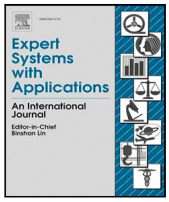
Expert Systems With Applications 237 (2024) 121533

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Controlled graph neural networks with denoising diffusion for anomaly detection

Xuan Li [a,e], Chunjing Xiao [b,d,*], Ziliang Feng [a], Shikang Pang [b], Wenxin Tai [c], Fan Zhou [c,d]

[a] National Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University, Chengdu 610065, China
[b] School of Computer and Information Engineering, Henan University, Kaifeng 475000, China
[c] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[d] Kash Institute of Electronics and Information Industry, Kashi 844199, China
[e] Sichuan Post and Telecommunication College, Chengdu 610067, China

ARTICLE INFO

Keywords:
Anomaly detection
Graph data augmentation
Diffusion models
Graph neural networks

ABSTRACT

Leveraging labels in a supervised learning framework as prior knowledge to enhance network anomaly detection has become a trend. Unfortunately, just a few labels are typically available due to the expensive labeling cost. The limited labeled data might not adequately represent the underlying distribution of the data, making the trained model fail to accurately detect anomalies in test data that fall outside the labeled data's distribution. Recently, data augmentation has been widely used to address this issue. However, general graph data augmentation methods might lead to suboptimal performance when directly applying to network anomaly detection as they might alter graph semantics. In this paper, we provide a denoising diffusion probabilistic model (DDPM)-based Controlled Graph Neural Networks (ConGNN) that can address the problem of insufficient labeled data. In particular, we propose a graph-specific diffusion model-based generator which can inject the characteristics of a reference node into another source node. This generator is adopted to steer the procedure of neighborhood aggregation in GNN to build a controlled GNN for generating augmented data. Based on these augmented data, we present a data-enclosing hypersphere model with our designed consistency regularization term to perform anomaly detection. Extensive experiments demonstrate the superior performance of our model compared to the state-of-the-art baselines.

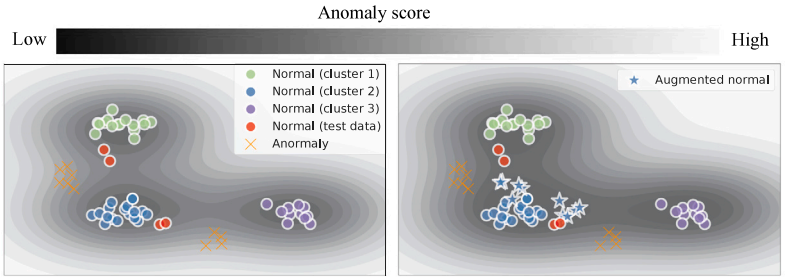
1. Introduction

Anomaly detection aims to identify the anomalies which are considered as data objects deviating dramatically from the majority (Hilal, Gadsden, & Yawney, 2022; Ruff et al., 2021). As a few anomalies may cause tremendous loss, anomaly detection is a crucial task for a wide range of applications from detecting network attacks in cybersecurity to inspecting sybil accounts in social networks (Ma et al., 2021; Pang, Shen, Cao, & Hengel, 2021). Correspondingly, growing attention has been paid to network anomaly detection. Since labeling anomalies is highly labor-intensive, existing methods are predominately developed in an unsupervised manner, such as autoencoder-based (Ding, Li, Bhanushali, & Liu, 2019; Li, Huang, Li, Du, & Zou, 2019) and matrix factorization-based methods (Bandyopadhyay, Lokesh, & Murty, 2019; Li, Dani, Hu, & Liu, 2017). However, unsupervised approaches may result in noisy or uninteresting data instances in predicted anomalies because of lacking prior knowledge on the anomalies of interest. Hence, leveraging labeled samples as the prior knowledge is desired (Kumagai, Iwata, & Fujiwara, 2021; Zhou, Huang, Liu, Tan, & Chung, 2022). This accordingly brings another challenge to researchers that the labeled samples are usually limited, since it is relatively costly to collect many labeled samples.

Insufficient labeled data can cause general detection models to suffer from performance degradation. Limited labeled samples cannot cover the data distribution well, which can lead to inaccurate decision boundaries for the test data with a different distribution from the labeled data. Generally, samples including normal and anomalous ones can be divided into different clusters (Zhou et al., 2022). Limited samples can only cover partial data distributions in the clusters. Thus, the model trained on limited samples might not provide proper decision boundaries to identify those samples out of the distribution of the training data. An example *w.r.t.* decision boundaries obtained based on limited samples is illustrated in Fig. 1(a). Here the decision boundaries are trained based on normal samples that can be grouped into three

* Corresponding author.
E-mail addresses: lixuanlmw@stu.scu.edu.cn (X. Li), chunjingxiao@gmail.com (C. Xiao), fengziliang@scu.edu.cn (Z. Feng), pangsk0604@henu.edu.cn (S. Pang), wxtai@std.uestc.edu.cn (W. Tai), fan.zhou@uestc.edu.cn (F. Zhou).

https://doi.org/10.1016/j.eswa.2023.121533
Received 25 April 2023; Received in revised form 25 August 2023; Accepted 8 September 2023
Available online 15 September 2023



(a) Decision boundaries decided by the labeled samples.

(b) Decision boundaries decided by labeled and augmented samples.

Fig. 1. An example of the need for sample augmentation. (a) The testing normal samples (red circles) might be *incorrectly* identified as anomalies based on the decision boundaries determined by the limited labeled samples. (b) The data can be *correctly* classified into the normal category based on the boundaries decided by both labeled and augmented samples.

clusters. Due to the lack of labeled training samples, the learned boundaries are difficult to properly discriminate other normal samples (red circles), which can end up with incorrect predictions.

Data augmentation can be adopted to enlarge training data to better reflect the true data distribution. However, current widely used graph data augmentation schemes, such as structure and feature perturbation (Ding, Xu, Tong, & Liu, 2022), principally adopt arbitrary augmentations on the input graph (Ding et al., 2022; Lee, Lee, & Park, 2022), which may cause augmented data to incline towards anomalous samples, leading to suboptimal detection performance. More seriously, imposing some perturbations on nodes, edges or features may alter the graph semantics even if these perturbations are very weak (Lee et al., 2022; Xia, Wu, Chen, Hu, & Li, 2022), which can degrade the detection accuracy.

To address the aforementioned issues, we propose a denoising diffusion probabilistic model (DDPM)-based Controlled GNN (ConGNN) for network anomaly detection. ConGNN can steer a GNN with a newly designed diffusion model to produce effective augmented data, in particular preserving the identification information (e.g., the category label) of a source sample while inclining towards another reference sample. Meanwhile, the augmented data can be flexibly adjusted by changing the reference samples. This can efficiently enhance sample diversity and alleviate the problems of insufficient labeled data. Introducing such augmented data for training can enable the model to learn more accurate decision boundaries for better unknown anomaly detection. Fig. 1(b) illustrates the benefit of our augmented samples in learning the more discriminative boundaries.

Our ConGNN framework mainly consists of three components: a DDPM-based embedding generator, a controlled GNN and a hypersphere-based semi-supervised detection model. The generator, which aims to generate manipulated node embeddings, is incorporated into a GNN to build a controlled GNN for producing augmented data. Coupled with the original data, the augmented data are then utilized to train a hypersphere-based model for anomaly detection. Specifically, inspired by the advantages of DDPMs in producing high-quality controllable images (Nichol & Dhariwal, 2021; Sinha, Song, Meng, & Ermon, 2021; Song, Meng, & Ermon, 2021), we design a DDPM-based embedding generator, which can incorporate the characteristics of the reference node into the source node. This generator is further adopted to steer neighborhood aggregation in GNN to build a controlled GNN. In this controlled GNN, for a target node, its neighbors are first transformed into the ones with the characteristics of the reference node by the generator, and then the transformed neighbors are involved in the neighborhood aggregation to compute the representation of the target node. In this way, the target node representations (augmented data) generated by the controlled GNN can preserve their original category labels while inclining toward the reference nodes. These augmented data can therefore complement the limited labeled samples to enhance data diversity and better cover the real data distribution.

After acquiring augmented data, we apply them to learn a hypersphere-based semi-supervised model for conducting anomaly detection. In particular, we adopt both original and augmented representations to minimize the volume of the hypersphere, where the node embeddings of normal instances are enclosed while anomalous ones are excluded. To ensure the model generalization, we further design a consistency regularization term to force their original and augmented representations to be consistent. The trained hypersphere can easily identify the anomalies by comparing the distance between the node representations and the hypersphere center. The main contributions of this study are:

- We propose a denoising diffusion probabilistic model-based Controlled GNN (ConGNN) for network anomaly detection, which can generate effective augmented data to alleviate the problem of lacking labeled samples.
- We design a graph-specific diffusion model-based generator which can inject the characteristics of a reference node into another source node. This generator is further adopted to steer the procedure of neighborhood aggregation in GNN to build a controlled GNN for generating augmented data.
- We present a data-enclosing hypersphere detection model with our designed consistency regularization term, and experimental results suggest the superior performance of ConGNN over several state-of-the-art baselines.

