GAD-EBM: Graph Anomaly Detection using Energy-Based Models

Amit Roy[*]
Purdue University
roy206@purdue.edu

Juan Shu
Purdue University
shu30@purdue.edu

Olivier Elshocht
Sony
Olivier.Elshocht@sony.com

Jeroen Smeets
Sony
jeroen.smeets@sony.com

Ruqi Zhang
Purdue University
ruqiz@purdue.edu

Pan Li
Georgia Institute of Technology, Purdue Unviersity
panli@gatech.edu, panli@purdue.edu

Abstract

Graph Anomaly Detection (GAD) is essential in fields ranging from network security, and bioinformatics to finance. Previous works often adopt auto-encoders to compute reconstruction errors for anomaly detection: anomalies are hard to reconstruct. In this work, we revisit the first principle for anomaly detection, i.e., the Neyman-Pearson rule, where the optimal anomaly detector is based on the likelihood of a data point given the normal distribution of data. However, in practice, the distribution is often unknown and the estimation of the distribution of graph-structured data may be hard. Moreover, the likelihood computation of a graph-structured data point may be challenging as well. In this paper, we propose a novel approach GAD-EBM that can estimate the distribution of graphs and compute likelihoods efficiently by using Energy-Based Models (EBMs) over graphs. GAD-EBM approaches the likelihood of a rooted subgraph of node v, and further can leverage the likelihood to accurately identify whether node v is anomalous or not. Traditional score matching for training EBMs may not be used to apply EBMs that model the distribution of graphs because of the complicated discreteness and multi-modality of graph data. We propose a Subgraph Score Matching (SSM) approach, which is specifically designed for graph data based on a novel framework of Subgraph State-Spaces. Experimentation conducted on six real-world datasets validates the effectiveness and efficiency of GAD-EBM and the source code for GAD-EBM is openly available.

1 Introduction

Graph Anomaly Detection (GAD) is a critical area of research [25, 47] that has a wide range of applications, spanning social network analysis [42, 38, 46, 32, 13], financial fraud detection [44, 5, 4, 14, 37, 34, 7, 29, 10, 41], cybersecurity [40, 19], and many more. The aim of GAD is to identify anomalous substructures, nodes, or edges within a graph that deviate significantly from the norm, indicating potential irregularities or issues. These anomalies could represent, for instance, fake users in a social network, compromised nodes in a computer network, or fraudulent financial transactions.

[*]Corresponding Author

New Frontiers in Graph Learning (GLFrontiers) Workshop at NeurIPS 2023.

Given the complexity and multi-modality characteristics of graph data, graph anomaly detection poses unique challenges, requiring sophisticated methodologies and algorithms to accurately and effectively identify anomalies.

There have been several approaches for GAD, which can be categorized into feature-based methods, structure-only methods, and deep-learning methods that considers both structure and feature information, and we will give a detailed review of them in Section 2. Most of them are based on heuristics. If we revisit the first principle of detecting anomalies, the optimal detection, based on the Neyman-Pearson rule (NP rule) [27], should be decided by the likelihood of a data point to be detected given the normal distribution of data. The lower the likelihood is, the more likely to be anomaly such a data point is. Although this rule is fundamental, it is hard to be directly used in practice, because the normal distribution of data is often unknown. Moreover, even if the normal distribution of the data can be estimated, whether the likelihood can be easily computed is another challenge. For example, VAEs [17], GANs [12], and diffusion models [16] can estimate the distribution given an empirical dataset, but the likelihood of a data point based on the distribution estimated by these models is hard to compute. Normalizing flow, another model possibly used to estimate a distribution, is able to compute the likelihood efficiently. However, normalizing flow, due to the requirement of reversible encoding, imposes massive constraints on the model architecture, which may be hard to accurately estimate the normal distribution for complex multi-modal graph-structured data. These observations motivate us to think about what should be the best model that can estimate the data distribution, compute likelihood efficiently and be expressive enough to capture subtle patterns of multi-modal graph-structured data in the same time.

The above question motivates us to investigate Energy-Based Models (EBMs) for GAD tasks. EBMs can tackle the above challenges by learning an unnormalized probability density function of the underlying data distribution [33]. Given a data point of interest, EBMs can directly compute its unnormalized likelihood. Although the normalization constant for an EBM is generally hard to compute, for anomaly detection tasks, the constant is not necessary. This is because such normalization constant is shared across different candidates and the ranking of anomalous labels of different candidates will be kept regardless of the constant. Moreover, EBMs often do not add constraints to their data encoders, which have the potential to accurately model the complex distribution of graph-structured data.

Albeit the decent properties of EBMs for GAD tasks, EBMs are often hard to train. Maximum likelihood estimation, score-matching, and contrastive divergence are the possible training approaches [33]. Among them, maximum likelihood estimation requires Markov Chain Monte Carlo (MCMC) sampling to estimate the normalization constant of the currently learned model [28]. Such a sampling procedure often takes a large number of iterations to converge. In contrast, score-matching [15] is often more efficient to train EBMs. Score in score-matching methods is defined as the gradient of the log probability density function, so score-matching-based training is typically applied to learn continuous distributions. However, graphs consist of nodes and edges that encapsulate discrete relationships, and they do not naturally reside in continuous spaces. Such discreteness impedes the use of conventional score-matching methods to train EBMs for GAD.

Present work: To efficiently and effectively train an EBM for GAD, in this work, we proposed Subgraph-Score Matching (SSM). SSM is to train an EBM with a graph neural network as the data encoder that models an unnormalized probability density function over (sub)graphs. Then, given the subgraph round one node in the observed graph, the learned EBM can compute unnormalized likelihood of this subgraph, which indicates the anomalous level and further indicates whether the center node is an anomaly or not.

SSM's efficient training is inspired by the recent idea of concrete score matching [26]. Concrete score defines a surrogate of the gradient to train EBMs in discrete spaces. It defines a state-space graph whose nodes correspond to all possible states (values) of a random variable in the discrete space and are connected via some pre-defined edges in the state-space graph. Note that this state-space graph is a math abstract, which is different from the graphs or subgraphs in data. Given this state-space graph, the directional difference between the likelihoods of a node and of its neighbors in this state-space graph provides an analogy of the gradient in the continuous space.

