Leveraging Diffusion Model as Pseudo-Anomalous Graph Generator for Graph-Level Anomaly Detection,

Jinyu Cai[1], Yunhe Zhang [2†], Fusheng Liu[1], See−Kiong Ng[1]

[1] Institute of Data Science, National University of Singapore, Singapore

[2] Department of Computer and Information Science, SKL-IOTSC, University of Macau, China †Corresponding Author

Paper:

Motivation

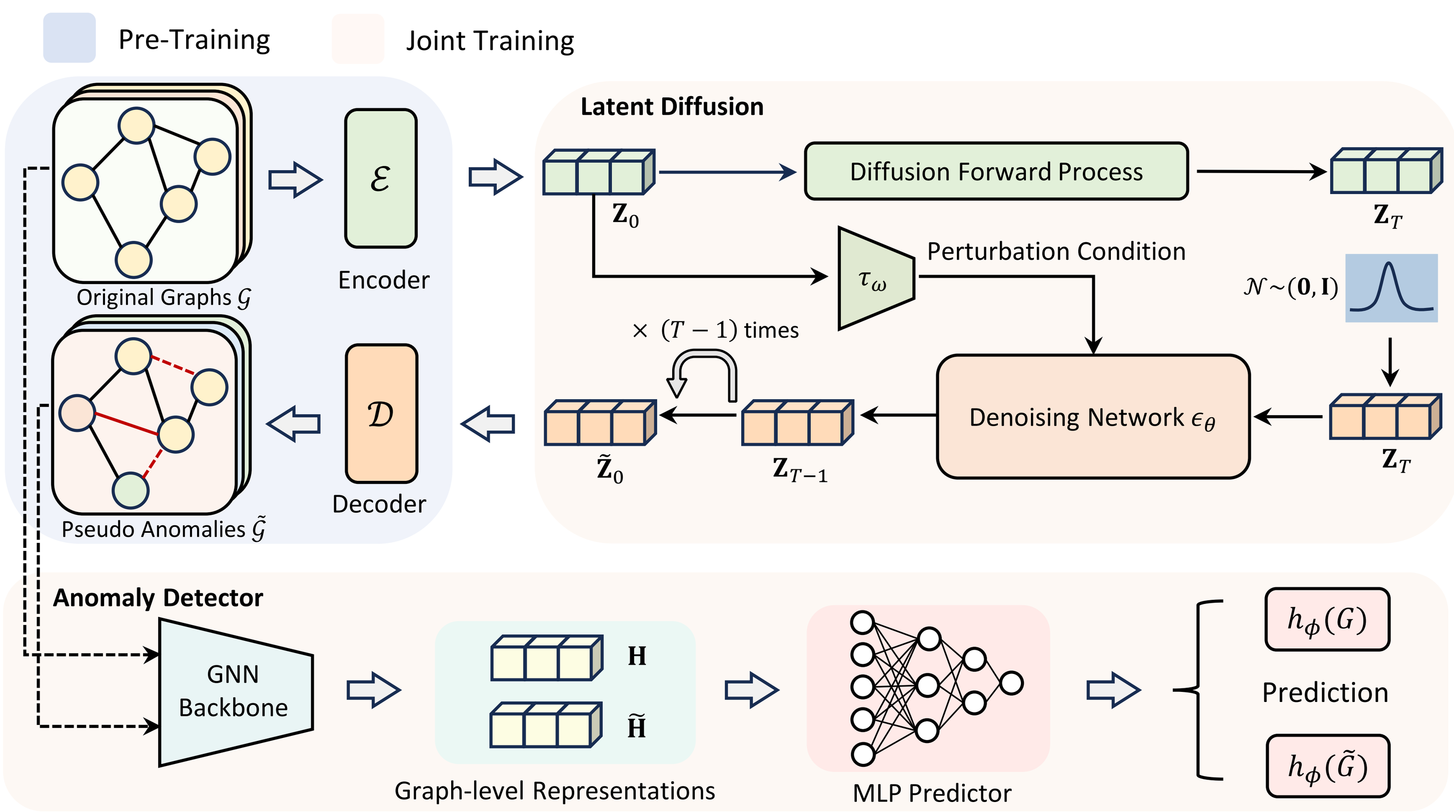
1. Unsupervised GLAD methods generally focus on modelling normal graph distributions, which struggles to identify subtle anomalies, especially those near the boundaries of normal graphs.
2. Semi-supervised GLAD methods can leverage limited labelled anomalies to enhance decision boundary learning. However, their effectiveness is constrained by the scarcity and diversity of labelled anomalous graph.

