Supplemental material

I. Derivation of Theorem 1

Theorem 1. Given a graph $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{X})$ containing structural anomalies, where the node set $\mathcal{V}=\{v_1,\ldots,v_m\}$ and the edge set $\mathcal{E}=\{e_1,\ldots,e_t\}$, the model performs a random sampling with put-back of edges in $\mathcal{E}$, taking k edges per round, for a total of r rounds of sampling. After r rounds of sampling, the relationship between the number of edges k sampled by the model in each round and the expectation $E(m)$ of the proportion of different nodes sampled in total in r rounds as: $E(m)=1-\left(\frac{t-k}{t}\right)^r$.

Derivation. The sampling of edges corresponds to the structure disturbance operation of the model, and the expectation of the number of different edges perceived after the $(r-1)$-th round of model disturbance is calculated as follows:

$$E(r-1)=\sum_{\alpha=k}^{t}\alpha\times P(r-1,\alpha) \tag{1}$$

where $P(r-1,\alpha)$ denotes the probability that the model has perceived α different edges. Based on Eq.(1), the expectation of the number of different edges perceived after the r-th round of model disturbance is calculated as follows:

$$\begin{aligned}E(r)&=\sum_{\alpha=k}^{t}\sum_{\beta=0}^{k}(\alpha+k-\beta)\times P(r-1,\alpha)\times\frac{C_\alpha^\beta C_{t-\alpha}^{k-\beta}}{C_t^k}\\&=\sum_{\alpha=k}^{t}(\alpha+k)\times P(r-1,\alpha)-\\&\quad\sum_{\alpha=k}^{t}P(r-1,\alpha)\sum_{\beta=0}^{k}\beta\times\frac{C_\alpha^\beta C_{t-\alpha}^{k-\beta}}{C_t^k}\end{aligned} \tag{2}$$

where β denotes the number of repeatedly disturbed edges among the edges perceived by the model in round r from round $(r-1)$-th, and the relationship between the expectation that the model has perceived different numbers of edges after r-th round and $(r-1)$-th round can be obtained from Eq.(1) and Eq.(2) as follows:

$$\begin{aligned}E(r)&=E(r-1)+k-\sum_{\alpha=k}^{t}P(r-1,\alpha)\times k\times\frac{\alpha}{t}\\&=E(r-1)+k-\frac{k}{t}E(r-1)\\&=\left(1-\frac{k}{t}\right)E(r-1)+k\end{aligned} \tag{3}$$

Further the relationship between k and $E(r)$ for the universal case can be obtained as follows.

$$\begin{aligned}E(r)&=t-\left(\frac{t-k}{t}\right)^{r-1}(t-k)\\&=t-\left(\frac{t-k}{t}\right)^{r}\frac{t}{t-k}(t-k)\\&=t-\left(\frac{t-k}{t}\right)^{r}t\end{aligned} \tag{4}$$

In addition, Eq.(4) reflects the relationship between r and $E(r)$ in the special case. When $r=1$, $E(r)=k$, and when $r\to\infty$, $E(r)\to t$. The average degree of the nodes corresponding to $\mathcal{G}$ is calculated as follows.

$$d=\frac{m}{t} \tag{5}$$

Considering that the target objects of anomaly detection are nodes, we calculate the expected $E(s)$ corresponding to the number of different nodes perceived by the r-th disturbance based on the expected $E(r)$ of the number of different edges perceived by the r-th disturbance of the model combined with the average node degree d as follows.

$$E(s)=\frac{E(r)}{d} \tag{6}$$

Finally, the traversal expectation $E(m)$ of the model's r-round structure disturbance on the nodes in $\mathcal{V}$ is calculated based on the number m of nodes in the node set $\mathcal{V}$ of $\mathcal{G}$ as follows.

$$E(m)=\frac{E(s)}{m}=\frac{t-\left(\frac{t-k}{t}\right)^r t}{t}=1-\left(\frac{t-k}{t}\right)^r \tag{7}$$

For a graph with t edges, Eq.(7) shows the relationship between the number of edges k sampled by the model in each round and the expectation $E(m)$ of the proportion of different nodes perceived by r rounds of disturbance.