In SSM, we define a novel Subgraph State-Space whose nodes essentially correspond to subgraphs in the observed graph-structured data, the EBM measures the likelihoods of these subgraphs. We define edges (or the neighboring relations) in the Subgraph State-Space when one subgraph can be

transformed into another via edge addition, edge deletion, or feature shuffle. Therefore, in our EBM, the directional changes of the likelihoods between the subgraphs directly connected in the Subgraph State-Space give the concrete scores in our case. With these concrete scores, we are able to use concrete score matching [26] to efficiently train the EBM that ultimately estimates the likelihoods of subgraphs. Concrete score matching provably learns the ground truth distribution if the state-space graph is connected, which can be proved for our defined Subgraph State-Space. As each subgraph state in our Subgraph State-Space may have exponentially many neighbors, we also propose a random sampling procedure over the Subgraph State-Space so that the training objective of concrete score matching can be åefficiently estimated. The contributions of our work can be summarized as follows:

- In this study, we introduce GAD-EBM, the first approach that utilizes Energy-Based Models (EBM) for Graph Anomaly Detection (GAD).
- We address the challenge of training Energy-Based Models in the discrete space of graphs by introducing a subgraph score-matching objective with a surrogate of the gradient at the subgraph level.
- We propose a novel State Space Graph which enables efficient training of Energy-Based Models via subgraph score-matching to learn the likelihood of the ego-subgraph of a target node that determines its anomaly label.
- Empirically, we showed that our model GAD-EBM may show competitive performance with baseline models in benchmark GAD datasets and can distinguish between normal and anomaly nodes.

2 Related Works

2.1 Graph Anomaly Detection

Early approaches for GAD often focused on contextual and structural anomalies where the contextual anomalies are featurewise different from other nodes and structural anomalies are densely connected [22]. The GAD methods can be broadly categorized into two groups. The first group of methods handles graph structure and node attributes independently. For example, feature-only approaches [1, 21, 31] only consider node features and perform well in detecting contextual anomalies whereas approaches that only consider network structure find success in detecting structural anomalies [45]. The second group of strategies identifies contextual and structural anomalies using a unified framework where generative-model-based approaches are the most popular. Auto-Encoders, Generative Adversarial Networks (GAN), normalizing flow, and diffusion models all belong to generative model-based approaches, which have been extensively used for anomaly detection [6, 18, 2, 11]. However, autoencoders try to use reconstruction error of links for detecting anomalies, GANs offer a means to detect anomalies but they may suffer from the unstable adversarial training procedure, thus leading to unstable and biased anomaly detection. Normalizing flow-based approaches depends on the inversion of learnable function whereas diffusion models consider the difference between the input data and denoised data to detect anomalies. These approaches find it difficult to accurately represent the true underlying data distribution and not principled for detecting anomalies.

2.2 Energy-Based Models

The Energy-Based Model (EBM) is a kind of generative model to directly model the unnormalized probability density function of the underlying data distribution, which has been used in different domains including image [9, 8], video [39], and text [24]. In the graph domain, EBM has been applied for tasks including OOD detection [43], molecular graph generation [23], scene-graph generation [35]. Several approaches are mentioned in the literature to train energy-based models [33] namely maximum likelihood via MCMC sampling [28], score-matching [36], denoising score-matching [20], and contrastive divergence [30]. The gradient of the log probability density function is defined as the score of the underlying data distribution. In score-matching-based approaches, the goal is to learn a distribution with the objective of minimizing the fisher divergence between the score of learned and true data distribution. With gradient not being defined in the discrete space, it becomes challenging to utilize the score-matching technique for training energy-based models in graphs. However, defining a surrogate of gradient concrete score-matching approach [26] for training

energy-based models is a promising direction to train energy-based models for graph-like discrete data.

3 Preliminaries

3.1 Graph Anomaly Detection

The primary objective of this study is to identify anomalous nodes within a graph. We denote a undirected graph as $G = \{V, E, X\}$, where $V = \{1, 2, \ldots, N\}$ denotes the node list and $E \in V \times V$ denotes the edge list. We use $X \in \mathbb{R}^{|V| \times d}$ to represent the attribute matrix and $x_v \in \mathbb{R}^d$ to denote the feature of node v and e_{vu} denotes the edge between node v and u. The study targets an unsupervised anomaly detection problem. Each node v is associated with an anomaly label y where $y = 0$ designates the node as normal and $y = 1$ implies node v is anomalous. These labels, however, are unknown during the training process. The goal of this study is to estimate the likelihood of the ego-subgraph G_v^{sub} rooted at a target node v and use the computed likelihood as the anomaly score for each node, where G_v^{sub} denotes the induced subgraph of the root vertex v in the graph G. The task is to learn a likelihood mapping function $f(G_v^{\text{sub}}) : G_v^{\text{sub}} \rightarrow \{0, 1\}, v \in V$ that leverages the local structural information (G_v^{sub}) of a target node v to measure the likelihood of v to be an anomaly.

3.2 Energy-Based Models and Score Matching

Energy-Based Models (EBM) are known to model the unnormalized probability density of a data distribution and can be trained without the restriction of the traceability of an unknown normalizing constant which makes them flexible to model a more expressive family of probability distributions. The density of a data distribution over data samples x given by the EBMs are

$$p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z_\theta}$$

where E_θ is a non-linear real-valued function parameterized by θ and Z_θ is the nomralizing constant.

Score-matching is one of the ways to train EBMs where the first-order gradient of the log probability density function is known as the score of a distribution. By converting the equivalent distribution to an equivalent score, it is easy to match the scores to train the EBMs $\nabla_x \log p_\theta(x) = -\nabla_x E_\theta(x)$ not involving any normalizing constant Z_θ. The problem of learning subgraph distribution can also be modeled using EBMs using score-matching as the training strategy.

Let the random variable $\mathbf{G}^{\text{sub}}$ denote a subgraph and $p_{\text{data}}(\mathbf{G}^{\text{sub}})$ be an unknown subgraph distribution over the subgraph space $\mathcal{G}^{\text{sub}}$. We have the observed subgraphs $\{G_v^{\text{sub}}\}_{v=1}^N \sim p_{\text{data}}(\mathbf{G}^{\text{sub}})$ where $\mathbf{G}_v^{\text{sub}}$ denote a subgraph centered at node v. The goal of score-matching is to learn a probability distribution $p_\theta(\mathbf{G}^{\text{sub}})$ that can be written as a form of energy $E_\theta(\mathbf{G}^{\text{sub}})$:

$$p_\theta(\mathbf{G}^{\text{sub}}) = \frac{\exp(-E_\theta(\mathbf{G}^{\text{sub}}))}{Z_\theta} \tag{1}$$

where energy $E_\theta(\mathbf{G}^{\text{sub}})$ is a non-linear function parameterized by θ . Z_θ is the normalizing constant satisfying $Z_\theta = \int \exp(-E_\theta(\mathbf{G}^{\text{sub}}))d\mathbf{G}^{\text{sub}}$. Unfortunately, Z_θ is a function of θ so the evaluation and differentiation of log-likelihood with respect to model parameters involve an intractable integral.