2. Preliminaries

In this section, we present a preliminary overview of the network anomaly detection problem and denoising diffusion probabilistic models. And these will serve as background or key design ingredients in our ConGNN framework.

2.1. Problem statement

Following the commonly used notations, we use calligraphic fonts, bold lowercase letters, and bold uppercase letters to denote sets (e.g., $\mathcal{V}$), vectors (e.g., $\mathbf{x}$), and matrices (e.g., $\mathbf{X}$). In general, an attributed network can be represented as $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ denotes a set of nodes, $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$ denotes a set of edges, and $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathbb{R}^{n \times h}$ denotes the h-dimensional attributes of n nodes. A binary adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the structural information of the attributed network, where $\mathbf{A}_{i,j} = 1$ if there is a link between nodes v_i and v_j, $\mathbf{A}_{i,j} = 0$ otherwise. Since the information of $\mathcal{V}$ and $\mathcal{E}$ are both contained in $\mathbf{A}$, an attributed network can also be denoted as $G = (\mathbf{A}, \mathbf{X})$. Accordingly, the problem of network anomaly detection on an attributed network is defined as follows:

Problem 1 (*Anomaly Detection on an Attributed network*)**.** Given an attributed network $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ with nodes $\mathcal{V} = \{v_1, \ldots, v_n\}$, we aim to learn a function $f(\cdot)$ using a small number of labeled normal and anomalous samples to calculate the anomaly score for each node. The anomaly score represents the degree of abnormality. By ranking all the nodes with their anomaly scores, the anomaly nodes can be detected according to their positions.

Because of limited labeled samples for model training, the prior distribution, denoted as $p(\mathbf{z})$, on the training data might not match the posterior distribution, $q(\mathbf{z})$, on the test data. Similar to the concept of holes in VAE (Aneja, Schwing, Kautz, & Vahdat, 2021; Sinha et al., 2021), this mismatch could create holes in the prior that the aggregate distribution fails to be covered during training, leading to inferior performance on the test data. Further, this notion can be formalized in the following definition.

Definition 1 (*Prior hole*)**.** Let $p(\mathbf{z})$, $q(\mathbf{z})$ be two distributions. We say that q has an (ϵ, δ)-prior hole with respect to (the prior) p for $\epsilon, \delta \in (0, 1), \delta > \epsilon$, if there exists a set $S \in \text{supp}(p)$, such that $\int_S p(\mathbf{z})\mathrm{d}\mathbf{z} \geq \delta$ and $\int_S q(\mathbf{z})\mathrm{d}\mathbf{z} \leq \epsilon$.

Intuitively, if $q(\mathbf{z})$ has a prior hole with a large δ and a small ϵ, it is very likely that latent variables within the hole have never been seen during training. Hence, the corresponding samples in the test set might not be accurately classified into proper categories. The prior holes are incurred by insufficient labeled data, which might result in overfitting. Taking the scene in Fig. 1 for example, the areas with the red circles can be regarded as the prior holes, because the labeled training samples fail to fully cover these areas. Correspondingly, these test samples can be easily assigned with incorrect labels. Hence, we try to generate complementary augmented samples to alleviate the problem of prior holes and further enhance the detection performance.

2.2. Denoising diffusion probabilistic models

Denoising diffusion probabilistic models (DDPMs) (Ho, Jain, & Abbeel, 2020) have become the state-of-the-art approach for generative modeling with respect to sample quality and mode coverage (Dhariwal & Nichol, 2021; Xiao, Kreis, & Vahdat, 2022). DDPMs consist of a forward diffusion process $q(\mathbf{z}^t|\mathbf{z}^{t-1})$ that gradually corrupts data from some target distribution $q(\mathbf{z}^0)$ into Gaussian noise, and a learned reverse process $p_\theta(\mathbf{z}^{t-1}|\mathbf{z}^t)$ that generates samples by turning noise into samples from $q(\mathbf{z}^0)$.

The forward process is a non-homogeneous Markov chain meaning that the dynamics of the process can be described by the one-step transition density for $t = 1, \ldots, T$.

$$q(\mathbf{z}^t|\mathbf{z}^{t-1}) = \mathcal{N}(\mathbf{z}^t|\mathbf{z}^{t-1}\sqrt{1-\beta_t}, \beta_t\mathbf{I}). \tag{1}$$

The quantity of noise added at each step is defined by a variance schedule $\beta_t \in (0, 1)$. The generative model, parameterized by θ, is the learned reverse process starting with $\mathbf{z}^T \sim \mathcal{N}(0, \mathbf{I})$ and sampling according to

$$p_\theta(\mathbf{z}^{t-1}|\mathbf{z}^t) = \mathcal{N}(\mathbf{z}^{t-1}|\mu_\theta(\mathbf{z}^t, t), \Sigma_\theta(\mathbf{z}^t, t)), \tag{2}$$

where $\Sigma_\theta(\mathbf{z}^t, t)$ can be fixed to a constant, and $\mu_\theta(\mathbf{z}^t, t)$ can be derived from $\epsilon_\theta(\mathbf{z}^t, t)$ (Ho et al., 2020; Sohl-Dickstein, Weiss, Maheswaranathan, & Ganguli, 2015). Here, $\epsilon_\theta(\mathbf{z}^t, t)$ is a model to be trained for predicting noise in $\mathbf{z}^t$.

3. Methodology

In this section, we present the details of our proposed framework, ConGNN. Specifically, we first introduce the DDPM-based embedding generator, which can inject the characteristics of the reference node into the source node. Next, we illustrate how to adopt this generator to steer the procedure of neighborhood aggregation in GNN to construct the controlled GNN, which can generate effective augmented data. Finally, we describe the application of this augmented data in learning a hypersphere-based semi-supervised model for network anomaly detection.

3.1. DDPM-enabled embedding generator

We first present the DDPM-based embedding generator, $\tilde{\mathbf{z}} = G_{\text{ano}}(\mathbf{z}_{\text{src}}, \mathbf{z}_{\text{ref}})$, which takes the source embedding ($\mathbf{z}_{\text{src}}$) and the reference embedding ($\mathbf{z}_{\text{ref}}$) as inputs, and generates an embedding ($\tilde{\mathbf{z}}$) containing the characteristics of both of them. This generator can be easily included in existing GNNs to build a controlled GNN, which can manufacture augmented node representations to alleviate the problem of labeled data deficiency.

Fig. 2 illustrates the framework of the generator, consisting of three components, the autoencoder, the diffusion, and the tailored prior. The autoencoder maps nodes in a graph to latent embeddings. Based on these latent embeddings, the diffusion model learns to generate synthetic node embeddings. However, when taking a general prior such as Gaussian noise as input, the diffusion model can only generate synthetic embeddings similar to the observed nodes' embeddings. Hence, the tailored prior part aims to generate an appropriate prior which involves the characteristics of the reference node. This tailored prior is further fed into the diffusion model for producing a manipulated embedding that possesses the characteristics of both the source node and the reference node. This manipulated embedding will be incorporated into the GNN to produce augmented data.

Autoencoder. The autoencoder with the encoder and decoder aims to learn the embeddings of nodes in a graph for training the diffusion model. The encoder takes the graph as input and embeds it into a group of latent embeddings $\mathbf{Z}$, while the decoder is to reconstruct the graph from the latent embeddings.

The encoder is built with multiple GNN layers that aggregate each node to a low-dimensional latent embedding:

$$\mathbf{H}^1 = \text{GNN}^1(\mathbf{A}, \mathbf{X}), \quad \ldots, \quad \mathbf{Z} = \text{GNN}^L(\mathbf{A}, \mathbf{H}^{L-1}), \tag{3}$$

where GNN^L is the last GNN layer and $\mathbf{Z}$ is the final learned node embeddings from the encoder. Here, the network encoder is compatible with any arbitrary GNN-based architecture (Hamilton, Ying, & Leskovec, 2017; Kipf & Welling, 2017; Veličković et al., 2018; Wu et al., 2019). Similar to the work (Ding, Zhou, Tong, & Liu, 2021; Xiao et al., 2023), we employ simple graph convolution (SGC) (Wu et al., 2019) in our implementation.