II. Overall Procedure

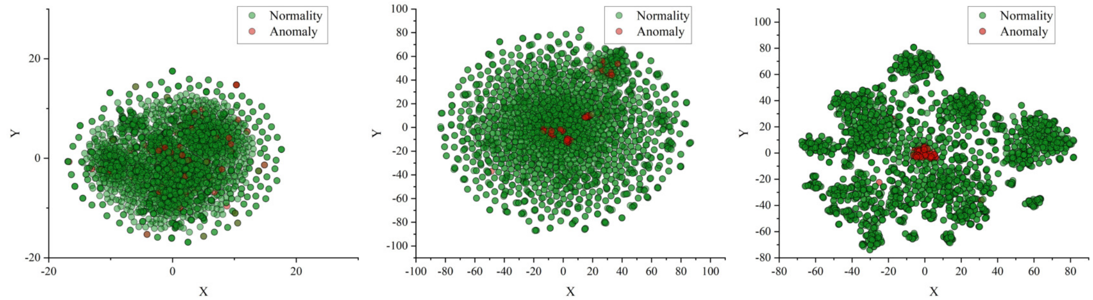
The overall procedure of GDSA framework is depicted in Algorithm 1.

$$\mathbf{E}_l=GCN\left(\mathbf{A}_l,\mathbf{X}_l\right) \tag{8}$$

$$\begin{cases}\mathbf{e}_l=\sigma\left(\mathbf{E}_l\right)\\ s_i=MLP\left(\mathbf{E}_l\right)=\sum_{j=1}^{h}\mathbf{e}_l[i,j]\end{cases} \tag{9}$$

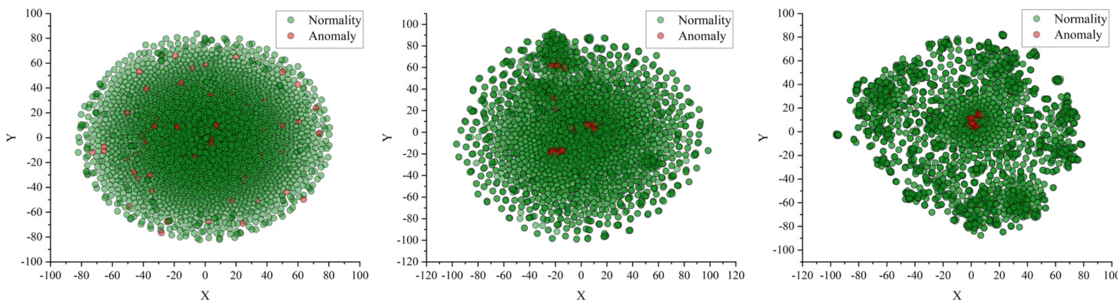
$$\mathcal{L}_{BCE}=\frac{1}{2m}\left(\sum_{i=1}^{2m}y_i\log s_i+(1-y_i)\log(1-s_i)\right) \tag{10}$$

$$c_i=\frac{1}{(1+e^{-s_i})} \tag{11}$$



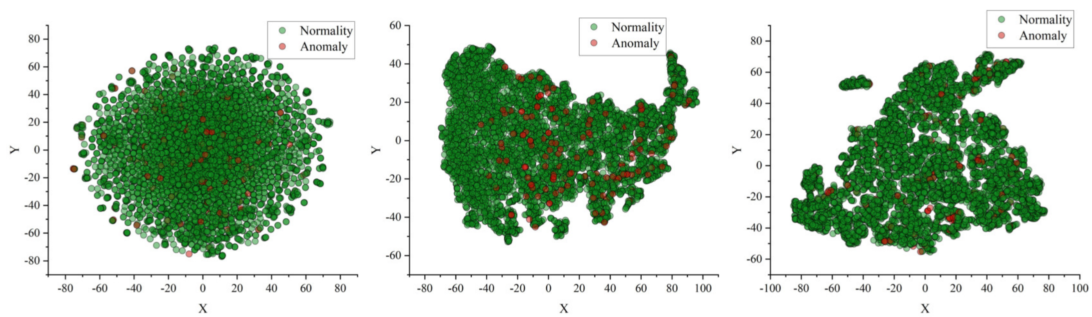
(a) The initial feature. (b) GDSA w/o both. (c) GDSA.