Instead of approximating the intractable normalizing constant, score matching aims at approximating the score function, estimating the $s_\theta(\mathbf{G}^{\text{sub}}) = \nabla_{\mathbf{G}^{\text{sub}}} \log p_\theta(\mathbf{G}^{\text{sub}})$, the gradient of the log probability density of the learned subgraph distribution. This is valid because when $f(\mathbf{G}^{\text{sub}})$ and $g(\mathbf{G}^{\text{sub}})$ are log density with equal first derivatives, the normalization requirement $\int \exp(f(\mathbf{G}^{\text{sub}}))\mathbf{d}\mathbf{G}^{\text{sub}} = \int \exp(g(\mathbf{G}^{\text{sub}}))\mathbf{d}\mathbf{G}^{\text{sub}} = 1$ guarantees that we have $f(\mathbf{G}^{\text{sub}}) \equiv g(\mathbf{G}^{\text{sub}})$ [33] . Therefore, by minimizing the following fisher divergence:

$$D_f(p_{\text{data}}(\mathbf{G}^{\text{sub}})||p_\theta(\mathbf{G}^{\text{sub}})) = \mathbb{E}_{p_{\text{data}}(\mathbf{G}^{\text{sub}})}[||\nabla_{\mathbf{G}^{\text{sub}}} \log p_{\text{data}}(\mathbf{G}^{\text{sub}}) - \nabla_{\mathbf{G}^{\text{sub}}} \log p_\theta(\mathbf{G}^{\text{sub}})||_2^2] \tag{2}$$

where $p_{\text{data}}(\mathbf{G}^{\text{sub}})$ is the underlying subgraph distribution and $p_\theta(\mathbf{G}^{\text{sub}})$ is the learnable subgraph distribution parameterized by $\theta \in \Theta$. $p_{\text{data}}(\mathbf{G}^{\text{sub}})$ is unknown during training so that a tractable

objective function [26] is derived as:

$$\mathbb{E}_{p_{\text{data}}(\mathbf{G}^{\text{sub}})}\left[\frac{1}{2}||s_\theta(\mathbf{G}^{\text{sub}})||_2^2 - tr(\nabla_{\mathbf{G}^{\text{sub}}} s_\theta(\mathbf{G}^{\text{sub}}))\right] \tag{3}$$

where $s_\theta(\mathbf{G}^{\text{sub}})$ is the score network parameterized by θ. However, when the data is a graph, standard gradient-based score-matching techniques may not apply directly because the gradient is not explicitly defined for graph data. That is to say, $\nabla_{\mathbf{G}^{\text{sub}}}$ is not well defined in the above equation. Therefore, we target to design a surrogate of the gradient in the graph domain so that we can train EBM with such surrogate score matching.

4 Preliminaries

4.1 Graph Anomaly Detection

The primary objective of this study is to identify anomalous nodes within a graph. We denote a undirected graph as $G = \{V, E, X\}$, where $V = \{1, 2, \ldots, N\}$ denotes the node list and $E \in V \times V$ denotes the edge list. We use $X \in \mathbb{R}^{|V| \times d}$ to represent the attribute matrix and $x_v \in \mathbb{R}^d$ to denote the feature of node v and e_{vu} denotes the edge between node v and u. The study targets an unsupervised anomaly detection problem. Each node v is associated with an anomaly label y where $y = 0$ designates the node as normal and $y = 1$ implies node v is anomalous. These labels, however, are unknown during the training process. The goal of this study is to estimate the likelihood of the ego-subgraph G_v^{sub} rooted at a target node v and use the computed likelihood as the anomaly score for each node, where G_v^{sub} denotes the induced subgraph of the root vertex v in the graph G. The task is to learn a likelihood mapping function $f(G_v^{\text{sub}}) : G_v^{\text{sub}} \to \{0, 1\}, v \in V$ that leverages the local structural information (G_v^{sub}) of a target node v to measure the likelihood of v to be an anomaly.

4.2 Energy-Based Models and Score Matching

Energy-Based Models (EBM) are known to model the unnormalized probability density of a data distribution and can be trained without the restriction of the traceability of an unknown normalizing constant which makes them flexible to model a more expressive family of probability distributions. The density of a data distribution over data samples x given by the EBMs are

$$p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z_\theta}$$

where E_θ is a non-linear real-valued function parameterized by θ and Z_θ is the nomralizing constant.

Score-matching is one of the ways to train EBMs where the first-order gradient of the log probability density function is known as the score of a distribution. By converting the equivalent distribution to an equivalent score, it is easy to match the scores to train the EBMs $\nabla_x \log p_\theta(x) = -\nabla_x E_\theta(x)$ not involving any normalizing constant Z_θ. The problem of learning subgraph distribution can also be modeled using EBMs using score-matching as the training strategy.

Let the random variable $\mathbf{G}^{\text{sub}}$ denote ego-subgraph and $p_{\text{data}}(\mathbf{G}^{\text{sub}})$ be an unknown subgraph distribution over the subgraph space $\mathcal{G}^{\text{sub}}$. We have the observed subgraphs $\{G_v^{\text{sub}}\}_{v=1}^N \sim p_{\text{data}}(\mathbf{G}^{\text{sub}})$ where $\mathbf{G}_v^{\text{sub}}$ denote a subgraph centered at node v. The goal of score-matching is to learn a probability distribution $p_\theta(\mathbf{G}^{\text{sub}})$ that can be written as a form of energy $E_\theta(\mathbf{G}^{\text{sub}})$:

$$p_\theta(\mathbf{G}^{\text{sub}}) = \frac{\exp(-E_\theta(\mathbf{G}^{\text{sub}}))}{Z_\theta} \tag{4}$$

where energy $E_\theta(\mathbf{G}^{\text{sub}})$ is a non-linear function parameterized by θ . Z_θ is the normalizing constant satisfying $Z_\theta = \int \exp(-E_\theta(\mathbf{G}^{\text{sub}}))d\mathbf{G}^{\text{sub}}$. Unfortunately, Z_θ is a function of θ so the evaluation and differentiation of log-likelihood with respect to model parameters involve an intractable integral.