After obtaining the embedding $\mathbf{Z}$, the decoder aims to reconstruct the attributed network. Specifically, the decoder takes $\mathbf{Z}$ as input and calculates the inner product to rebuild the adjacency matrix $\widetilde{\mathbf{A}}$:

$$\widetilde{\mathbf{A}} = \text{sigmoid}(\mathbf{Z} \cdot \mathbf{Z}^T). \tag{4}$$

To approximate the original node attributes from the encoded representation, we leverage a simple fully-connected layer to reconstruct the attribute information as follows:

$$\widetilde{\mathbf{X}} = f_{\text{relu}}(\mathbf{Z} \cdot \mathbf{W} + \mathbf{b}), \tag{5}$$

where $\mathbf{W}$ is the weight matrix, and $\mathbf{b}$ is the corresponding bias term. Thus, considering the reconstruction errors, the objective function of the autoencoder can be formulated as:

$$L_{\text{AE}} = (1-\beta)\|\mathbf{A} - \widetilde{\mathbf{A}}\|_F^2 + \beta\|\mathbf{X} - \widetilde{\mathbf{X}}\|_F^2, \tag{6}$$

where $\|\cdot\|_F^2$ refers to the Frobenius norm and β is an important controlling parameter that balances the trade-off between the structure reconstruction and the attribute reconstruction.

Diffusion model. Based on the node embeddings from the encoder, we next train a DDPM-based generator to produce manipulated node embeddings. DDPM is an emerging alternative paradigm for generative modeling (Ho et al., 2020; Nichol & Dhariwal, 2021), which can

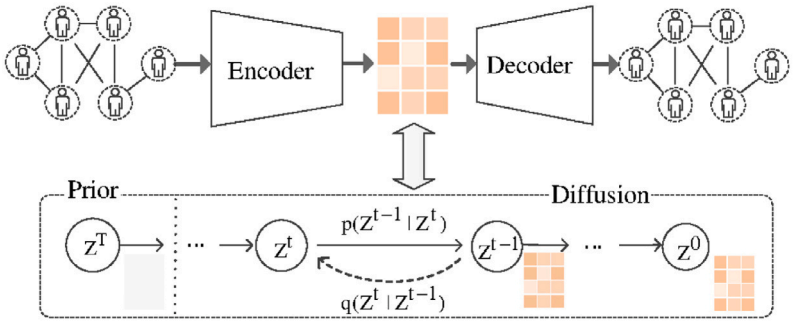


Fig. 2. Framework of the DDPM-based embedding generator.

even outperform the state-of-the-art GAN-based methods for image processing (Dhariwal & Nichol, 2021). DDPM can generate expected data by changing the prior, as it generates data by gradually removing noise in the input prior (Sinha et al., 2021; Xiao, Gou, Tai, Zhang, & Zhou, 2023). Hence, we introduce the DDPM for data generation and try to customize a specific prior for the DDPM to generate the node embeddings that contain the characteristics of both the source node and the reference node.

Formally, we assume that $\mathbf{z}^T \sim p\left(\mathbf{z}^T\right) = \mathcal{N}(0, \boldsymbol{I})$ is the noisy latent variable, and $\mathbf{z}^0$ is the clean latent variable, such as the node embedding obtained from the autoencoder. The generative process can be then defined as:

$$\mathbf{z}^0 \sim p_\theta\left(\mathbf{z}^0 | \mathbf{z}^T\right), \tag{7}$$

where $\mathbf{z}^T$ is the prior distribution for the diffusion model. This generative model takes noise $\mathbf{z}^T$ as input, and produces gradually less-noisy samples until reaching a clean node embedding $\mathbf{z}^0$.

To train this generative model, we adopt the encoder to map nodes in the graph G to latent variables $\mathbf{Z} = \left[\mathbf{z}_1^0, \ldots, \mathbf{z}_n^0\right]$, where n is the number of nodes. Therefore, the objective function of the diffusion model is defined as:

$$L_{\mathrm{DM}} = \mathbb{E}_{t\sim[1,T], \mathbf{z}^0\sim p(\mathbf{z}^0), \epsilon\sim\mathcal{N}(0,\boldsymbol{I})}\left[\|\epsilon - \epsilon_\theta(\mathbf{z}^t, t)\|_2^2\right], \tag{8}$$

where $\mathbf{z}^t = \mathbf{z}^0\sqrt{\bar{\alpha}_t} + \epsilon\sqrt{1-\bar{\alpha}_t}$, $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=0}^{T}\alpha_i$. Here $\beta_t \in (0,1)$ is a variance schedule. $\epsilon_\theta(\mathbf{z}^t, t)$ is the denoising function that estimates the noise vector that is added to $\mathbf{z}^t$. Once trained, the diffusion model can acquire clean node embeddings through a gradual denoising process.

Tailored prior. When taking random noise $\mathbf{z}^T \sim \mathcal{N}(0, \boldsymbol{I})$ as input, the diffusion model can only produce embeddings similar to the observed ones. However, to better cover the data distribution, our method requires the complementary nodes, which can fill the gap among limited training samples. To this end, we tailor a prior for the diffusion model to generate augmented embeddings, which involve the characteristics of both the source node and the reference one.

Specifically, we introduce the energy-based model (Arbel, Zhou, & Gretton, 2021; Du & Mordatch, 2019) to produce this prior. The energy-based model is a form of generative model, which can learn an optimal function to describe a given system or target distribution. We consider the closeness between the latent variables and the classifier likelihood to define the energy-based model. Here the closeness between the latent variables forces the generated embedding to favor closeness to the reference node, and the classifier likelihood ensures that the generated embedding belongs to the category of the source node. As a result, the generated prior can favor closeness to the reference embedding, $\mathbf{z}_{\mathrm{ref}}$, and preserve the label of the source node, $\mathbf{z}_{\mathrm{src}}$. Here, we compute the closeness via the L2 distance between the latent variable $\mathbf{z}_{\mathrm{ref}}$ and the manipulated latent embedding. Then, the energy-based model can be obtained by multiplying this distance with the classifier likelihood:

$$p_\phi(\hat{\mathbf{z}}) = \exp\left(-r_\psi(y|\hat{\mathbf{z}})\right) \cdot \|\mathbf{z}_{\mathrm{ref}} - \hat{\mathbf{z}}\|_2^2, \tag{9}$$

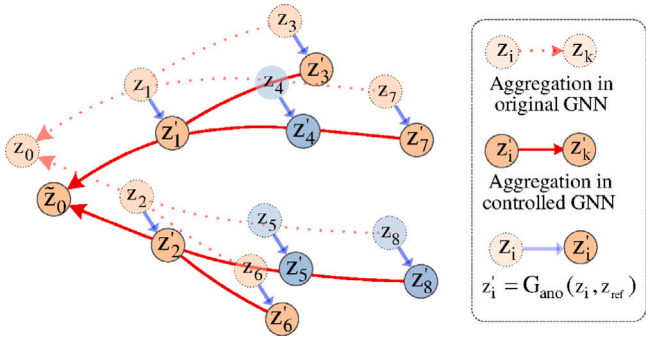


Fig. 3. Original GNN vs. controlled GNN.

where $r_\psi()$ is a classifier with y being the class label. If the source sample is labeled data, y refers to the label of $\mathbf{z}_{\mathrm{src}}$, and $r_\psi()$ can be trained using the labeled data. If the source sample does not have a label (i.e., we merely want to generate new samples similar to the given ones), we can train a positive-unlabeled classifier (Elkan & Noto, 2008), where we assign positive to a few of the most similar samples and unlabeled to other samples.

Then, like Langevin dynamics (Neal et al., 2011), we approximately draw samples from this energy by taking a gradient step:

$$\bar{\mathbf{z}} \leftarrow \hat{\mathbf{z}} + \eta\nabla_{\mathbf{z}} r_\psi(y|\mathbf{z})|_{\mathbf{z}=\hat{\mathbf{z}}}, \tag{10}$$

where hyper-parameter $\eta > 0$ denotes the step size. And then we add noise into it to generate the prior:

$$\mathbf{z}^T \leftarrow \sqrt{\alpha}\bar{\mathbf{z}} + \sqrt{1-\alpha}\epsilon, \tag{11}$$

where $\alpha \in (0,1)$ indicates the diffusion noise magnitude. This prior $\mathbf{z}^T$ keeps the key features of $\mathbf{z}_{\mathrm{scr}}$ invariant, but involves the characteristics of $\mathbf{z}_{\mathrm{ref}}$. The prior will be fed into the diffusion model to generate manipulated embeddings.

3.2. Controlled graph neural networks

Based on the embedding generator, we now build a Controlled GNN to generate augmented representations that can keep the identification information of the original node invariant and slightly incline towards the reference node. These augmented representations will be adopted to train a hypersphere-based semi-supervised detection model. The main idea is that during the procedure of neighborhood aggregation in the GNN, we utilize the DDPM-based generator to inject the characteristics of the reference node into each neighbor so that the target node representation can keep the identification information of the original node invariant and gently incline toward the reference node.

