Energy-based Out of Distribution Detection for Graph Neural Networks

Presenter: Jiaqing Xie

2024.06.07

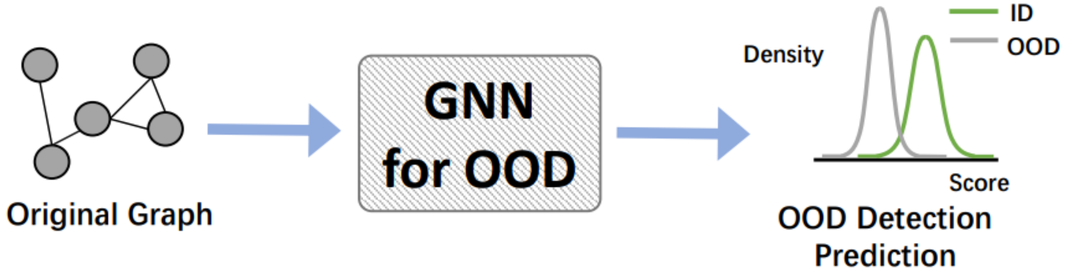
Traditional Graph Learning vs. Graph OOD Detection

V: Vertexes, E: Edges, A: Adjacency Matrix

Graph G = (V, E, X), X: node features

Graph Neural Network F(A, X)

Traditional Graph Learning
Task: Node / Graph Classification, …



Graph OOD Detection
Task: Binary Classification

A GNN Baseline (Example: GCN)

- Layer-wise Propagation

$$Z^{(l)} = \sigma\left(D^{-1/2}\tilde{A}D^{-1/2}Z^{(l-1)}W^{(l)}\right), \quad Z^{(l-1)} = [\mathbf{z}_i^{(l-1)}]_{i\in\mathcal{I}}, \quad Z^{(0)} = X$$

$$h_\theta(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i}) = \mathbf{z}_i^{(L)}$$ (dim = C)

- GNN Classifier (softmax)

$$p(y \mid \mathbf{x}, \mathcal{G}_{\mathbf{x}}) = \frac{e^{h_\theta(\mathbf{x},\mathcal{G}_{\mathbf{x}})_{[y]}}}{\sum_{c=1}^{C} e^{h_\theta(\mathbf{x},\mathcal{G}_{\mathbf{x}})_{[c]}}}.$$

Limitation of Softmax for OOD Detection

- In Image Domain, directly use the predictions of CNNs for OOD detection would lead to overconfidence of OOD data [Nguyen et al.].
- In Graph Domain, this overconfidence also exists for Graph OOD data [Wu et al.].
- Motivation : Use of Energy function, which is proved to be aligned with probability density of input data [Liu et al.].

Deep Neural Networks are easily fooled. Nguyen et al. CVPR 2015

Energy-based Out of Distribution Detection. Liu et al. NeurIPS 2020

Energy-based Out of Distribution Detection for Graph Neural Networks. Wu et al. ICLR 2023

Motivation 1: Use of Energy

- Energy

$$E(\mathbf{x}, \mathcal{G}_{\mathbf{x}}, y; h_\theta) = -h_\theta(\mathbf{x}, \mathcal{G}_{\mathbf{x}})_{[y]}$$

- Free energy function

$$E(\mathbf{x}, \mathcal{G}_{\mathbf{x}}; h_\theta) = -\log \sum_{c=1}^{C} e^{h_\theta(\mathbf{x}, \mathcal{G}_{\mathbf{x}})_{[c]}}$$

- Loss Objective (Original Predictor)

$$\mathcal{L}_{sup} = \mathbb{E}_{(\mathbf{x}, \mathcal{G}_{\mathbf{x}}, y) \sim \mathcal{D}_{in}} \left(-\log p(y \mid \mathbf{x}, \mathcal{G}_{\mathbf{x}})\right)$$

$$= \sum_{i \in \mathcal{I}_s} \left(-h_\theta(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i})_{[y_i]} + \log \sum_{c=1}^{C} e^{h_\theta(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i})_{[c]}} \right)$$

Original Predictor

$$p(y \mid \mathbf{x}, \mathcal{G}_{\mathbf{x}}) = \frac{e^{h_\theta(\mathbf{x}, \mathcal{G}_{\mathbf{x}})_{[y]}}}{\sum_{c=1}^{C} e^{h_\theta(\mathbf{x}, \mathcal{G}_{\mathbf{x}})_{[c]}}}$$

OOD Predictor

$$G(\mathbf{x}, \mathcal{G}_{\mathbf{x}}; h_\theta) = \begin{cases} 1, & \text{if} \quad \tilde{E}(\mathbf{x}, \mathcal{G}_{\mathbf{x}}; h_\theta) \leq \tau \\ 0, & \text{if} \quad \tilde{E}(\mathbf{x}, \mathcal{G}_{\mathbf{x}}; h_\theta) > \tau \end{cases}$$