Instead of approximating the intractable normalizing constant, score matching aims at approximating the score function, estimating the $s_\theta(\mathbf{G}^{\text{sub}}) = \nabla_{\mathbf{G}^{\text{sub}}} \log p_\theta(\mathbf{G}^{\text{sub}})$, the gradient of the log probability density of the learned subgraph distribution. This is valid because when $f(\mathbf{G}^{\text{sub}})$ and $g(\mathbf{G}^{\text{sub}})$ are log density with equal first derivatives, the normalization requirement $\int \exp(f(\mathbf{G}^{\text{sub}}))\mathbf{d}\mathbf{G}^{\text{sub}} =$

$\int \exp(g(\mathbf{G}^{\text{sub}}))\mathbf{dG}^{\text{sub}} = 1$ guarantees that we have $f(\mathbf{G}^{\text{sub}}) \equiv g(\mathbf{G}^{\text{sub}})$ [33] . Therefore, by minimizing the following fisher divergence:

$$D_f(p_{\text{data}}(\mathbf{G}^{\text{sub}})||p_\theta(\mathbf{G}^{\text{sub}})) = \mathbb{E}_{p_{\text{data}}(\mathbf{G}^{\text{sub}})}[||\nabla_{\mathbf{G}^{\text{sub}}} \log p_{\text{data}}(\mathbf{G}^{\text{sub}}) - \nabla_{\mathbf{G}^{\text{sub}}} \log p_\theta(\mathbf{G}^{\text{sub}})||_2^2] \quad (5)$$

where $p_{\text{data}}(\mathbf{G}^{\text{sub}})$ is the underlying subgraph distribution and $p_\theta(\mathbf{G}^{\text{sub}})$ is the learnable subgraph distribution parameterized by $\theta \in \Theta$. $p_{\text{data}}(\mathbf{G}^{\text{sub}})$ is unknown during training so that a tractable objective function [26] is derived as:

$$\mathbb{E}_{p_{\text{data}}(\mathbf{G}^{\text{sub}})}\left[\frac{1}{2}||s_\theta(\mathbf{G}^{\text{sub}})||_2^2 - tr(\nabla_{\mathbf{G}^{\text{sub}}} s_\theta(\mathbf{G}^{\text{sub}}))\right] \quad (6)$$

where $s_\theta(\mathbf{G}^{\text{sub}})$ is the score network parameterized by θ. However, when the data is a graph, standard gradient-based score-matching techniques may not apply directly because the gradient is not explicitly defined for graph data. That is to say, $\nabla_{\mathbf{G}^{\text{sub}}}$ is not well defined in the above equation. Therefore, we target to design a surrogate of the gradient in the graph domain so that we can train EBM with such surrogate score matching.

5 Methodology

In this section, we describe our model GAD-EBM that aims at the computation of the likelihood of a target node's neighborhood to determine its anomaly label with the help of energy-based models. We designed Subgraph State-Space , defined as a network structure of ego-subgraphs that provides the flexibility to design a training strategy for energy-based models in discrete data-like graphs. We leverage the idea of Concrete Score [26] which defines a surrogate of the gradient by considering the likelihood change rate in a neighborhood structure in the discrete space.

5.1 Concrete Score Matching

Given the subgraph space $\mathcal{G}$ and the function mapping termed as the neighborhood structure $\mathcal{N}_G : \mathcal{G} \to \mathcal{G}^K$ from each observed subgraph $G_v^{\text{sub}} \in \mathcal{G}$ to its neighborhood subgraphs $\{G_1^{sub}, G_2^{sub}, \cdots, G_K^{sub}\}$, the concrete-score $c_{p_{\text{data}}}(G_v; \mathcal{N}_G)$ over the distribution of subgraphs $p_{\text{data}}(\mathbf{G}^{\text{sub}})$ is defined as follows

$$c_{p_{\text{data}}}(G_v; \mathcal{N}_G) \triangleq \left[\frac{p_{\text{data}}\left(G_1^{\text{sub}}\right) - p_{\text{data}}(G_v^{\text{sub}})}{p_{\text{data}}(G_v^{\text{sub}})}, \ldots, \frac{p_{\text{data}}\left(G_K^{\text{sub}}\right) - p_{\text{data}}(G_v^{\text{sub}})}{p_{\text{data}}(G_v^{\text{sub}})}\right]^T \quad (7)$$

The intuition of concrete score is to find the local directional changes of likelihood between a observed subgraph G_v^{sub} and its neighbor subgraphs $\{G_1^{sub}, G_2^{sub}, \cdots, G_K^{sub}\}$, which can be considered as a surrogate of the gradient for discrete space. The concrete score is constrained by the requirement of the neighborhood structure $\mathcal{N}_G$ to be connected.

However, the extension of the concrete score to the graph domain is by no means an easy task. Graphs encapsulate relationships between nodes and edges, and the discrete, combinatorial structure of these relationships may not align well with learning EBMs with the concrete score objective. Additionally, the assumptions underlying the use of concrete score are more straightforward in the predefined neighborhood structure of discrete feature space but do not hold in the more complex setting of graph data where both structure and feature information are available. Therefore, designing a score that can leverage the intricate dependencies and interactions within graph data is of great importance.

5.2 Subgraph-State-Space

To propose a score for the subgraph setting that will facilitate the training of Energy-Based Models on the subgraph distribution, we will first define a neighborhood structure subgraph-state-space. Before we define the Subgraph-state space, it is necessary to first introduce some prerequisite definitions that will be used in its characterization.

Definition 1. *(Ego-neighborhood). Given a target node v, its 1-hop neighborhood $G_v^{sub} = \{\{v, u\}, \{x_v, x_u\}, \{e_{vu}\} | d_G(u, v) = 1, u \in V\}$ is defined as the ego-neighborhood of node v.*

Definition 2. *(Subgraph State-Space). The Subgraph State-Space is defined as a set of ego-neighborhoods $\{v_G, u_{G_1}, \cdots, u_{G_\Omega}\}$, which are composed of one observed ego-neighborhood of target node v, denoted as v_G ($v_G \equiv G_v^{sub}$) and Ω ego-neighborhoods as the neighbor node set of v_G, denoted as $\mathcal{N}_{v_G}$, where Ω is the number of all possible ego-neighborhoods. Each ego-neighborhood $u_G \in \mathcal{N}_{v_G}$ is constructed from v_G by applying a collection of operations: 1)* ***edge-addition****: connecting node v with any nodes $w \in V$, 2)* ***edge-deletion****: disconnecting v with any nodes $w \in \mathcal{N}_v$ and 3)* ***feature shuffle****: shuffling the features of nodes in v_G with any node features in X.*

Definition 3. *(Neighbor Subgraphs). We define two subgraphs as neighbor subgraphs in the subgraph space if one of them can be converted into another by performing either one of the edge-addition, edge-deletion, and feature-shuffle operations. As in definition 3, we can see v_G is connected to every $u_G \in \mathcal{N}_{v_G}$.*

Due to the way we produce the negative neighbors in a Subgraph State-Space, the quantity of them is exponentially large. In practice, we only sample some of them in a Subgraph State-Space for computational efficiency, e.g., $|\mathcal{N}_{v_G}| = l$, where $l \ll |\Omega|$. An example of a Subgraph State-Space can be found in Figure 1, where $l = 2$ as produce two neighbor states in the Subgraph State-Space by perturbing the neighborhood structure and shuffling features.