We denote the original GNN as $\mathrm{GNN}^{\mathrm{ori}}$, and the controlled GNN as $\mathrm{GNN}^{\mathrm{con}}$. The procedure of their neighbor aggregation is presented in

Fig. 3. For the original GNN, the target node representation, $\mathbf{z}_0$, is obtained by aggregating the embeddings of its neighbors directly. Instead, for the controlled GNN we proposed, the augmented representation, $\tilde{\mathbf{z}}_0$, is obtained by aggregating the manipulated neighbor embeddings. Here, the embedding of each neighbor $\mathbf{z}_i$ is regarded as the source one, and the reference node $\mathbf{z}_{\text{ref}}$ is chosen from a cluster different from that of the target node $\mathbf{z}_0$. The neighbor node and the reference node are fed into the DDPM-enabled generator $G_{\text{ano}}(\mathbf{z}_i, \mathbf{z}_{\text{ref}})$ to produce the manipulated node $\mathbf{z}_i'$, which is further involved in the neighborhood aggregation in the GNN to produce augmented data $\tilde{\mathbf{z}}_0$. In this way, the augmented representation, $\tilde{\mathbf{z}}_0$, can keep the identification information of its category while inclining towards the reference embedding $\mathbf{z}_{\text{ref}}$. These augmented representations can be adopted to alleviate the prior hole and enhance detection performance. A visible example is shown in Fig. 1(b). In this figure, assuming blue circles as the original embeddings and purple circles as the reference embeddings, the augmented embeddings (blue star) will lie between the blue and purple circles. These augmented embeddings have the same label as the blue circles while gently inclining towards the purple circles, which can be adopted to bridge the gap between blue and purple circles and enhance model robustness.

Specifically, we design ConGCN based on the graph attention networks (GATs) (Veličković et al., 2018), which introduces masked the head attention mechanism to represent the importance of different adjacent nodes. Note that other GNN-based architectures (Hamilton et al., 2017; Kipf & Welling, 2017; Wu et al., 2019), can also be adopted to build the controlled GNN. We employ GATs in the implementation because adding the attention mechanism can generally advocate performance.

Original GNN. Formally, in each layer $l-1$, node v_i integrates the features of neighboring nodes to obtain representations of layer l via

$$h_i^{(l)} = \sigma\left(\sum_{j\in\mathcal{V}_i\cup\{v_i\}} a_{ij}\mathbf{W}\cdot\mathbf{h}_j^{(l-1)}\right), \tag{12}$$

where σ refers to a nonlinear activation function (e.g., ReLU), $\mathcal{V}_i$ is the set of neighbors for v_i, and a_{ij} represents the attention coefficient between node v_i and node v_j, which can be computed as:

$$a_{ij} = \frac{\exp(\sigma(\mathrm{a}^{\mathrm{T}}[\mathbf{W}\mathbf{h}_i^{(l)} \oplus \mathbf{W}\mathbf{h}_j^{(l)}]))}{\sum_{k\in\mathcal{V}_i\cup\{v_i\}}\exp(\sigma(a^{\mathrm{T}}[\mathbf{W}\mathbf{h}_i^{(l)} \oplus \mathbf{W}\mathbf{h}_k^{(l)}]))}, \tag{13}$$

where $\oplus$ is the concatenation operation and attention vector a is a trainable weight vector that assigns importance to different neighbors of node v_i, allowing the model to highlight the features of the important neighboring node that is more task-relevant.

To incorporate high-order neighborhood, multiple GAT layers are adopted to build the graph attentive encoder.

$$\begin{aligned}
&\mathbf{h}_i^{(1)} = \sigma\left(\sum_{j\in\mathcal{V}_i\cup\{v_i\}} a_{ij}^{(1)}\mathbf{W}^{(1)}\cdot\mathbf{x}_j\right),\\
&\quad\cdots\cdots\\
&\mathbf{z}_i = \sigma\left(\sum_{j\in\mathcal{V}_i\cup\{v_i\}} a_{ij}^{(L)}\mathbf{W}^{(L)}\cdot\mathbf{h}_j^{(l-1)}\right),
\end{aligned} \tag{14}$$

where $\mathbf{z}_i$ is the latent representation of node v_i. In this way, the graph attentive encoder is able to map the learned node representations by capturing the nonlinearity of topological structure and nodal attributes.

Controlled GNN. To manipulate node representations, we steer the procedure of neighborhood aggregation to generate the augmented representations which possess the identification information of a given category, as well as the characteristics of the reference node. Then, the controlled aggregation procedure is defined as:

$$\begin{aligned}
&\mathbf{h}_i^{(1)} = \sigma\left(\sum_{j\in\mathcal{V}_i\cup\{v_i\}} a_{ij}^{(1)}\mathbf{W}^{(1)}\cdot\mathbf{x}_j\right),\\
&\tilde{\mathbf{h}}_i^{(1)} = G_{\text{ano}}(\mathbf{h}_i^{(1)}, \mathbf{z}_{ref}),\\
&\quad\cdots\cdots\\
&\tilde{\mathbf{z}}_i = \sigma\left(\sum_{j\in\mathcal{V}_i\cup\{v_i\}} a_{ij}^{(L)}\mathbf{W}^{(L)}\cdot\tilde{\mathbf{h}}_j^{(l-1)}\right),
\end{aligned} \tag{15}$$

where $G_{\text{ano}}(\mathbf{h}_i^{(1)}, \mathbf{z}_{ref})$ represents the function of the DDPM-based generator, which generates a manipulated embedding preserving the category label of $\mathbf{h}_i^{(1)}$ while containing the characteristics of $\mathbf{z}_{\text{ref}}$. The generated representation $\tilde{\mathbf{z}}_i$ will be used as augmented data to train a hypersphere-based model for anomaly detection.

3.3. Hypersphere learning

After obtaining node representations by GNN^{ori} and GNN^{con}, we adopt them to learn a hypersphere-based semi-supervised model for anomaly detection. Hypersphere learning aims to enclose normal nodes to further discriminate anomalies in the latent space (Ruff et al., 2020; Wang et al., 2021). We utilize a small number of labeled data and a large number of unlabeled data to train the hypersphere-based model. For labeled data, we adopt their original representations and augmented representations produced by GNN^{ori} and GNN^{con}, respectively, to build two supervised losses, which aim to enclose normal nodes within the corresponding hypersphere while pushing anomalies away from the normal hyperspheres and minimizing the volume of the data-enclosing hyperspheres. For unlabeled data, we utilize their original representations and augmented representations to construct a consistency loss which forces the distance between the original representations and center c to keep consistent with the distance between the augmented representations and c. Here the hypersphere centers c can be obtained by averaging the node representations.

Supervised losses. For the labeled node set $\mathcal{V}^l = \{v_1^l, \ldots, v_m^l\}$ with the label set $\mathcal{Y} = \{y_1, \ldots, y_m\}$, we adopt the original GNN (GNN^{ori}) to transform the nodes in $\mathcal{V}^l$ into latent representations $\mathbf{Z}^l = \{\mathbf{z}_1^l, \ldots, \mathbf{z}_m^l\}$ and build a supervised loss:

$$\mathbf{Z}^l = \text{GNN}^{\text{ori}}(\mathcal{V}^l), \tag{16}$$

$$\mathcal{L}_{\text{lab}}^{\text{ori}} = \frac{\lambda_0}{m}\sum_{j=1}^{m}(\|\mathbf{z}_j^l - c\|^2)^{y_j} + \frac{\lambda_1}{2}\sum_{l=1}^{L}\|\mathbf{W}^l\|_F^2, \tag{17}$$

where c is a predetermined center point, $\mathbf{W}$ refers to the weights in the original GNN, and λ_0 and λ_1 are hyper-parameters to adjust the importance. For labeled normal nodes ($y_j = +1$), we impose a quadratic loss on the distances of the mapped points to the center c, thus intending to overall learn a latent distribution which concentrates the normal data. For the labeled anomalies ($y_j = -1$) in contrast, we penalize the inverse of the distances such that anomalies must be mapped further away from the center.

Further, the nodes in $\mathcal{V}^l$ are transformed into augmented representations $\widetilde{\mathbf{Z}}^l = \{\tilde{\mathbf{z}}_1^l, \ldots, \tilde{\mathbf{z}}_m^l\}$ by the controlled GNN (GNN^{con}). Similarity, we utilize $\widetilde{\mathbf{Z}}^l$ to build a supervised loss:

$$\widetilde{\mathbf{Z}}^l = \text{GNN}^{\text{con}}(\mathcal{V}^l), \tag{18}$$

$$\mathcal{L}_{\text{lab}}^{\text{con}} = \frac{\lambda_2}{m}\sum_{j=1}^{m}(\|\widetilde{\mathbf{Z}}^l - c\|^2)^{y_j}, \tag{19}$$

where c, λ_2 and y_j mean the same with Eq. (17).