Contribution

1. We introduce AGDiff, the first framework that explores the potential of diffusion models to mitigate the anomaly scarcity challenge in GLAD.
2. We propose a latent diffusion process with perturbation conditions to generate pseudo-anomalous graphs without relying on any labelled anomalies for improving decision boundary learning.
3. We demonstrate the effectiveness of AGDiff across extensive comparisons with state-of-the-art GLAD baselines on diverse graph benchmarks.

Figure 1. An illustration of the proposed AGDiff framework.

The framework consists of three main components:
(1) Pre-train model;
(2) Latent diffusion-based graph generation model;
(3) Anomaly detector.



Solution

(1) Modeling Normality via Variational Inference:

We first pre-train a graph representation learning model aiming at capturing the normality of graphs.

Variational posterior: $q(\mathbf{Z}|\mathbf{X},\mathbf{A}) = \prod_{i=1}^{n} q(\mathbf{z}_i|\mathbf{X},\mathbf{A}),$

w.r.t $q(\mathbf{z}_i|\mathbf{X},\mathbf{A}) = \mathcal{N}(\mathbf{z}_i|\boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}_i^2)),$

Reconstruction: $\hat{\mathbf{A}} = \mathcal{T}(\mathbf{Z}\mathbf{Z}^\top), \quad \hat{\mathbf{X}} = \mathcal{D}(\mathbf{Z}),$

Pre-train loss:

$$\mathcal{L}_{\text{pretrain}} = \ell_{\text{r}}^{\text{attr}} + \ell_{\text{r}}^{\text{edge}} + \ell_{\text{KL}} = \sum_{i=1}^{N} (\|\mathbf{X}_i - \hat{\mathbf{X}}_i\|_F^2 + \mathcal{H}(\mathbf{A}_i, \hat{\mathbf{A}}_i) - \mathrm{KL}(q(\mathbf{Z}_i|\mathbf{X}_i,\mathbf{A}_i)|\mathcal{P}(\mathbf{Z}))),$$

(2) Generating Anomalous Graphs via Latent Diffusion:

Building on a well-structured latent space that effectively captures normal graph patterns, we propose a novel approach that utilizes latent diffusion models to generate pseudo-anomalous graphs.

Forward diffusion process: $\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(\mathbf{0},\mathbf{I}),$

Reverse denoising process: $\mathbf{z}_{t-1} = \frac{1}{\sqrt{\alpha}}\left(\mathbf{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{z}_t, t, \mathbf{c})\right) + \tilde{\beta}\mathbf{v},$

The condition vector $\boldsymbol{c}$ is obtained via a perturbation condition model τ_ω to add auxiliary noise information to the generation process:

Perturbation condition: $\mathbf{c} = \tau_\omega(\mathbf{z}_0) = \sigma(\mathbf{W_c}(\mathbf{z}_0 + \eta) + \mathbf{b_c}),$

$\eta \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ is a Gaussian noise vector that introduces perturbations to the initial latent representation, $\tau_\omega(\cdot)$ transforms the perturbed representation to a more expressive feature space.

Loss of the latent diffusion model: $\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathbf{z}_0,\epsilon,t,\mathbf{c}}\left[\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t, \mathbf{c})\|_2^2\right].$

(3) Detecting Anomalies from Subtle Deviations:

We employ a GIN-based anomaly detector $h_\varphi(\cdot)$ to distinguish between normal graphs and pseudo-anomalous graphs, and adopt a following binary cross-entropy loss $\mathcal{L}_{\text{cls}}$ to train the anomaly detector:

$$h_\phi(G) = \mathrm{MLP}(\mathrm{GIN}(\mathbf{X},\mathbf{A})), \qquad \mathcal{L}_{\text{cls}} = -\frac{1}{|\mathcal{G}\cup\tilde{\mathcal{G}}|}\sum_{G\in\mathcal{G}\cup\tilde{\mathcal{G}}}(y_G \log h_\phi(G) + (1-y_G)\log(1-h_\phi(G))),$$

(4) Joint Training:

Total loss: $\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda\mathcal{L}_{\text{diff}}$