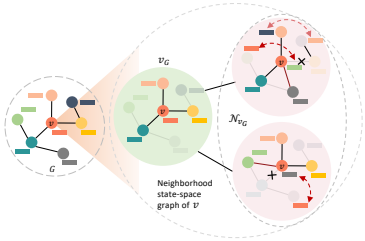

Figure 1: Example of a Subgraph State-Space

5.3 The Subgraph Score

With the previously established definitions in place, we can now align the surrogate of gradient and the likelihood differences of state subgraph, akin to the directional derivative in concrete score matching. Utilizing this concept of the "gradient", we formulate the subgraph score as the rate of change of the likelihood with respect to the observed subgraph, as detailed below.

Definition 4. *(Subgraph Score). The subgraph score of an ego-neighborhood rooted at node v is defined as $b_{p_{data}}(v_G) : \mathcal{G}^{sub} \to (0,1)^{\mathcal{N}_{v_G}}$ for a given distribution $p_{data}(\mathbf{G}_v)$.*

$$b_{p_{data}}(v_G) \triangleq \left[\cdots, \frac{p_{data}(u_G) - p_{data}(v_G)}{p_{data}(v_G)}, \cdots\right]_{u_G \in \mathcal{N}_{v_G}} \tag{8}$$

Subgraph score is a valid surrogate of the gradient for score matching only when it enables us to uniquely identify the $b_{p_{\text{data}}}(v_G)$. The following Corollary 1 guarantees that this identification can be achieved. Before that, we need to define the connectivity between subgraphs, which is the prerequisite of the corollary.

For the Figure 1, the subgraph score $b_{p_{\text{data}}}(v_G) : \mathcal{G}^{sub} \to (0,1)^2$ which will act as the surrogate of the gradient by computing the ratio of the difference of the neighbor state likelihood and observed state likelihood in the Subgraph State-Space to the likelihood of observed state.

Corollary 1. *(Theorem 1 in [26]) Let $p_{data}(v_G)$ be a distribution of the observed subgraph v_G in a Subgraph State-Space of node v. The subgraph score constructed on $p_{data}(v_G)$ and $p_\theta(v_G)$ of node v is denoted as $b_{p_{data}}(v_G)$ and $b_\theta(v_G)$ respectively, where $p_\theta(v_G)$ is a learnt empirical distribution parameterized by $\theta \in \Theta$. Due to the connectivity of the v_G and $u_G \in \mathcal{N}_{v_G}$, we can conclude that $b_{p_{data}}(v_G) = b_\theta(v_G)$ implies that $p_{data}(v_G) = p_\theta(v_G)$.*

Proof sketch : The way we produce neighbors in the state-space graph by shuffling features among the nodes in the state-space graph and via edge addition/deletion we can conclude that we can produce any observed subgraph as a neighbor state in the state-space graph. From the concrete score-matching [26] idea, we have that if the neighboring structure is connected the learned data distribution becomes equal to the true data distribution. Therefore, we can conclude that our strategy of producing a state in the Subgraph State-Space can reach all possible states in the Subgraph State-Space. This implies that the Subgraph State-Space is connected resulting in $p_{\text{data}}(v_G) = p_\theta(v_G)$.

5.4 Connection to Concrete Score

We leverage the idea of calculating a surrogate score based on directional changes of input in a discrete feature domain of images. In detail, we measure $(p(u_G) - p(v_G))/p(v_G)$ as a surrogate score function, where $u_G \in \mathcal{N}_{v_G}$. However, the Concrete score [26] was developed to handle general discrete cases. But because of the complex topological structure of graph data, concrete score matching is unable to handle such discreteness. The subgraph score is designed specifically for graph data. It recognizes and handles the inherent intricacies of graphs, accommodating the local structural information that is key to understanding the behavior of a target node and thus recovering the likelihood successfully.

5.5 Subgraph Score Matching objective

We apply the fisher divergence that is commonly used in the score matching [33] objective function.

$$\mathcal{L}_{SSM}(\theta) = \sum_{v_G \in \mathcal{V}} p_{\text{data}}(v_G) || b_\theta(v_G) - b_{p_{\text{data}}}(v_G) ||_2^2 \tag{9}$$

where $b_\theta(v_G)$ is a subgraph score model parameterized by $\theta \in \Theta$. In the experiment, we use a 1-layer GNN to approximate the true score $b_{p_{\text{data}}}(v_G)$. However, $b_{p_{\text{data}}}(v_G)$ is unknown during the training so the objective function Eq.(9) is intractable.

Theorem 1. *Optimizing objective function in Equation (9) is equivalent to optimizing the following:*

$$\mathcal{L}_{SSM}(\theta) = \underbrace{\sum_{v_G \in \mathcal{V}} p_{data}(v_G) \left(\|b_\theta(v_G)\|_2^2 + 2\|b_\theta(v_G)\| \right)}_{\mathcal{J}_1} - \underbrace{\sum_{v_G \in \mathcal{V}} ||b_\theta(v_G)|| \sum_{u_G \in \mathcal{N}_{v_G}} 2p_{data}(u_G)}_{\mathcal{J}_2} \tag{10}$$

The proof of Theorem 1 is deferred to the Appendix in Section 8.

In the objective function, we can view that in the first term $\mathcal{J}_1$ is minimized which essentially minimizes $||b_\theta(v_G)||_2^2$ i.e. according to the definition of subgraph score the objective function maximizes the likelihood of the observed ego-subgraph $p_{\text{data}}\,(v_G)$ present in the network while decreasing the likelihood of the ego-subgraphs $p_{\text{data}}(u_G)$ present as a neighbor in the state-space graph. For the second term $\mathcal{J}_2$, the optimizer will push this term to be as large as possible to minimize the objective function. So, we use the inverse of the equation $b_\theta(u_G)^2$ to increase the $p_{\text{data}}(v_G)$ and decrease the $p_{\text{data}}(u_G)$ which matches our intuition that the likelihood of the ego-subgraph will be higher whereas the likelihood of anomaly node's subgraphs will be comparatively low.

5.6 Experimental Settings

Our first experimental settings follow the benchmark outlier node detection paper (BOND) [22]. To estimate the p_{data}, we experimented with different types of GNN architectures e.g. GCN, GraphSAGE, GIN. Other than shuffling the node features, we perturbed the structure of the ego-subgraph of each target/centered node by adding/removing c percentage of edges and experimented with $c = \{5\%, 10\%, 15\%, 20\%\}$. We also tuned the number of neighbors in the state-space graph, $l = 1, 2, 5, 10, 20$. We optimize the parameters of GAD-EBM using Adam optimizer with different learning rates $\{0.001, 0.01, 0.1\}$ and regularization co-efficient $\{0.01, 0.1, 1.0, 10.0\}$.