Consistency loss. For the unlabeled data, we design a consistency regularization term about sample distances to enhance model robustness. The main idea is that the distance between the augmented representation and center point c should keep consistent with the distance between the original representation and c. This regularization

Table 1
Statistics of datasets.

Datasets	# node	# edge	# feature	# anomaly	# class
Cora	2,708	5,429	1,433	418	7
Citeseer	3,327	4,732	3,703	264	6
PubMed	19,717	44,338	500	4,103	3
Photo	7,487	119,043	745	696	8
Computer	13,381	245,778	767	580	10

term tries to keep both distances invariant, which can help the model capture the characteristics of unlabeled data and further enhance the generalization capacity of the model.

Let $\mathcal{V}^u = \{v_1^u, \ldots, v_n^u\}$ be the unlabeled node set. The corresponding representations $\mathbf{Z}^u = \{\mathbf{z}_1^u, \ldots, \mathbf{z}_n^u\}$ and augmented representations $\widetilde{\mathbf{Z}}^u = \{\tilde{\mathbf{z}}_1^u, \ldots, \tilde{\mathbf{z}}_n^u\}$ are produced by GNN^{ori} and GNN^{con}, respectively. We construct the consistency loss to force the distance between the augmented representations and c to be close to that of the original representations and c:

$$\mathcal{L}_{\text{unl}}^{\text{sim}} = \frac{\lambda_3}{n} \sum_{j=1}^{n} \left| \|\mathbf{z}_j^u - c\|^2 - \|\tilde{\mathbf{z}}_j^u - c\|^2 \right|. \tag{20}$$

Coupled with the supervised losses in Eq. (17) and (19), the final loss function of the model is illustrated as:

$$L = \mathcal{L}_{\text{lab}}^{\text{ori}} + \mathcal{L}_{\text{lab}}^{\text{con}} + \mathcal{L}_{\text{unl}}^{\text{sim}}. \tag{21}$$

After model training, each test sample v_i will be mapped to the representation by the original GNN, and the anomaly score is defined as the Euclidean distance between its representation and the hypersphere center c in the latent space: $f(v_i) = \|\mathbf{z}_i - c\|_2^2$, where $\mathbf{z}_i$ is the representation of v_i.

4. Experimental evaluation

In this section, we conduct empirical evaluations to demonstrate the effectiveness of our method in terms of anomaly detection performance, data efficiency, and the role of graph data augmentation methods and detection models.

4.1. Experimental settings

Datasets. We employ five real-world attributed networks: Cora, CiteSeer, PubMed, Amazon-Photo, and Amazon-Computer (McAuley, Targett, Shi, & Van Den Hengel, 2015; Sen et al., 2008) for performance evaluation. Their descriptive statistics are presented in Table 1. The first three datasets are citation networks, where the node and edge denote the scientific publication and citation connections between two publications, respectively. Amazon-Photo and Amazon-Computer are from the Amazon co-purchase graphs where nodes refer to products and edges indicate the co-purchase behavior. For all datasets, the attribute of each node is described by a bag-of-word attribute feature vector. Although these datasets have several classes, we formulate a binary class problem for each dataset by regarding the smallest class as anomalous and the remaining classes as normal following the works (Kumagai, Iwata, & Fujiwara, 2019; Zhou et al., 2022). For each dataset, we select 5% of normal nodes and 20 anomalies as labeled data and 55% of all instances as unlabeled data for model training. We use 20% of the data for validation and the remaining for testing.

Baselines. We compared the proposed framework with nine baselines with different techniques, including unsupervised approaches (Ding et al., 2019; Liu et al., 2021; Zhou & Paffenroth, 2017), few-shot learning-based methods (Ding, Wang, Caverlee, & Liu, 2021; Ding, Zhou, et al., 2021), one-class and semi-supervised models (Kumagai et al., 2021; Ruff et al., 2020; Wang et al., 2021; Zhou et al., 2022). *Autoencoder* (Zhou & Paffenroth, 2017) is a feature-based unsupervised deep autoencoder model which introduces an anomaly regularizing penalty based upon L1 or L2 norms. *DOMINANT* (Ding et al., 2019) is an unsupervised anomaly detection framework, which utilizes a graph convolution autoencoder to reconstruct the adjacency matrix and attribute matrix. Then, the abnormality score is calculated by the reconstruction error. *CoLA* (Liu et al., 2021) is a contrastive learning-based approach, which adopts subgraph sampling to augment graph data and build a contrastive loss for anomaly detection. *GDN* (Ding, Zhou, et al., 2021) is a graph convolutional network-based method that leverages a few labeled anomalies for anomaly detection. *Meta-PN* (Ding, Wang, et al., 2021) is a graph meta-learning framework for node classification, which builds a meta-learned label propagation strategy to generate pseudo labels for enlarging insufficient labeled data. *OC-GNN* (Wang et al., 2021) is a one-class method which combines GNNs with the one-class objective for anomaly detection. *DeepSAD* (Ruff et al., 2020) is a deep learning approach for general semi-supervised anomaly detection. In our experiment, we leverage node attributes as input features. *GCN-AN* (Kumagai et al., 2021) is a semi-supervised detection method, which identifies anomalies by adopting a few labeled normal and abnormal instances. *MHGL* (Zhou et al., 2022) is a multi-hypersphere graph learning framework to identify both seen and unseen anomalies.

Experiment Setup. For the generator, we use 1000 diffusion steps for the DDPM as this value can obtain a more stable result, and train a linear model over the latent variables as $r_\psi(y|\hat{\mathbf{z}})$ since it is computationally effective. For the selection of the reference node, we adopt K-means to divide nodes into different clusters, and select the source node and the reference node from different clusters. Here K denotes the number of the clusters, which is set to 6, 6, 3, 8 and 9 for Cora, Citeseer, PubMed, Photo and Computer datasets, individually. The hyper-parameter η in Eq. (10) is empirically set to 10. For the hypersphere-based detection model, we set $\lambda_0 = 0.45$, $\lambda_1 = 0.25$, $\lambda_2 = 0.14$ and $\lambda_3 = 0.16$ in Eq. (17), Eq. (19) and Eq. (20) since these values can obtain a better result. All these hyper-parameters are tuned based on the validation set. Following previous works (Ding, Zhou, et al., 2021; Zhou et al., 2022), we utilize the two standard evaluation metrics (AUC and Precision@K) to measure the effectiveness of all the methods[1].

4.2. Detection performance comparisons

The anomaly detection results between our proposed method and baselines are reported in Table 2 regarding AUC and Precision@K (P@K), where the percentages in the last row refer to improvement rates of our model over the best baseline, MHGL. From the table, we have the following observations. For overall detection results, our proposed ConGNN model yields better performance than these baseline methods across the five datasets. In particular, ConGNN achieves on average 1.32%, 3.63% and 3.41% improvement over the best baseline, MHGL, in terms of AUC, P@50 and P@100, respectively. Besides, ConGNN surpasses baselines by a larger margin on Cora. The reason is that Cora has a relatively small number of nodes, which are inadequate for training robust baseline models. In contrast, our model can generate augmented samples to complement limited data and therefore exhibit distinct superiority.

For unsupervised methods, CoLA achieves better performance than Autoencoder and DOMINANT, which proves the superiority of graph data augmentation and contrastive learning. However, the few-shot learning-based model, such as GDN, significantly outperforms these unsupervised methods. This result indicates that leveraging labeled samples can efficiently advance the performance of anomaly detection, and more improvements in detection performance can be obtained at the cost of collecting more labeled samples.

The semi-supervised models GCN-AN and MHGL, which leverage both normal and abnormal data for training, have acquired great performances and significantly outperform OCGNN — which only learns

[1] Codes are available at https://github.com/ChunjingXiao/ConGNN

Table 2
Performance comparison between ConGNN and baselines on five datasets.

