outperforms GCN (0.7189) and GraphSAGE (0.7259) models, resulting in a 1% AUROC improvement. It is noteworthy that SNGNN exhibits at least a 5% AUROC improvement on NFTGraph-Tiny, significantly higher than the approximately 1% improvement observed on other datasets. This performance disparity stems from the notably higher degree values of social networks in NFTGraph-Tiny compared to Weibo, Reddit, and Questions, as evidenced in Table 3, underscoring SNGNN's superior efficacy in this context.

### 5.3 Ablation Study

To validate the effectiveness of SNGNN, we design several ablation tests. Specifically, while keeping the other parts and hyperparameters unchanged, Dummy

Table 9: Comparison of AUROC between SNGNN and other GADs. The best performance is shown in bold, while the second-best performance is underlined.

| Model / Dataset | NFTGraph-Tiny | Weibo | Reddit | Questions |
|---|---|---|---|---|
| GCN | 0.5953 | 0.9875 | <u>0.7189</u> | 0.6819 |
| GAT | 0.6226 | 0.9902 | 0.6733 | 0.7167 |
| GraphSAGE | 0.6428 | <u>0.9917</u> | 0.6800 | <u>0.7259</u> |
| PCGNN | 0.5832 | 0.9848 | 0.7079 | 0.6784 |
| GAS | 0.5552 | 0.9915 | 0.6996 | 0.7111 |
| GIN | 0.5929 | 0.9908 | 0.6872 | 0.7160 |
| AMNet | 0.6263 | 0.9764 | 0.6731 | 0.7064 |
| GHRN | <u>0.6479</u> | 0.9860 | 0.6963 | 0.7164 |
| SNGNN | **0.6980** | **0.9926** | **0.7272** | **0.7325** |

Node (referred to as w/o DN) and Link Predictor (referred to as w/o LP) are removed separately, and then the performance is observed.

Table 10 shows the results of ablation tests. Notably, the removal of the Dummy Node (DN) results in a diminished detection AUROC for SNGNN. This effect is particularly pronounced on the NFTGraph-Tiny and Questions datasets, where the AUROC for SNGNN drops by over 5% and 3%, respectively. Similarly, the elimination of the Link Predictor (LP) also leads to a reduction in AUROC, with a significant decrease of more than 6% on the NFTGraph-Tiny dataset. Consequently, the incorporation of both Dummy Node and Link Predictor is essential for the effectiveness of SNGNN.

Table 10: Results of ablation study for SNGNN.

| Dataset/Model | w/o DN | w/o LP | SNGNN |
|---|---|---|---|
| NFTGraph-Tiny | 0.6414 | 0.6351 | **0.6980** |
| Weibo | 0.9914 | 0.9919 | **0.9926** |
| Reddit | 0.7188 | 0.7239 | **0.7272** |
| Questions | 0.7086 | 0.727 | **0.7325** |

Table 11: Performance of noise reducing.

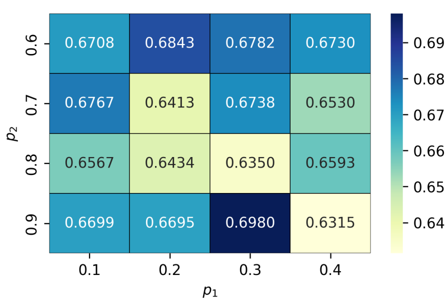
| Dataset/Model | w/o DN&LP | SNGNN |
|---|---|---|
| NFTGraph-Tiny | 0.5807 | **1.9401** |
| Weibo | 1.6492 | **2.9917** |
| Reddit | 0.8738 | **1.4031** |
| Questions | 1.1473 | **1.7393** |

Additionally, we compute the average inter-group distance (DisInter) [26] between anomalies and normal nodes. A higher DisInter metric indicates that SNGNN more effectively distances the embeddings of anomalous nodes from those of normal ones, signifying more precise outcomes. This metric serves as an indicator of the model's efficiency in noise reduction. In Table 11, the baseline represents the SNGNN model without the Dummy Node (DN) and Link Predictor (LP), essentially constituting the GNN backbone of SNGNN (corresponding to the *Update* stage in Figure 3). For the Reddit dataset, the baseline is GCN, while for other datasets, it is GraphSAGE. Table 11 demonstrates that SNGNN achieves a greater DisInter value compared to its baseline, indicating that the incorporation of DN and LP enhances the model's ability to reduce noise.