5.7 Evaluation Metric

We utilized the area under the Receiver Operating Characteristic (ROC) curve as our metric for evaluation. The ROC curve is constructed by plotting the true positive rate versus the false positive rate at different threshold levels. In our experiment, the anomaly nodes are treated as the positive class, and the Area Under the Curve (AUC) is calculated accordingly. An AUC value of 1 signifies that the model has achieved perfect prediction, while an AUC value of 0.5 indicates that the model lacks any discriminatory power. Unlike accuracy, AUC is preferred for evaluating anomaly detection tasks, as it remains insensitive to the imbalanced class distribution often present in the data.

Algorithm	Weibo	Reddit	Disney	Books	Enron	DGraph
LOF	56.5 ± 0.0 (56.5)	57.2 ± 0.0 (57.2)	47.9 ± 0.0 (47.9)	36.5 ± 0.0 (36.5)	46.4 ± 0.0 (46.4)	TLE
IF	53.5 ± 2.8 (57.5)	45.2 ± 1.7 (47.5)	57.6 ± 2.9 (63.1)	43.0 ± 1.8 (47.5)	40.1 ± 1.4 (43.1)	**60.9 ± 0.7 (62.0)**
MLPAE	82.1 ± 3.6 (86.1)	50.6 ± 0.0 (50.6)	49.2 ± 5.7 (64.1)	42.5 ± 5.6 (52.6)	73.1 ± 0.0 (73.1)	37.0 ± 1.9(41.3)
SCAN	63.7 ± 5.6 (70.8)	49.9 ± 0.3 (50.0)	50.5 ± 4.0 (56.1)	49.8 ± 1.7 (52.4)	52.8 ± 3.4 (58.1)	TLE
Radar	**98.9 ± 0.1 (99.0)**	54.9 ± 1.2 (56.9)	51.8 ± 0.0 (51.8)	52.8 ± 0.0 (52.8)	80.8 ± 0.0 (80.8)	OOM_C
ANOMALOUS	**98.9 ± 0.1 (99.0)**	54.9 ± 5.6 (60.4)	51.8 ± 0.0 (51.8)	52.8 ± 0.0 (52.8)	80.8 ± 0.0 (80.8)	OOM_C
GCNAE	90.8 ± 1.2 (92.5)	50.6 ± 0.0 (50.6)	42.2 ± 7.9 (52.7)	50.0 ± 4.5 (57.9)	66.6 ± 7.8 (80.1)	40.9 ± 0.5 (42.2)
DOMINANT	85.0 ± 14.6 (92.5)	56.0 ± 0.2 (56.4)	47.1 ± 4.5 (54.9)	50.1 ± 5.0 (58.1)	73.1 ± 8.9 (85.0)	OOM_C
DONE	85.3 ± 4.1 (88.7)	53.9 ± 2.9 (59.7)	41.7 ± 6.2 (50.6)	43.2 ± 4.0 (52.6)	46.7 ± 6.1 (67.1)	OOM_C
AdONE	84.6 ± 2.2 (87.6)	50.4 ± 4.5 (58.1)	48.8 ± 5.1 (59.2)	53.6 ± 2.0 (56.1)	44.5 ± 2.9 (53.6)	OOM_C
AnomalyDAE	91.5 ± 1.2 (92.8)	55.7 ± 0.4 (56.3)	48.8 ± 2.2 (55.4)	62.2 ± 8.1 (73.2)	54.3 ± 11.2 (69.1)	OOM_C
GAAN	92.5 ± 0.0 (92.5)	55.4 ± 0.4 (56.0)	48.0 ± 0.0 (48.0)	54.9 ± 5.0 (61.9)	73.1 ± 0.0 (73.1)	OOM_C
GUIDE	OOM_C	OOM_C	38.8 ± 8.9 (52.5)	48.4 ± 4.6(63.5)	OOM_C	OOM_C
CONAD	85.4 ± 14.3 (92.7)	56.1 ± 0.1 (56.4)	48.0 ± 3.5 (53.1)	52.2 ± 6.9 (62.9)	71.9 ± 4.9 (84.9)	34.7±1.2 (36.5)
GAD-EBM	93.16 ± 1.84	**58.50 ± 1.58**	**74.52 ± 0.57**	**64.30 ± 0.92**	**80.94 ± 1.42**	60.26 ± 2.48

Table 1: Performance comparison (ROC-AUC) of GAD-EBM with baseline models mentioned in BOND paper [22] on six different real-world graph anomaly detection datasets.

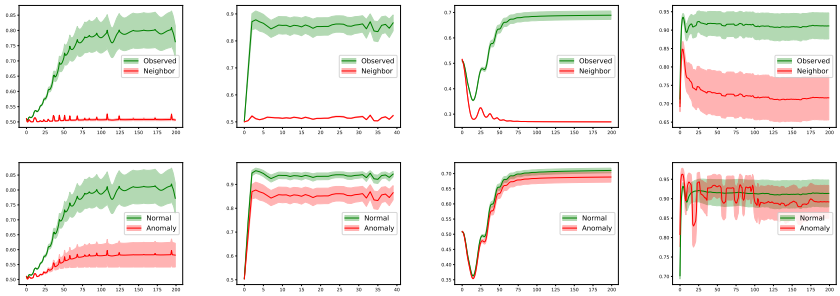

Figure 2: Likelihood change of observed and neighbor subgraphs (top) and normal and anomaly nodes (bottom) plotted after every five epochs for Disney, Books, Reddit and Weibo dataset respectively

5.8 Anomaly Detection Performance

In table 1, we present the performance of GAD-EBM to detect benchmark anomalies in six real-world graph anomaly detection datasets. From the results, we can observe that GAD-EBM can outperform baseline models in four datasets Reddit, Disney, Books, and Enron while showing the second-best performance in Weibo and DGraph datasets. From the results, it is evident that our design of the Subgraph State-Space to optimize the subgraph score-matching objective is suitable for differentiating between the normal and anomaly nodes by learning the likelihood of the ego-subgraph. Our objective function increases the likelihood of the observed ego-subgraphs v_G in a state-space graph while decreasing the likelihood of neighbor ego-subgraphs $u_G \in \mathcal{N}_{v_G}$ which eventually results in differentiating the normal and anomaly nodes

5.9 Average likelihood comparison

In Figure 2, we plot the likelihood change for every five epochs for the observed and neighbor ego-subgraph of a target node (top) as well as for normal and anomaly nodes (bottom) in the state-space graph for the Disney, Books, Enron, and Weibo datasets. From the likelihood change plot between the observed and neighbor ego-subgraph in the state-space graph, we can observe that the likelihood of the observed ego-subgraph depicted with the green band increases whereas the red band suggests that the likelihood of the neighbor ego-subgraph in the state-space graph doesn't improve consistently with epochs. This behavior supports our intuition that the subgraph score-matching objective function creates a contrastive difference between the likelihood of the observed ego-subgraph and the neighbor ego-subgraphs. The average likelihood comparison plot between normal and anomaly nodes also supports our intuition. The average likelihood of normal nodes is always higher than Disney, Books, and Reddit datasets whereas we can observe some fluctuation for the Weibo dataset. The fact that the normal nodes get a higher likelihood than the anomaly nodes makes it easier for the model to differentiate between normal and anomaly nodes, resulting in the better performance of GAD-EBM.