Method	Cora			CiteSeer			PubMed			Photo			Computer		
	AUC	P@50	P@100	AUC	P@50	P@100	AUC	P@50	P@100	AUC	P@50	P@100	AUC	P@50	P@100
Autoencoder	0.510	19.900	22.137	0.613	28.012	30.820	0.540	46.035	46.682	0.412	28.543	29.892	0.501	42.687	43.287
DOMINANT	0.523	20.407	23.701	0.631	29.759	31.642	0.509	43.283	43.891	0.381	26.395	27.643	0.468	39.876	40.435
CoLA	0.563	21.969	24.438	0.631	31.258	34.392	0.599	51.048	51.765	0.448	31.070	32.538	0.551	46.936	47.596
GDN	0.641	25.087	27.564	0.734	27.834	30.069	0.912	77.724	79.187	0.743	48.262	50.442	0.935	65.251	67.352
Meta-PN	0.801	31.726	34.594	0.764	33.672	36.614	0.763	65.927	66.683	0.831	57.541	60.124	0.911	77.225	79.212
OC-GNN	0.830	32.071	36.023	0.782	38.561	43.054	0.719	63.636	62.716	0.628	43.781	47.183	0.850	73.371	75.183
DeepSAD	0.631	24.791	27.397	0.764	33.964	36.761	0.817	69.564	70.726	0.660	34.826	45.112	0.778	67.751	65.221
GCN-AN	0.860	33.084	37.912	0.801	37.983	42.691	0.941	78.421	79.568	0.832	57.432	60.183	0.944	80.525	81.574
MHGL	0.867	36.196	40.076	0.826	43.941	48.169	0.954	81.221	82.441	0.830	57.891	60.273	0.975	87.685	84.240
ConGNN	**0.881**	**39.332**	**42.411**	**0.833**	**46.441**	**50.902**	**0.962**	**81.965**	**83.117**	**0.851**	**58.465**	**61.559**	**0.983**	**89.351**	**86.446**
(improvement↑)	(1.6%)	(8.7%)	(5.8%)	(0.8%)	(5.7%)	(5.7%)	(0.8%)	(0.9%)	(0.8%)	(2.5%)	(1.0%)	(2.1%)	(0.8%)	(1.9%)	(2.6%)

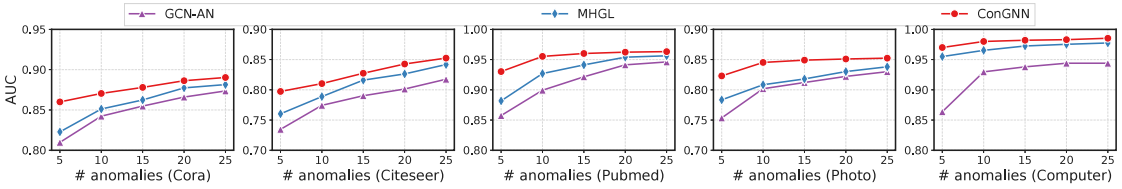


Fig. 4. AUC Performance *w.r.t.* using different numbers of labeled anomalies.

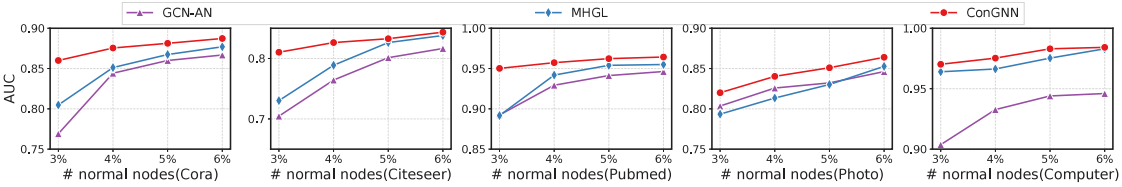


Fig. 5. AUC Performance *w.r.t.* different ratios of labeled normal samples.

normal patterns to discriminate anomalies. This result demonstrates that exploiting supervised information of both annotated normal and anomalous samples is an efficient way of detecting anomalies. Meanwhile, by adding augmented samples, ConGNN is capable of complementing the training data and thus outperforms these semi-supervised baselines by a large margin.

4.3. Data efficiency

Here we observe the efficiency of labeled data for our model and the baselines. It is essential to effectively utilize labeled data for practical anomaly detection applications as obtaining numerous labels is costly and time-consuming. We investigate the impact on detection performance when varying labeled anomalous data and labeled normal data. For labeled anomalies, we vary their number from 5 to 25 and freeze the labeled normal nodes at 5%. For labeled normal nodes, we change their amount from 3% to 6% and fix labeled anomalies as 20. For all these experiments, the testing data remains unchanged. We select the two recent semi-supervised baselines (GCN-AN and MHGL) for comparison since they are specifically designed for exploiting the value of both labeled and unlabeled data to conduct network anomaly detection.

Figs. 4 and 5 report the impact of labeled anomalous and normal nodes, respectively. As shown in Fig. 4, these two semi-supervised baselines and our model achieve increasing detection accuracy along with the rise of anomalies. While our designed ConGNN is the most data-efficient method, which achieves the best average performance *w.r.t.* the different number of labeled anomalies. Notably, compared with the baselines, ConGNN achieves a larger gain in terms of AUC when there are fewer labeled anomalies. This is because ConGNN can generate augmented samples for model training, which is especially effective under the scenario of insufficient labeled samples. For labeled normal data shown in Fig. 5, our model ConGNN achieves comparatively steady performance when the sample number varies for PubMed. For example, its AUC using 3% of normal data is quite close to that using 6% of normal data. The reason is that PubMed has a large number of normal nodes and coupled with the augmented data generated by our model, 3% of them is adequate for obtaining better performance.

4.4. Role of graph data augmentation methods

We here compare the efficiency of our model with widely used graph data augmentation methods, i.e., we freeze the hypersphere-based detection model but change the augmented data produced by different augmentation methods. (1) *Basic*, which excludes the augmented data from the input data of the detection model. (2) *Edge*, which adopts the edge operation approach illustrated in the work (Zhu et al., 2021) to manufacture augmented data. (3) *Feature*, which utilizes the feature augmentation method presented in the work (Liu et al., 2022) to produce augmented data. (4) *ConFinal*, which excludes the manipulated neighborhood aggregation, and uses the embedding generator to manipulate the final node representations obtained by the original neighborhood aggregation to generate augmented data.

The detection performance of different data augmentation methods under the five datasets is illustrated in Fig. 6. We summarize the observations reported in this figure as follows. First, our model *ConGNN* surpasses other graph data augmentation methods. Since these results are inferred using the same detection model, the improvement of our model is primarily attributed to the effective augmented data. Our model can tailor augmented data to alleviate the prior hole and better cover the data distribution, and thus achieve better results. Second, after removing the augmented data, *Basic* performs the worst. While, both

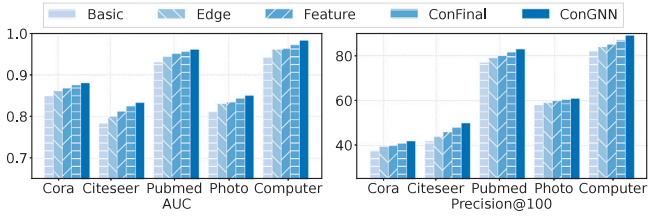
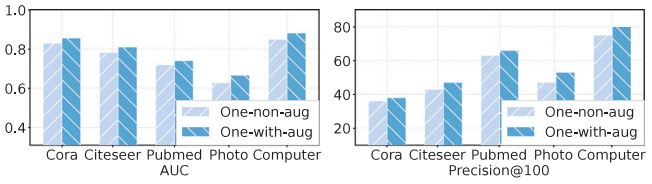
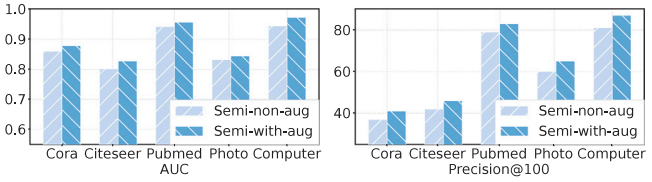


Fig. 6. Performance of different data augmentation methods.



(a) Detection performance using one-class model.



(b) Detection performance using semi-supervised model.

Fig. 7. The effect of the augmented data.

Edge and *Feature* acquire higher results than *Basic*, which demonstrates that utilizing augmented data is an effective way to improve the performance of network anomaly detection. Third, *ConFinal* performs better than *Edge* and *Feature*, indicating that the augmented data generated by manipulating the final node representation can effectively enhance detection performance. However, *ConFinal* obtains worse results than *ConGNN*. This suggests that manipulating neighborhood aggregation can generate smoother augmented data and further boost the detection performance.

4.5. Role of detection models

The analyses in the previous section suggest that ConGNN obtains remarkable gains when applying the augmented data to the hypersphere-based model. Here we further investigate the effectiveness of the augmented data when applying them to other detection models: one-class detection model (Wang et al., 2021) and semi-supervised detection method (Kumagai et al., 2021). We evaluate the model performance with/without the augmented data for model training, named as $\mathcal{X}$-with-aug/$\mathcal{X}$-non-aug, where $\mathcal{X}$ denotes the one-class (one) or semi-supervised (semi) model.