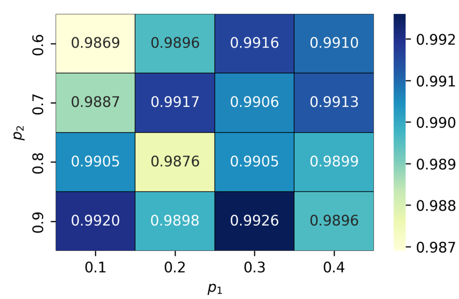
## 5.4 Parameter Sensitivity Analysis

The hyperparameters $p_1$ and $p_2$ represent the $p_1$ and $p_2$ percentiles of the LP output probabilities, determining the removal and addition of edges of SN in each training iteration. $p_1$ and $p_2$ indicate the strength and confidence of edge removal (addition), thus it is necessary to analyze the sensitivity to $p_1$ and $p_2$.

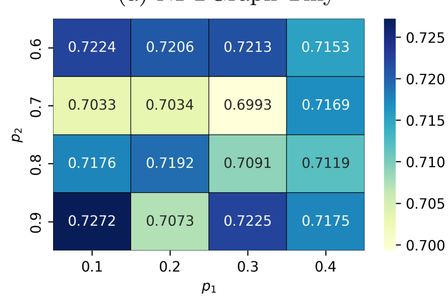
Figure 4 shows the detection AUROC of the SNGNN model under different values of $p_1$ and $p_2$ on four datasets. The optimal values of $p_1$ and $p_2$ for NFTGraph-Tiny and Weibo are (0.3, 0.9), for Reddit is (0.1, 0.9), and for Questions is (0.4, 0.7). The optimal values for these four datasets are not located in the central area (i.e., $0.2 \leqslant p_1 \leqslant 0.3$, $0.7 \leqslant p_2 \leqslant 0.8$), indicating that for the edges of SN, the best practice is to ensure that the number of edge removals and additions is either relatively high or relatively low, or one is high while the other is low. If the number of edge removals and additions in each iteration is kept at a moderate level, it is difficult to achieve the optimal state.



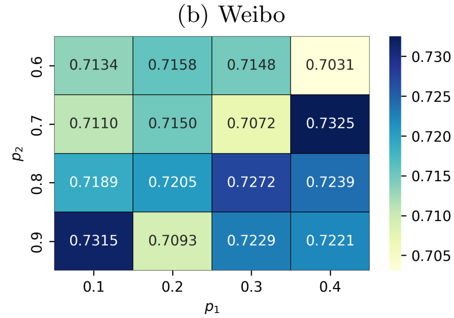

(a) NFTGraph-Tiny




(b) Weibo




(c) Reddit




(d) Questions

Fig. 4: Parameter sensitivity.

## 6 Conclusion and Limitation

In this paper, our focus is on exploring the influence of extremely high-degree nodes on graph anomaly detection (GAD). To address this objective, we first introduce a novel graph dataset, NFTGraph, and conduct a comprehensive analysis of the influence, including the introduction of noise to GAD, the escalation of computational costs, and the exacerbation of over-smoothing phenomena. Additionally, we propose a novel model called Super Node-Aware Graph Neural Network (SNGNN) to mitigate the noise introduced by extremely high-degree nodes. SNGNN demonstrates superior performance compared to existing models, achieving an average improvement in detection AUROC of over 2% while efficiently reducing noise. Although SNGNN specifically targets the SN, it can be extended and applied to all extremely high-degree nodes. Furthermore, there is potential to enhance SNGNN further to mitigate the influence on computational costs and over-smoothing. Additionally, it is important to recognize that extremely high-degree nodes are prevalent in various real-world networks, such as influencers in social networks, banks in financial trading networks, and super-spreaders in disease transmission networks. In the future, our aim is to investigate these aspects beyond the scope of blockchain transaction networks.

## References

1. Bandyopadhyay, S., Vivek, S.V., Murty, M.: Outlier resistant unsupervised deep architectures for attributed network embedding. In: Proceedings of the 13th international conference on web search and data mining. pp. 25–33 (2020)
2. Bhattacharya, R., Nagwani, N.K., Tripathi, S.: Detecting influential nodes with topological structure via graph neural network approach in social networks. International Journal of Information Technology **15**(4), 2233–2246 (2023)
3. Chen, L., Peng, J., Liu, Y., Li, J., Xie, F., Zheng, Z.: Phishing scams detection in ethereum transaction network. ACM transactions on internet technology(TOIT) **21**(1), 1–16 (2020)
4. Chen, W., Zheng, Z., Cui, J., Ngai, E., Zheng, P., Zhou, Y.: Detecting ponzi schemes on ethereum: Towards healthier blockchain technology. In: Proceedings of the world wide web conference(WWW). pp. 1409–1418 (2018)
5. Giles, C.L., Bollacker, K.D., Lawrence, S.: Citeseer: An automatic citation indexing system. In: Proceedings of the third conference on digital libraries. pp. 89–98 (1998)
6. Hafiene, N., Karoui, W., Romdhane, L.B.: Influential nodes detection in dynamic social networks: A survey. Expert Systems with Applications **159**, 113642 (2020)
7. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in Neural Information Processing Systems(NeurIPS) **30** (2017)
8. Hu, H., Sun, Z., Wang, F., Zhang, L., Wang, G.: Exploring influential nodes using global and local information. Scientific Reports **12**(1), 22506 (2022)
9. Kang, J., Zhu, Y., Xia, Y., Luo, J., Tong, H.: Rawlsgcn: Towards rawlsian difference principle on graph convolutional network. In: Proceedings of the world wide web conference(WWW). pp. 1214–1225 (2022)
10. Kumar, S., Zhang, X., Leskovec, J.: Predicting dynamic embedding trajectory in temporal interaction networks. In: Proceedings of the 27th sigkdd conference on knowledge discovery and data mining(SIGKDD). pp. 1269–1278 (2019)