Motivation 2: Label Propagation

- Not all graph data are labeled

$$\mathcal{L}_{sup} = \mathbb{E}_{(\mathbf{x},\mathcal{G}_{\mathbf{x}},y)\sim\mathcal{D}_{in}}(-\log p(y \mid \mathbf{x},\mathcal{G}_{\mathbf{x}}))$$

- Graph data are inter-dependent, propagation with energy reinforces the confidence on detection

- Label Propagation, a non-parametric semi-supervised learning algorithm

Label Propagation

- Initialize Energy

Recall $$E(\mathbf{x}, \mathcal{G}_{\mathbf{x}}; h_\theta) = -\log \sum_{c=1}^{C} e^{h_\theta(\mathbf{x}, \mathcal{G}_{\mathbf{x}})_{[c]}}$$

$$\mathbf{E}^{(0)} = [E(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i}; h_\theta)]_{i \in \mathcal{I}}$$

- Belief Propagation

$$\mathbf{E}^{(k)} = \alpha \mathbf{E}^{(k-1)} + (1-\alpha) D^{-1} A \mathbf{E}^{(k-1)}, \quad \mathbf{E}^{(k)} = [E_i^{(k)}]_{i \in \mathcal{I}}$$

- OOD Detector

$$\tilde{E}(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i}; h_\theta) = E_i^{(K)}$$

$$G(\mathbf{x}, \mathcal{G}_{\mathbf{x}}; h_\theta) = \begin{cases} 1, & \text{if} \quad \tilde{E}(\mathbf{x}, \mathcal{G}_{\mathbf{x}}; h_\theta) \leq \tau, \\ 0, & \text{if} \quad \tilde{E}(\mathbf{x}, \mathcal{G}_{\mathbf{x}}; h_\theta) > \tau. \end{cases}$$

Motivation 3: Regularization

- Previous settings didn't include training Graph OOD data.
- Energy range: [t_in, t_out]
- Loss Objective $\mathcal{L}_{sup} + \lambda \mathcal{L}_{reg}$

$$\mathcal{L}_{sup} = \mathbb{E}_{(\mathbf{x}, \mathcal{G}_{\mathbf{x}}, y) \sim \mathcal{D}_{in}} \left(-\log p(y \mid \mathbf{x}, \mathcal{G}_{\mathbf{x}}) \right)$$
$$= \sum_{i \in \mathcal{I}_s} \left(-h_\theta(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i})_{[y_i]} + \log \sum_{c=1}^{C} e^{h_\theta(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i})_{[c]}} \right)$$

$$\mathcal{L}_{reg} = \underbrace{\frac{1}{|\mathcal{I}_s|} \sum_{i \in \mathcal{I}_s} \left(\mathrm{ReLU}\left(\tilde{E}\left(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i}; h_\theta\right) - t_{in} \right) \right)^2}_{\text{In distribution}} + \underbrace{\frac{1}{|\mathcal{I}_o|} \sum_{j \in \mathcal{I}_o} \left(\mathrm{ReLU}\left(t_{out} - \tilde{E}\left(\mathbf{x}_j, \mathcal{G}_{\mathbf{x}_j}; h_\theta\right) \right) \right)^2}_{\text{Out-of distribution}}$$

Experiments

Baselines:

- MSP
- ODIN
- Mahalanobis
- OE (Outlier Exposure)
- Energy
- Energy without energy propagation
- GKDE
- GPN

Experiments

Graph OOD data creation follow two settings: multi-graph and single graph

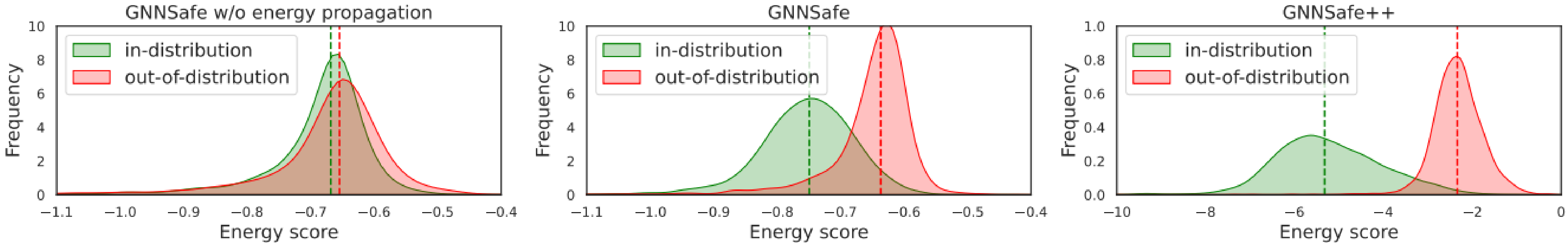
1) Multi-graph
    - Choose subgraph DE as IID, other 5 subgraphs as OOD [Twitch]
2) Single graph
    - Change structures / features / labels [Cora, Amazon, Coauthor]
    - Partition nodes (Arxiv), before 2015 IID, after 2017 OOD