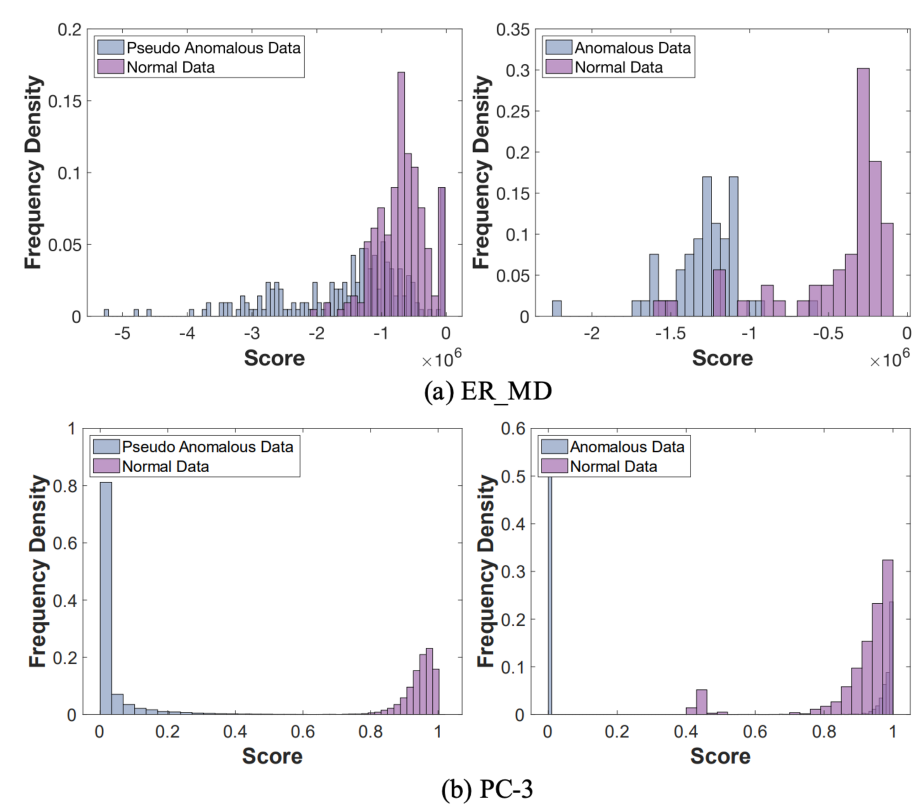
1. The latent diffusion model learns to generate increasingly challenging pseudo-anomalies that explore the decision boundary of the anomaly detector.
2. The gradient of the detector directs the diffusion process toward generating more informative pseudo-anomalous samples.
3. The iterative refinement between generation and detection leads to a more robust anomaly detector.

Comparison Results

Table 1. Average AUCs and F1-Scores with standard deviation (10 trials) on four small and moderated graph datasets. The best results are marked in **bold**, and "OM" denotes out-of-memory.

Method	MUTAG		DD		COX2		ER_MD	
	AUC	F1-Score	AUC	F1-Score	AUC	F1-Score	AUC	F1-Score
SP (Borgwardt & Kriegel, 2005)	67.52±0.00	60.00±0.00	82.73±0.00	76.09±0.00	54.08±0.00	49.32±0.00	40.92±0.00	37.74±0.00
WL (Shervashidze et al., 2011)	60.00±0.00	89.12±0.00	81.57±0.00	74.64±0.00	49.32±0.00	50.19±0.00	37.74±0.00	45.71±0.00
NH (Hido & Kashima, 2009)	79.97±0.40	76.00±0.00	81.61±0.32	73.91±0.65	61.41±0.82	56.44±1.03	51.55±2.00	50.19±0.92
RW (Vishwanathan et al., 2010)	86.98±0.00	83.33±0.00	OM	OM	52.43±0.00	30.00±0.00	78.94±0.00	65.96±0.00
OCGIN (Zhao & Akoglu, 2023)	74.66±1.68	62.95±0.00	66.59±4.44	56.12±0.00	59.64±5.78	47.95±0.00	47.63±3.59	50.94±1.89
OCGTL (Qiu et al., 2022)	87.04±1.74	80.00±0.00	77.52±0.43	71.65±0.73	60.42±0.90	55.62±5.24	72.67±0.20	67.17±0.92
GLocalKD (Ma et al., 2022)	90.59±0.61	86.17±0.91	80.59±0.00	73.48±0.57	51.42±0.66	51.24±0.60	78.94±0.00	70.21±0.00
iGAD (Zhang et al., 2022)	92.58±1.25	85.20±2.30	74.83±2.30	70.39±2.60	72.09±2.29	61.94±1.09	80.56±2.57	74.57±2.45
SIGNET (Liu et al., 2023a)	87.73±2.45	73.07±4.11	59.53±3.45	56.76±3.47	52.80±2.53	20.24±4.92	77.02±1.07	77.06±1.70
MUSE (Kim et al., 2024)	83.81±5.17	75.36±5.02	61.06±3.03	58.32±3.08	54.14±3.23	52.14±3.49	31.07±4.58	35.67±4.68
DO2HSC (Zhang et al., 2024)	88.83±6.58	86.80±6.21	77.12±2.15	70.87±2.73	63.16±3.36	58.36±2.95	68.31±4.31	66.63±3.04
AGDiff	**95.83±2.15**	**89.45±1.37**	**88.23±0.67**	**84.06±0.59**	**77.59±3.39**	**68.15±1.49**	**91.21±1.84**	**86.04±2.26**

Scoring Distribution



Visualization Results