5.10 Running Time Efficiency

In Table 2, we compare the running time of GAD-EBM with other baseline GAD models. From the comparison we can observe that the running time efficiency of GAD-EBM is comparable to other baseline GAD approaches. The reason behind such efficiency is the flexible design of GAD-EBM's framework to produce neighbors in the state-space graphs. We adopt a full-batch implementation of the objective function by applying row shuffle in the feature matrix X and adding/removing a certain percentage of edges w.r.t. node degree which results in a faster pipeline of GAD-EBM.

Algorithm	**10**	**100**	**200**	**300**	**400**
LOF	0.10	0.10	0.10	0.10	0.10
IF	0.09	0.09	0.09	0.09	0.09
MLPAE	0.04	0.46	0.82	1.37	1.74
SCAN	0.02	0.02	0.02	0.02	0.02
Radar	0.02	0.10	0.19	0.34	0.36
ANOMALOUS	0.02	0.09	0.17	0.26	0.36
GCNAE	0.06	0.49	0.96	1.45	1.94
DOMINANT	0.08	0.70	1.41	2.10	2.79
DONE	0.08	0.77	1.53	2.30	3.08
AdONE	0.10	0.91	1.81	2.71	3.62
AnomalyDAE	0.43	0.64	1.28	1.92	2.55
GAAN	0.06	0.49	0.98	1.47	1.97
GUIDE	50.77	51.92	53.40	54.27	55.21
CONAD	0.11	1.04	2.07	3.07	4.10
GAD-EBM	0.09	0.62	1.24	1.89	2.47

Table 2: Running time (in seconds) comparison of baseline GAD-models with GAD-EBM for five different number of epochs. Total time is reported for the non-iterative algorithms, i.e. LOF, IF, and SCAN.

6 Conclusion

In this work, we proposed GAD-EBM to explicitly evaluate the likelihood of normal and anomalous nodes to address the graph anomaly detection problem with an energy-based model. We propose a novel framework Subgraph State-Space to train the energy-based model that is flexible to estimate the likelihood of ego-subgraph present in the network. We introduce a subgraph score that is a surrogate of the gradient by the change of likelihood between the observed and neighbor ego subgraph in the Subgraph State-Space. The subgraph score-matching objective help GAD-EBM to differentiate between the normal and anomaly nodes. With extensive experimental analysis, we have shown that GAD-EBM achieves superior performance compared with SOTA baselines in real-world benchmark graph anomaly detection datasets.

7 Acknowledgement

AR and PL were supported partially by the Sony Award and the NSF grant IIS-2239565. The authors would like to greatly thank Prof. Sharon (Yixuan) Li and anonymous reviewers for their insightful suggestions to improve the paper.

References

[1] Sambaran Bandyopadhyay, N Lokesh, and M Narasimha Murty. Outlier aware network embedding for attributed networks. In *AAAI*, 2019.

[2] Sambaran Bandyopadhyay, Saley Vishal Vivek, and MN Murty. Outlier resistant unsupervised deep architectures for attributed network embedding. In *WSDM*, 2020.

[3] Derrick Blakely, Jack Lanchantin, and Yanjun Qi. Time and space complexity of graph convolutional networks. *Accessed on: Dec*, 31, 2021.

[4] Tianyi Chen and Charalampos Tsourakakis. Antibenford subgraphs: Unsupervised anomaly detection in financial networks. In *KDD*, 2022.

[5] Dawei Cheng, Yujia Ye, Sheng Xiang, Zhenwei Ma, Ying Zhang, and Changjun Jiang. Anti-money laundering by group-aware deep graph learning. *TKDE*, 2023.

[6] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. Deep anomaly detection on attributed networks. In *SDM*. SIAM, 2019.

[7] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *CIKM*, 2020.

[8] Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. *NeurIPS*, 2020.

[9] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *NeurIPS*, 2019.

[10] Andrew Elliott, Mihai Cucuringu, Milton Martinez Luaces, Paul Reidy, and Gesine Reinert. Anomaly detection in networks with application to financial transaction networks. *arXiv preprint arXiv:1901.00402*, 2019.

[11] Haoyi Fan, Fengbin Zhang, and Zuoyong Li. Anomalydae: Dual autoencoder for anomaly detection on attributed networks. In *ICASSP*. IEEE, 2020.

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[13] Nicholas A Heard, David J Weston, Kiriaki Platanioti, and David J Hand. Bayesian anomaly detection methods for social networks. *The Annals of Applied Statistics*, 2010.

[14] Xuanwen Huang, Yang Yang, Yang Wang, Chunping Wang, Zhisheng Zhang, Jiarong Xu, Lei Chen, and Michalis Vazirgiannis. Dgraph: A large-scale financial dataset for graph anomaly detection. *NeurIPS*, 2022.

[15] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 2005.

[16] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *NeurIPS*, 2021.

[17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *NeurIPS Workshop on Bayesian Deep Learning*, 2016.

[19] Bishal Lakha, Sara Lilly Mount, Edoardo Serra, and Alfredo Cuzzocrea. Anomaly detection in cybersecurity events through graph neural network and transformer based model: A case study with beth dataset. In *Big Data*. IEEE, 2022.

[20] Zengyi Li, Yubei Chen, and Friedrich T Sommer. Learning energy-based models in high-dimensional spaces with multi-scale denoising score matching. *arXiv preprint arXiv:1910.07762*, 2019.

[21] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. *TKDD*, 2012.

[22] Kay Liu, Yingtong Dou, Yue Zhao, Xueying Ding, Xiyang Hu, Ruitong Zhang, Kaize Ding, Canyu Chen, Hao Peng, Kai Shu, et al. Bond: Benchmarking unsupervised outlier node detection on static attributed graphs. In *NeurIPS Datasets and Benchmarks Track*, 2022.