Results 1

- Metrics: AUROC, AUPR, FPR95

Model	OOD Expo	Twitch AUROC	Twitch AUPR	Twitch FPR	Twitch ID ACC	Arxiv AUROC	Arxiv AUPR	Arxiv FPR	Arxiv ID ACC
MSP	No	33.59	49.14	97.45	68.72	63.91	**75.85**	**90.59**	53.78
ODIN	No	**58.16**	72.12	93.96	70.79	55.07	68.85	100.0	51.39
Mahalanobis	No	55.68	66.42	**90.13**	70.51	56.92	69.63	94.24	51.59
Energy	No	51.24	60.81	91.61	70.40	**64.20**	75.78	90.80	53.36
GKDE	No	46.48	62.11	95.62	67.44	58.32	72.62	93.84	50.76
GPN	No	51.73	66.36	95.51	68.09	-	-	-	-
GNNSAFE	No	66.82	**70.97**	76.24	70.40	71.06	80.44	87.01	53.39
OE	Yes	55.72	70.18	95.07	70.73	69.80	80.15	85.16	52.39
Energy FT	Yes	**84.50**	**88.04**	**61.29**	70.52	**71.56**	**80.47**	**80.59**	53.26
GNNSAFE++	Yes	95.36	97.12	33.57	70.18	74.77	83.21	77.43	53.50

Results 2



(a) The energy distributions on `Twitch` where nodes in different sub-graphs are OOD instances

Thanks

Motivation 3: Regularization

- Additional regularization doesn't effect NLL
- Proof:

$$\frac{e^{h_{\theta^\dagger}}(\mathbf{x}, \mathcal{G}_\mathbf{x})_{[y]}}{\sum_{c=1}^{C} e^{h_{\theta^\dagger}}(\mathbf{x}, \mathcal{G}_\mathbf{x})_{[c]}} = \mathrm{argmin}_{p(y|\mathbf{x},\mathcal{G}_\mathbf{x})} \mathbb{E}_{(\mathbf{x},\mathcal{G}_\mathbf{x},y)\in\mathcal{D}_{in}} [-\log p(y|\mathbf{x}, \mathcal{G}_x)]$$

$$\begin{aligned} E(\mathbf{x}, \mathcal{G}_\mathbf{x}; h_{\theta^*}) &= E(\mathbf{x}, \mathcal{G}_\mathbf{x}; h_{\theta^*}) - E(\mathbf{x}, \mathcal{G}_\mathbf{x}; h_{\theta^\dagger}) - \log \sum_{c=1}^{C} e^{h_{\theta^\dagger}(\mathbf{x},\mathcal{G}_\mathbf{x})} \\ &= -\log \left(e^{-E(\mathbf{x},\mathcal{G}_\mathbf{x};h_{\theta^*})+E(\mathbf{x},\mathcal{G}_\mathbf{x};h_{\theta^\dagger})} \cdot \sum_{c=1}^{C} e^{h_{\theta^\dagger}(\mathbf{x},\mathcal{G}_\mathbf{x})} \right) \\ &= -\log \sum_{c=1}^{C} e^{h_{\theta^\dagger}(\mathbf{x},\mathcal{G}_\mathbf{x})-E(\mathbf{x},\mathcal{G}_\mathbf{x};h_{\theta^*})+E(\mathbf{x},\mathcal{G}_\mathbf{x};h_{\theta^\dagger})}. \end{aligned}$$

$$\begin{aligned} p(y|\mathbf{x}, \mathcal{G}_\mathbf{x}) &= \frac{e^{h_{\theta^\dagger}(\mathbf{x},\mathcal{G}_\mathbf{x})_{[y]}-E(\mathbf{x},\mathcal{G}_\mathbf{x};h_{\theta^*})+E(\mathbf{x},\mathcal{G}_\mathbf{x};h_{\theta^\dagger})}}{\sum_{c=1}^{C} e^{h_{\theta^\dagger}(\mathbf{x},\mathcal{G}_\mathbf{x})_{[c]}-E(\mathbf{x},\mathcal{G}_\mathbf{x};h_{\theta^*})+E(\mathbf{x},\mathcal{G}_\mathbf{x};h_{\theta^\dagger})}} \\ &= \frac{e^{h_{\theta^\dagger}(\mathbf{x},\mathcal{G}_\mathbf{x})_{[y]}}}{\sum_{c=1}^{C} e^{h_{\theta^\dagger}(\mathbf{x},\mathcal{G}_\mathbf{x})_{[c]}}}. \end{aligned}$$