Fig. 7 shows the AUC and Precision@100 for the one-class model and semi-supervised model with/without the augmented data generated by our designed ConGNN. As shown in Fig. 7(a), the performance of one-with-aug significantly outperforms that of one-non-aug for all five datasets. For example, the AUC of one-with-aug on the Cora dataset is around 3% higher than that of one-non-aug. The same trends are observed for the semi-supervised method in 7(b). The results suggest that the models using the augmented data generated by our method always surpass the ones without the augmented data, and our model can be applied to different anomaly detection models.

5. Related work

In this section, we review the related work in terms of *network anomaly detection*, *graph data augmentation* and *diffusion probabilistic models*.

Network anomaly detection. Due to the advance of deep learning, GNN-based architecture has become a mainstream technique for anomaly detection on attributed graph (Caville, Lo, Layeghy, & Portmann, 2022; Pang et al., 2021; Van Belle, Van Damme, Tytgat, & De Weerdt, 2022). Initially, unsupervised models with GNNs are used for label-agnostic anomaly detection because of the high cost of collecting numerous labeled samples. These models generally utilize a GNN-based autoencoder to reconstruct the attribute and structure information simultaneously, and the abnormality is evaluated by reconstruction errors (Antwarg, Miller, Shapira, & Rokach, 2021; Ding et al., 2019). On the basis of this framework, a tailored deep graph convolutional network is designed to detect local, global, and structure anomalies by capturing community structure in the graph (Luo et al., 2022), and a deep attention mechanism is incorporated into the graph deep autoencoder for anomaly detection in multi-attributed networks (Shao et al., 2023). Further, contrastive learning and self-supervised learning are introduced to identify the anomalies in attributed networks (Zhang, Wang, & Chen, 2022; Zheng et al., 2021).

To further improve performance, labeled samples have been exploited to design semi-supervised models for anomaly detection (Villa-Pérez et al., 2021). In this field, a graph deviation network is designed to leverage a few labeled anomalies to enforce statistically significant deviations between abnormal and normal nodes and further improve detection performance (Ding, Zhou, et al., 2021; Pang, Shen, & van den Hengel, 2019). Also, hypersphere-based methodologies are proposed to explore labeled normal and anomalous samples for semi-supervised anomaly detection (Kumagai et al., 2021; Ruff et al., 2020). Further, a multi-hypersphere graph learning approach is designed to effectively leverage existing labels by learning fine-grained normal patterns to distinguish both seen and unseen anomalies (Zhou et al., 2022). Our model is also a hypersphere-based semi-supervised detection method. While we mainly focus on designing the controlled GNN to generate effective augmented data to improve detection performance.

Graph data augmentation. Graph data augmentation techniques aim to generate extra data by applying label-preserving transformations to inputs for improving model generalization (Ding et al., 2022; Wu et al., 2022). A widely used scheme for graph data augmentation is related to edge operations, such as edge dropping and subgraph sampling (Ding et al., 2022; Xian et al., 2021; Zhao et al., 2021). Initially, randomly dropping a fixed fraction of edges is adopted to generate new graph views as data augmentation for node classification (Rong, Huang, Xu, & Huang, 2020; You et al., 2020). Following this, task-irrelevant edges are identified and removed by the MLP-based graph sparsification model (Zheng et al., 2020) and the nuclear norm regularization loss (Luo et al., 2021) to improve the generalization performance. These augmentation methods are also applied to contrastive learning and self-supervised learning for node classification and anomaly detection (Liu et al., 2021; Zhu et al., 2021).

Recently, feature augmentation has been adopted for graph data augmentation, which aims to improve the node feature quality by learning additional task-relevant features. Feature augmentation is generally utilized to initiate node features on plain graphs to smoothly incorporate into graph neural network models and supplement additional node features which are hard to be captured by downstream models (Kong et al., 2022; Liu et al., 2022). Our model can be regarded as a kind of feature augmentation method. Different from these works, we introduce diffusion models to steer the GNN to generate augmented data which can preserve original labels while being close to another sample. In other words, our model can produce tailored augment data to effectively alleviate the problem of limited labeled data.

Diffusion probabilistic models. Diffusion probabilistic models aim to generate high-quality data by reversing the diffusion process using a Markov chain with discrete timesteps (Ho et al., 2020). The models have acquired state-of-the-art generation instances for different applications. Image generation models are one of the major applications where diffusion models are exploited to produce high-quality images (Meng et al., 2022; Song, Sohl-Dickstein, et al., 2021), speed up the sampling process (Dockhorn, Vahdat, & Kreis, 2022; Xiao et al., 2022), conduct image super-resolution (Li et al., 2022; Saharia et al., 2022) and image-to-image translation (Sinha et al., 2021). Waveform synthesis is another main application where the diffusion model is utilized to manufacture time-domain speech audio from the prior noise (Chen et al., 2021; Lam, Wang, Su, & Yu, 2022). Also, diffusion models have been applied to voice conversion (Popov et al., 2022), shape generation (Zhou, Du, & Wu, 2021) and time series forecasting (Rasul, Seward, Schuster, & Vollgraf, 2021).

Compared with these works, we extend the diffusion models to the problem of anomaly detection on attributed graphs. Due to the inherent differences between the graph data structures and the irregular data types like images and waveforms, we present a graph-specific data generator with a tailed prior and adopt the generator to steer the neighborhood aggregation process of the GNN to generate augmented data. We further design an augmented data-based consistency regularization term for the hypersphere-based semi-supervised model to perform anomaly detection.

6. Conclusions

In this paper, we proposed a denoising diffusion probabilistic model-based Controlled GNN (ConGNN), which can more accurately conduct anomaly detection with limited labeled samples. We designed a graph-specific diffusion model-based generator with the tailored prior, which can inject the characteristic of a node into another one. This generator is adopted to steer the procedure of neighborhood aggregation in GNNs to produce effective augmented data, which can bridge the gap between limited training data. We further apply the augmented data to a hypersphere-based semi-supervised model for anomaly detection. The experiments conducted on five real-world datasets demonstrate that our proposed approach achieves state-of-the-art performance.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 62176043 and 62072077), Natural Science Foundation of Sichuan Province (Grant No. 2022NSFSC0505).

CRediT authorship contribution statement

Xuan Li: Software, Methodology, Resources, Writing – original draft. **Chunjing Xiao:** Conceptualization, Methodology, Writing – original draft. **Ziliang Feng:** Investigation, Writing – review & editing. **Shikang Pang:** Software, Validation, Resources, Visualization. **Wenxin Tai:** Software, Validation, Data curation. **Fan Zhou:** Writing – review & editing, Resources, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Aneja, J., Schwing, A., Kautz, J., & Vahdat, A. (2021). A contrastive learning approach for training variational autoencoder priors. *Advances in Neural Information Processing Systems*, 480–493.

Antwarg, L., Miller, R. M., Shapira, B., & Rokach, L. (2021). Explaining anomalies detected by autoencoders using Shapley Additive Explanations. *Expert Systems with Applications*, *186*, Article 115736.

Arbel, M., Zhou, L., & Gretton, A. (2021). Generalized energy based models. In *International conference on learning representations*.

Bandyopadhyay, S., Lokesh, N., & Murty, M. N. (2019). Outlier aware network embedding for attributed networks. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 12–19).

Caville, E., Lo, W. W., Layeghy, S., & Portmann, M. (2022). Anomal-E: A self-supervised network intrusion detection system based on graph neural networks. *Knowledge-Based Systems*, *258*, Article 110030.

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., & Chan, W. (2021). WaveGrad: Estimating gradients for waveform generation. In *International conference on learning representations*.

Dhariwal, P., & Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 8780–8794.

Ding, K., Li, J., Bhanushali, R., & Liu, H. (2019). Deep anomaly detection on attributed networks. In *Proceedings of the SIAM international conference on data mining* (pp. 594–602).

Ding, K., Wang, J., Caverlee, J., & Liu, H. (2021). Meta propagation networks for graph few-shot semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*.

Ding, K., Xu, Z., Tong, H., & Liu, H. (2022). Data augmentation for deep graph learning: A survey. *ACM SIGKDD Explorations Newsletter*, *24*(2), 61–77.

Ding, K., Zhou, Q., Tong, H., & Liu, H. (2021). Few-shot network anomaly detection via cross-network meta-learning. In *Proceedings of the web conference* (pp. 2448–2456).

Dockhorn, T., Vahdat, A., & Kreis, K. (2022). Score-based generative modeling with critically-damped langevin diffusion. In *International conference on learning representations*.

Du, Y., & Mordatch, I. (2019). Implicit generation and modeling with energy based models. In *Advances in neural information processing systems*.

Elkan, C., & Noto, K. (2008). Learning classifiers from only positive and unlabeled data. In *Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining* (pp. 213–220).

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in neural information processing systems* (pp. 1025–1035).