[23] Meng Liu, Keqiang Yan, Bora Oztekin, and Shuiwang Ji. Graphebm: Molecular graph generation with energy-based models. *EBM Workshop ICLR*, 2021.

[24] Nan Liu, Yilun Du, Shuang Li, Joshua B Tenenbaum, and Antonio Torralba. Unsupervised compositional concepts discovery with text-to-image generative models. *ICCV*, 2023.

[25] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *TKDE*, 2021.

[26] Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. *NeurIPS*, 2022.

[27] Jerzy Neyman and Egon Sharpe Pearson. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 1933.

[28] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. In *AAAI*, 2020.

[29] Yulong Pei, Fang Lyu, Werner Van Ipenburg, and Mykola Pechenizkiy. Subgraph anomaly detection in financial transaction networks. In *ICAIF*, 2020.

[30] Yixuan Qiu, Lingsong Zhang, and Xiao Wang. Unbiased contrastive divergence algorithm for training energy-based latent variable models. In *ICLR*, 2019.

[31] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *MLSDA*, 2014.

[32] David Savage, Xiuzhen Zhang, Xinghuo Yu, Pauline Chou, and Qingmai Wang. Anomaly detection in online social networks. *Social networks*, 2014.

[33] Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

[34] Michele Starnini, Charalampos E Tsourakakis, Maryam Zamanipour, André Panisson, Walter Allasia, Marco Fornasiero, Laura Li Puma, Valeria Ricci, Silvia Ronchiadin, Angela Ugrinoska, et al. Smurf-based anti-money laundering in time-evolving transaction networks. In *ECML PKDD*. Springer, 2021.

[35] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In *CVPR*, 2021.

[36] Kevin Swersky, Marc'Aurelio Ranzato, David Buchman, Nando D Freitas, and Benjamin M Marlin. On autoencoders and score matching for energy based models. In *ICML*, 2011.

[37] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *ICML*, 2022.

[38] Véronique Van Vlasselaer, Tina Eliassi-Rad, Leman Akoglu, Monique Snoeck, and Bart Baesens. Gotcha! network-based fraud detection for social security fraud. *Management Science*, 2017.

[39] Hung Vu, Dinh Phung, Tu Dinh Nguyen, Anthony Trevors, and Svetha Venkatesh. Energy-based models for video anomaly detection. *arXiv preprint arXiv:1708.05211*, 2017.

[40] Cheng Wang and Hangyu Zhu. Wrongdoing monitor: a graph-based behavioral anomaly detection in cyber security. *IEEE Transactions on Information Forensics and Security*, 2022.

[41] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. A semi-supervised graph attentive network for financial fraud detection. In *ICDM*. IEEE, 2019.

[42] Qi Wang, Weiliang Zhao, Jian Yang, Jia Wu, Chuan Zhou, and Qianli Xing. Atne-trust: Attributed trust network embedding for trust prediction in online social networks. In *ICDM*. IEEE, 2020.

[43] Qitian Wu, Yiting Chen, Chenxiao Yang, and Junchi Yan. Energy-based out-of-distribution detection for graph neural networks. In *ICLR*, 2023.

[44] Sheng Xiang, Mingzhi Zhu, Dawei Cheng, Enxia Li, Ruihui Zhao, Yi Ouyang, Ling Chen, and Yefeng Zheng. Semi-supervised credit card fraud detection via attribute-driven graph representation. In *AAAI*, 2023.

[45] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas AJ Schweiger. Scan: a structural clustering algorithm for networks. In *KDD*, 2007.

[46] Junting Ye and Leman Akoglu. Discovering opinion spammer groups by network footprints. In *ECML PKDD*. Springer, 2015.

[47] Rose Yu, Huida Qiu, Zhen Wen, ChingYung Lin, and Yan Liu. A survey on social media anomaly detection. *ACM SIGKDD Explorations Newsletter*, 2016.

8 Appendix

Proof of **Theorem** 1.

$$\mathcal{L}_{SSM}(\theta) = \sum_{v_G \in \mathcal{G}^{\text{sub}}} p_{\text{data}}(v_G)||b_\theta(v_G) - b_{p_{\text{data}}}(v_G)||_2^2$$

$$\begin{aligned}
&\arg\min_\theta \mathcal{J}_{\text{SSM}}(\theta) \\
&= \arg\min_\theta \sum_{v_G \in \mathcal{G}^{\text{sub}}} p_{\text{data}}(v_G)||b_\theta(v_G) - b_{p_{\text{data}}}(v_G)||_2^2 \\
&= \arg\min_\theta \sum_{v_G \in \mathcal{G}^{\text{sub}}} p_{\text{data}}\,(v_G)[\|b_\theta(v_G))\|_2^2 - 2b_\theta(v_G)b_{p_{\text{data}}}(v_G) + \|b_{p_{\text{data}}}(v_G)\|_2^2] \\
&= \arg\min_\theta \sum_{v_G \in \mathcal{G}^{\text{sub}}} p_{\text{data}}\,(v_G)[\|b_\theta(v_G)\|_2^2 - 2b_\theta(v_G)b_{p_{data}}(v_G)] \\
&= \arg\min_\theta \underbrace{\sum_{v_G \in \mathcal{G}^{\text{sub}}} p_{\text{data}}(v_G)\left(\|b_\theta(v_G)\|_2^2 + 2\|b_\theta(v_G)\|\right)}_{\mathcal{J}_1} - \underbrace{\sum_{v_G \in \mathcal{G}^{\text{sub}}} ||b_\theta(v_G)|| \sum_{u_G \in \mathcal{N}_{v_G}} 2p_{\text{data}}(u_G)}_{\mathcal{J}_2} \\
&= \arg\min_{\mathcal{J}_{\text{SSM}}(\theta)}
\end{aligned}$$

Time Complexity of GAD-EBM: For each node v we need to calculate the subgraph score for its ego neighborhood. To do that, we calculate the $p_{data}(v_G)$ for the observed ego-neighborhood for node v and also $p_{data}(u_G)$ for the ego-neighborhood for the l neighbors in the Subgraph State-Space. To generate a neighborhood, we can produce it by randomly adding or removing c percentage of edges from the observed subgraph which can be done in $O(N^2)$. The feature shuffling can be done by shuffling the feature matrix which can be done in $O(N)$. Next, we run GNN to calculate the $p_{data}(v_G)$ and $p_{data}(u_G)$ for observed and neighborhood subgraphs which can be parallelized and takes the time complexity $O(LNF^2 + L|E|F)$ where L is the number of layers in the graph neural network, N is the number of nodes, $|E|$ is the size of the edge set and F is the size of the dimension [3]. Since we are only dealing with ego-neighborhood or a 1^{st} order neighborhood so number of layers in the GNN is 1 which results in a fast pipeline of GAD-EBM.