Fig. 1. Visualization of the feature distribution on the Cora dataset.



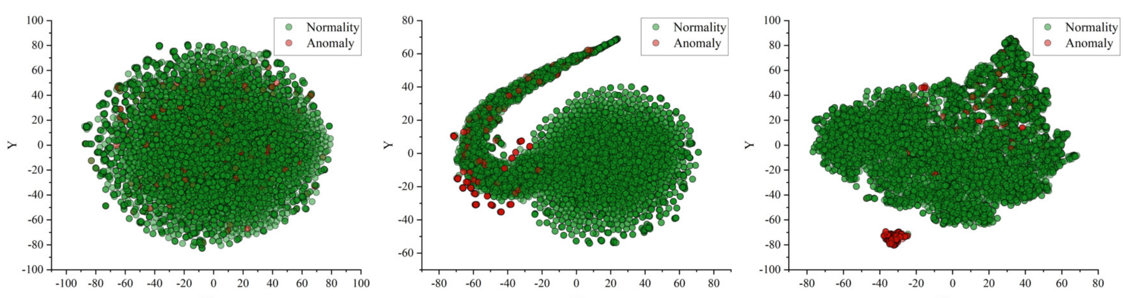
(a) The initial feature. (b) GDSA w/o both. (c) GDSA.

Fig. 2. Visualization of the feature distribution on the Citeseer dataset.



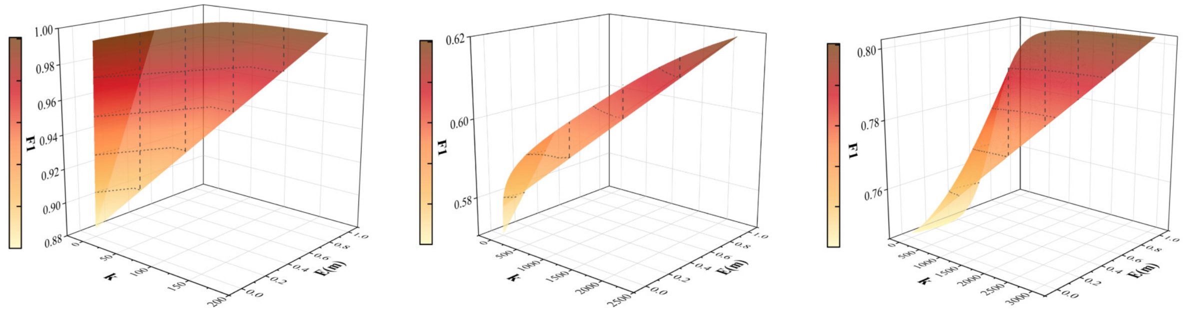
(a) The initial feature. (b) GDSA w/o both. (c) GDSA.

Fig. 3. Visualization of the feature distribution on the BlogCatalog dataset.



(a) The initial feature. (b) GDSA w/o both. (c) GDSA.

Fig. 4. Visualization of the feature distribution on the Flickr dataset.



(a) Citeseer (b) BlogCatalog (c) Flickr

Fig. 5. The relationship between k, $E(m)$ and F1 on the datasets.

Algorithm 1 The Overall Procedure of GDSA

Input: $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, Predefined proportion for feature augmentation: w, Number of edges per round of disturbance in *view 2* structure disturbance: k, Number of training epochs: R.