11. Liu, K., Dou, Y., Zhao, Y., Ding, X., Hu, X., Zhang, R., Ding, K., Chen, C., Peng, H., Shu, K., et al.: Bond: Benchmarking unsupervised outlier node detection on static attributed graphs. Advances in Neural Information Processing Systems(NeurIPS) **35**, 27021–27035 (2022)
12. Liu, Y., Ao, X., Qin, Z., Chi, J., Feng, J., Yang, H., He, Q.: Pick and choose: a gnn-based imbalanced learning approach for fraud detection. In: Proceedings of the world wide web conference(WWW). pp. 3168–3177 (2021)
13. Liu, Z., Nguyen, T.K., Fang, Y.: Tail-gnn: Tail-node graph neural networks. In: Proceedings of the 27th sigkdd conference on knowledge discovery and data mining(SIGKDD). pp. 1109–1119 (2021)
14. LIU, Z., NGUYEN, T.K., FANG, Y.: On generalized degree fairness in graph neural networks. In: Proceedings of the 37th international conference on artificial intelligence(AAAI). pp. 7–14 (2023)
15. McAuley, J.J., Leskovec, J.: From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In: Proceedings of the 22nd international conference on blockchain. pp. 897–908 (2013)
16. Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., Prokhorenkova, L.: A critical look at the evaluation of gnns under heterophily: are we really making progress? arXiv preprint arXiv:2302.11640 (2023)
17. Rayana, S., Akoglu, L.: Collective opinion spam detection: Bridging review networks and metadata. In: Proceedings of the 21th sigkdd conference on knowledge discovery and data mining(SIGKDD). pp. 985–994 (2015)
18. Tang, J., Hua, F., Gao, Z., Zhao, P., Li, J.: Gadbench: Revisiting and benchmarking supervised graph anomaly detection. Advances in Neural Information Processing Systems(NeurIPS) **36** (2024)
19. Tang, J., Li, J., Gao, Z., Li, J.: Rethinking graph neural networks for anomaly detection. In: Proceedings of the international conference on machine learning(ICML). pp. 21076–21089 (2022)
20. Tang, X., Yao, H., Sun, Y., Wang, Y., Tang, J., Aggarwal, C., Mitra, P., Wang, S.: Investigating and mitigating degree-related biases in graph convoltuional networks. In: Proceedings of the 29th international conference on information and knowledge management(CIKM). pp. 1435–1444 (2020)
21. Weber, M., Domeniconi, G., Chen, J., Weidele, D.K.I., Bellei, C., Robinson, T., Leiserson, C.E.: Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. arXiv preprint (2019)
22. Wu, J., He, J., Xu, J.: Demo-net: Degree-specific graph neural networks for node and graph classification. In: Proceedings of the 25th sigkdd conference on knowledge discovery and data mining(SIGKDD). pp. 406–415 (2019)
23. Xu, Z., Huang, X., Zhao, Y., Dong, Y., Li, J.: Contrastive attributed network anomaly detection with data augmentation. In: Advances in knowledge discovery and data mining: 26th pacific-asia Conference. pp. 444–457. Springer (2022)
24. Zhao, T., Deng, C., Yu, K., Jiang, T., Wang, D., Jiang, M.: Error-bounded graph anomaly loss for gnns. In: Proceedings of the 28th international conference on information and knowledge management(CIKM). pp. 1873–1882 (2020)
25. Zhao, Y., Nasrullah, Z., Li, Z.: Pyod: A python toolbox for scalable outlier detection. Journal of machine learning research **20**(96), 1–7 (2019)
26. Zhou, K., Huang, X., Li, Y., Zha, D., Chen, R., Hu, X.: Towards deeper graph neural networks with differentiable group normalization. Advances in Neural Information Processing Systems(NeurIPS) **33**, 4917–4928 (2020)