Hilal, W., Gadsden, S. A., & Yawney, J. (2022). Financial fraud: A review of anomaly detection techniques and recent advances. *Expert Systems with Applications*, *193*, Article 116429.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840–6851.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International conference on learning representations*.

Kong, K., Li, G., Ding, M., Wu, Z., Zhu, C., Ghanem, B., et al. (2022). Robust optimization as data augmentation for large-scale graphs. In *IEEE/CVF conference on computer vision and pattern recognition* (pp. 60–69).

Kumagai, A., Iwata, T., & Fujiwara, Y. (2019). Transfer anomaly detection by inferring latent domain representations. *Advances in Neural Information Processing Systems*.

Kumagai, A., Iwata, T., & Fujiwara, Y. (2021). Semi-supervised anomaly detection on attributed graphs. In *International joint conference on neural networks* (pp. 1–8).

Lam, M. W., Wang, J., Su, D., & Yu, D. (2022). BDDM: Bilateral denoising diffusion models for fast and high-quality speech synthesis. In *International conference on learning representations*.

Lee, N., Lee, J., & Park, C. (2022). Augmentation-free self-supervised learning on graphs. In *Proceedings of the AAAI conference on artificial intelligence (vol. 36), no. 7* (pp. 7372–7380).

Li, J., Dani, H., Hu, X., & Liu, H. (2017). Radar: Residual analysis for anomaly detection in attributed networks. In *International joint conferences on artificial intelligence* (pp. 2152–2158).

Li, Y., Huang, X., Li, J., Du, M., & Zou, N. (2019). Specae: Spectral autoencoder for anomaly detection in attributed networks. In *Proceedings of the ACM international conference on information and knowledge management* (pp. 2233–2236).

Li, H., Yang, Y., Chang, M., Chen, S., Feng, H., Xu, Z., et al. (2022). Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, *479*, 47–59.

Liu, Y., Li, Z., Pan, S., Gong, C., Zhou, C., & Karypis, G. (2021). Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15.

Liu, S., Ying, R., Dong, H., Li, L., Xu, T., Rong, Y., et al. (2022). Local augmentation for graph neural networks. In *International conference on machine learning* (pp. 14054–14072).

Luo, D., Cheng, W., Yu, W., Zong, B., Ni, J., Chen, H., et al. (2021). Learning to drop: Robust graph neural network via topological denoising. In *Proceedings of the ACM international conference on web search and data mining* (pp. 779–787).

Luo, X., Wu, J., Beheshti, A., Yang, J., Zhang, X., Wang, Y., et al. (2022). ComGA: Community-aware attributed graph anomaly detection. In *Proceedings of the ACM international conference on web search and data mining* (pp. 657–665).

Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q. Z., et al. (2021). A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 1–25.

McAuley, J., Targett, C., Shi, Q., & Van Den Hengel, A. (2015). Image-based recommendations on styles and substitutes. In *Proceedings of the international ACM SIGIR conference on research and development in information retrieval* (pp. 43–52).

Meng, C., Song, Y., Song, J., Wu, J., Zhu, J.-Y., & Ermon, S. (2022). Sdedit: Image synthesis and editing with stochastic differential equations. In *International conference on learning representations*.

Neal, R. M., et al. (2011). MCMC using Hamiltonian dynamics. In *Handbook of markov chain monte carlo (vol. 2), no. 11* (p. 2).

Nichol, A. Q., & Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In *International conference on machine learning* (pp. 8162–8171).

Pang, G., Shen, C., Cao, L., & Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. *ACM Computing Surveys*, *54*(2), 1–38.

Pang, G., Shen, C., & van den Hengel, A. (2019). Deep anomaly detection with deviation networks. In *Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining* (pp. 353–362).

Popov, V., Vovk, I., Gogoryan, V., Sadekova, T., Kudinov, M., & Wei, J. (2022). Diffusion-based voice conversion with fast maximum likelihood sampling scheme. In *International conference on learning representations*.

Rasul, K., Seward, C., Schuster, I., & Vollgraf, R. (2021). Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International conference on machine learning* (pp. 8857–8868).

Rong, Y., Huang, W., Xu, T., & Huang, J. (2020). DropEdge: Towards deep graph convolutional networks on node classification. In *International conference on learning representations*.

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., et al. (2021). A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, *109*(5), 756–795.

Ruff, L., Vandermeulen, R. A., Görnitz, N., Binder, A., Müller, E., Müller, K., et al. (2020). Deep semi-supervised anomaly detection. In *International conference on learning representations*.

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., & Norouzi, M. (2022). Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(4), 4713–4726.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., & Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine*, *29*(3), 93.

Shao, M., Lin, Y., Peng, Q., Zhao, J., Pei, Z., & Sun, Y. (2023). Learning graph deep autoencoder for anomaly detection in multi-attributed networks. *Knowledge-Based Systems*, *260*, Article 110084.

Sinha, A., Song, J., Meng, C., & Ermon, S. (2021). D2C: Diffusion-decoding models for few-shot conditional generation. *Advances in Neural Information Processing Systems*, 12533–12548.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning* (pp. 2256–2265).

Song, J., Meng, C., & Ermon, S. (2021). Denoising diffusion implicit models. In *International conference on learning representations*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International conference on learning representations*.

Van Belle, R., Van Damme, C., Tytgat, H., & De Weerdt, J. (2022). Inductive graph representation learning for fraud detection. *Expert Systems with Applications*, *193*, Article 116463.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *International conference on learning representations*.

Villa-Pérez, M. E., Alvarez-Carmona, M. A., Loyola-González, O., Medina-Pérez, M. A., Velazco-Rossell, J. C., & Choo, K.-K. R. (2021). Semi-supervised anomaly detection algorithms: A comparative summary and future research directions. *Knowledge-Based Systems*, *218*, Article 106878.

Wang, X., Jin, B., Du, Y., Cui, P., Tan, Y., & Yang, Y. (2021). One-class graph neural networks for anomaly detection in attributed networks. *Neural Computing and Applications*, *33*(18), 12073–12085.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. (2019). Simplifying graph convolutional networks. In *International conference on machine learning* (pp. 6861–6871).

Wu, T., Yang, N., Chen, L., Xiao, X., Xian, X., Liu, J., et al. (2022). ERGCN: Data enhancement-based robust graph convolutional network against adversarial attacks. *Information Sciences*, *617*, 234–253.

Xia, J., Wu, L., Chen, J., Hu, B., & Li, S. Z. (2022). Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM web conference* (pp. 1070–1079).

Xian, X., Wu, T., Qiao, S., Wang, W., Wang, C., Liu, Y., et al. (2021). DeepEC: Adversarial attacks against graph structure prediction models. *Neurocomputing*, *437*, 168–185.

Xiao, C., Gou, Z., Tai, W., Zhang, K., & Zhou, F. (2023). Imputation-based time-series anomaly detection with conditional weight-incremental diffusion models. In *Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining* (pp. 2742–2751).

Xiao, Z., Kreis, K., & Vahdat, A. (2022). Tackling the generative learning trilemma with denoising diffusion gans. In *International conference on learning representations*.

Xiao, C., Xu, X., Lei, Y., Zhang, K., Liu, S., & Zhou, F. (2023). Counterfactual graph learning for anomaly detection on attributed networks. *IEEE Transactions on Knowledge and Data Engineering*.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., & Shen, Y. (2020). Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, *33*, 5812–5823.

Zhang, J., Wang, S., & Chen, S. (2022). Reconstruction enhanced multi-view contrastive learning for anomaly detection on attributed networks. In *International joint conferences on artificial intelligence*.

Zhao, T., Liu, Y., Neves, L., Woodford, O., Jiang, M., & Shah, N. (2021). Data augmentation for graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 11015–11023).

Zheng, Y., Jin, M., Liu, Y., Chi, L., Phan, K. T., & Chen, Y.-P. P. (2021). Generative and contrastive self-supervised learning for graph anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*.

Zheng, C., Zong, B., Cheng, W., Song, D., Ni, J., Yu, W., et al. (2020). Robust graph representation learning via neural sparsification. In *International conference on machine learning* (pp. 11458–11468).

Zhou, L., Du, Y., & Wu, J. (2021). 3D shape generation and completion through point-voxel diffusion. In *International conference on computer vision* (pp. 5826–5835).

Zhou, S., Huang, X., Liu, N., Tan, Q., & Chung, F.-L. (2022). Unseen anomaly detection on networks via multi-hypersphere learning. In *Proceedings of the SIAM international conference on data mining* (pp. 262–270).

Zhou, C., & Paffenroth, R. C. (2017). Anomaly detection with robust deep autoencoders. In *Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining* (pp. 665–674).

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., & Wang, L. (2021). Graph contrastive learning with adaptive augmentation. In *Proceedings of the web conference* (pp. 2069–2080).