Output: The vector $\mathbf{s}$ consisting of the anomaly values of all nodes.

```
The trainable parameters are initialized randomly.
// Training phase.
for r ∈ 1, 2, ..., R do
    Obtain the positive group attribute matrix X_Al by feature augmentation of the attribute matrix X in G.
    Obtain the negative group attribute matrix X_Cl by performing view 1 structure disturbance based on X_Al.
    Obtain the adjacency matrix A_l by performing view 2 structure disturbance based on A in G.
    Calculate representations E_Al and E_Cl with Encoder module via Eq.(8).
    Calculate scalar information s_i with Readout module via Eq.(9).
    Calculate L_BCE via Eq.(10).
    Update parameters by backpropagation.
end for
// Testing phase.
Calculate representations E_l with Encoder module based on A and X in G via Eq.(8).
Calculate anomaly values (scalar information) s_i with Readout module based on E_l via Eq.(9).
(optional) Calculate anomaly scores c_i with sigmoid function based on s_i via Eq.(11).
return anomaly values s = (s_1, ..., s_n),
       (optional) anomaly scores c = (c_1, ..., c_n).
```

III. Complexity analysis

The time complexity of GDSA is mainly composed of two components, the Encoder-Readout module and the group discrimination. For graph $\mathcal{G}$ with m nodes and t edges, in GDSA, we analyze the Encoder component by choosing GCN and the Readout component by choosing MLP as an example. The Encoder component needs to perform feature extraction for the attribute matrix of positive and negative groups (twice calculation), and the time complexity of this process is $O(m+t)$. The Readout component is used to process the output of the Encoder. When the number of linear layers of MLP is set to L_l and the size of hidden layers is set to L_h, the time complexity of L_l layer is calculated as $O(L_l m L_h)$. In this part, the time complexity of our aggregation of the generated embedding by simple summation is $O(mL_h)$. Thus, the time complexity of the Encoder-Readout module part is $O(2(m + t + L_l m L_h + mL_h))$.

The time complexity of the group discrimination part depends only on the BCE loss calculation shown in Eq.(10) (Eq.3 in the *Paper*), which implements the computation of the scalar information of positive and negative group nodes (a total of $2m$ nodes) with the time complexity of $O(2m)$. Therefore, the time complexity of GDSA to compute graph $\mathcal{G}$ is $O(2(m + t + L_l m L_h + mL_h + m)) \rightarrow O(m + t + L_l m L_h + mL_h + m) \rightarrow O(L_l m L_h)$. It can be seen that GDSA reduces the time complexity of the loss calculation part of graph anomaly detection to $O(1)$, and the time complexity of the final model mainly depends on the Encoder-Readout module.

TABLE I

The total number of anomalies and statistics of the datasets, the first four datasets are benchmark datasets, and the last one is a large-scale dataset.

Dataset	Nodes	Edges	Attributes	Anomalies
Cora	2,708	5,429	1,433	75
Citeseer	3,327	4,732	3,703	75
BlogCatalog	5,196	171,743	8,189	150
Flickr	7,575	239,738	12,407	225
ogbn-arxiv	169,343	1,166,243	128	3,000

IV. Dataset statistics

We generate the abnormal nodes in the real-world dataset by selecting η_1 nodes at a time and making them fully connected, repeating this process η_2 times so that the total number of structural anomalies is $\eta_1 \times \eta_2$. Table I summarizes the total number of anomalies and statistics for the datasets.

V. IMPLEMENTATION DETAILS

Parameter settings. For all datasets, the number of convolution layers in GNNs is set to 1, and the number of linear layers in the projector is set to 1. For the four benchmark datasets, we set the hidden layers in GCN to 512, the learning rate to 1e-3, and the epoch to 400. For the large-scale dataset ogbn-axiv, we set the hidden layers in GCN to 1500 and the learning rate to 5e-5.

Computing infrastructure. For the four benchmark datasets, we used NVIDIA Tesla T4 (16GB memory) and Intel Xeon Gold 6248 with 8 cores for all their experiments. For the large-scale dataset (ogbn-arxiv), we used NVIDIA GeForce RTX 3090 (24GB memory) and Intel Xeon Platinum 8255C with 12 cores.

VI. VISUALIZATION OF FEATURE DISTRIBUTION

The visualization of the initial feature distribution, the feature distribution after being processed by GDSA w/o both and the feature distribution after being processed by GDSA model on the benchmark datasets is shown in Fig.1 to Fig.4.

VII. PARAMETER VISUALIZATION

The visualization of the relationship between k, $E(m)$ and F1 on the datasets (Citeseer, BlogCatalog and Flickr) is shown in Fig.5.