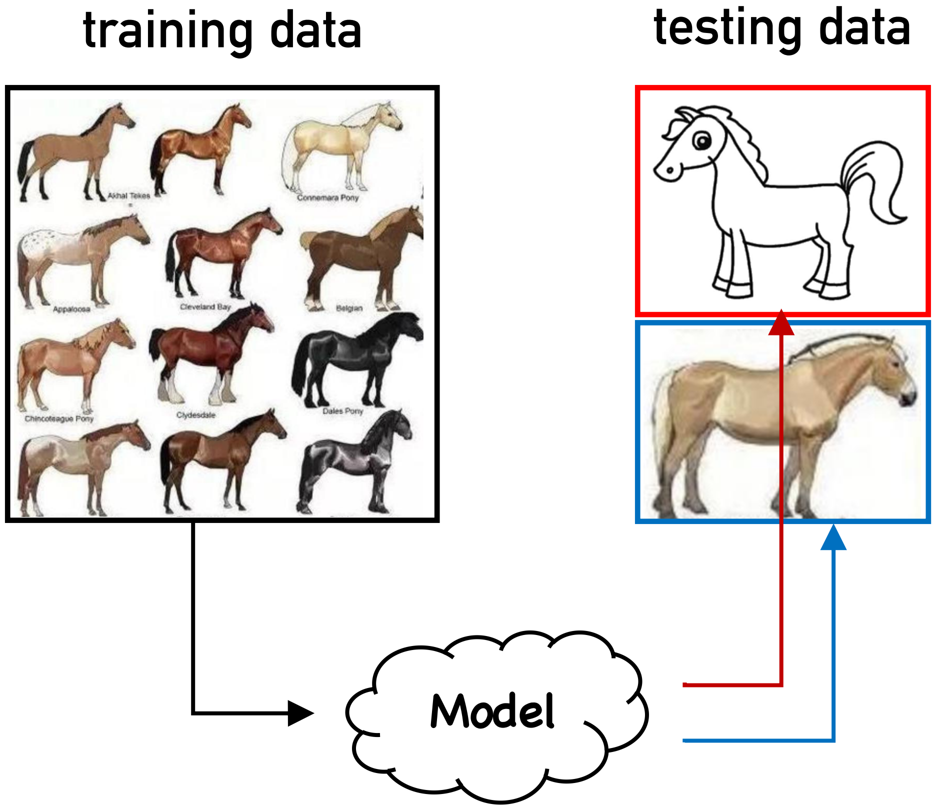
Energy-based Out-of-Distribution Detection for Graph Neural Networks

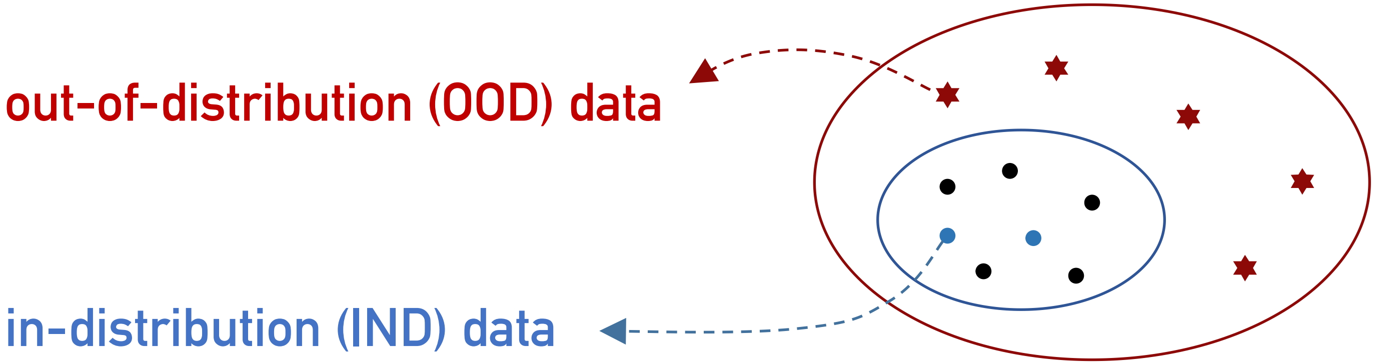
Qitian Wu, Yiting Chen, Chenxiao Yang, Junchi Yan

code

blog

Out-of-Distribution Generalization





1: perform well on IND testing data

2: perform well on OOD testing data

OOD Generalization:

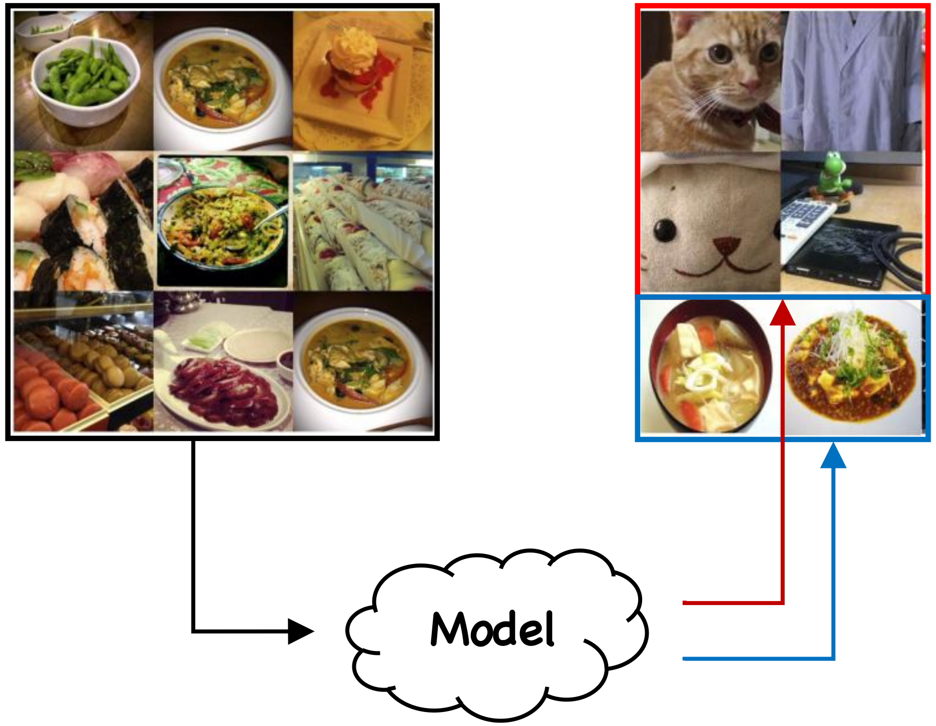
Train a robust classifier that can perform well on testing samples from disparate distributions than training data

Qitian Wu, et al., Handling Distribution Shifts on Graphs: An Invariance Perspective, in ICLR'22
Nianzu Yang, et al., Learning Substructure Invariance for Out-of-Distribution Molecular Representations, in NeurIPS'22
Chenxiao Yang et al., Towards out-of-distribution sequential event prediction: A causal treatment, in NeurIPS'22

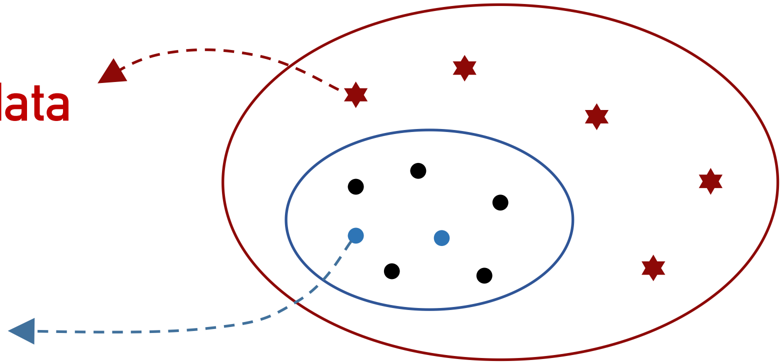
Out-of-Distribution Detection

training data

testing data



out-of-distribution (OOD) data



in-distribution (IND) data

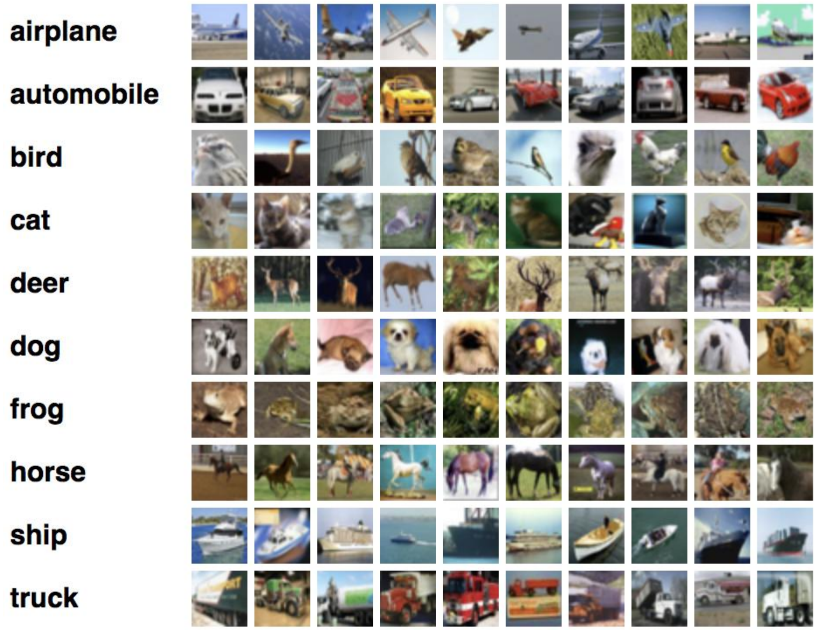
1: perform well on IND testing data

2: identify OOD testing data

OOD Detection:

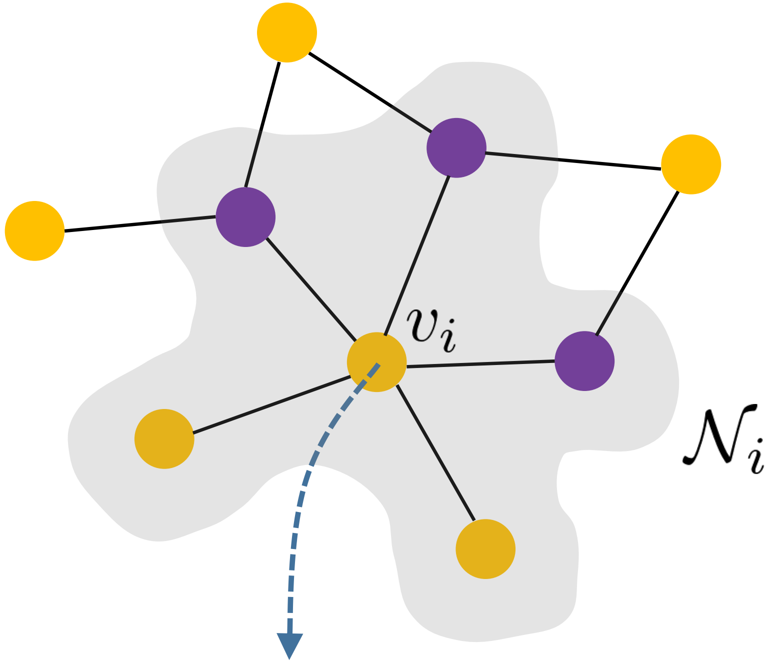
Train a robust classifier that can identify samples from disparate distributions than (in-distribution) training data

Challenges of Graph Data Modeling



$$(x_i, y_i) \sim p(x, y)$$

each instance is drawed from the same data distribution independently (i.i.d.)



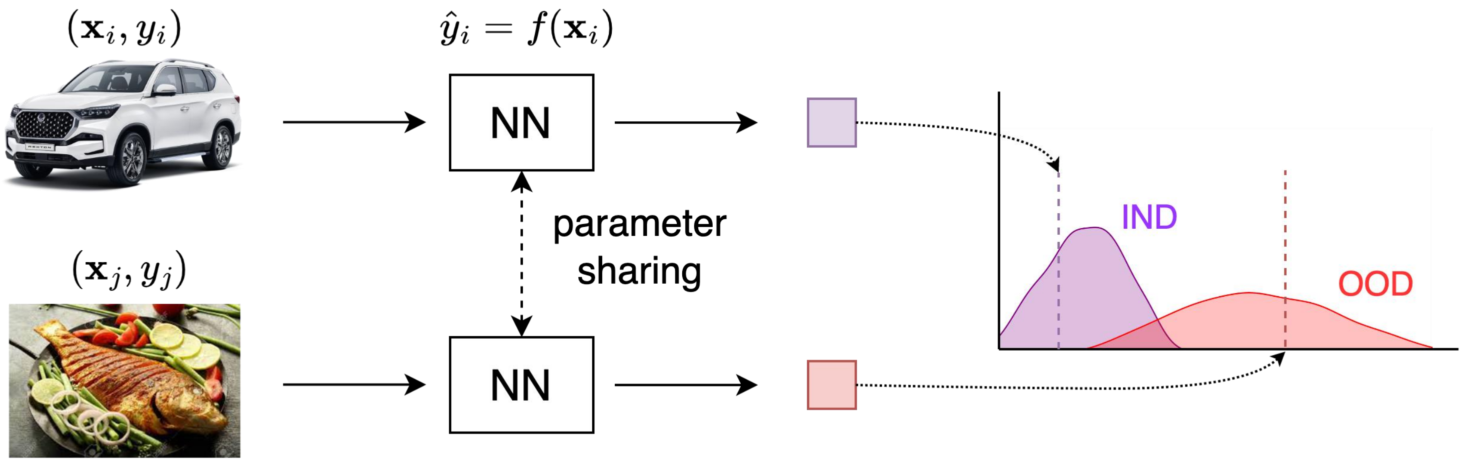
$$(x_i, y_i) \sim p(x, y | \mathcal{N}_i)$$

instances have inter-connection and cannot be treated as i.i.d. samples (non-i.i.d.)

Image Data v.s. Graph Data

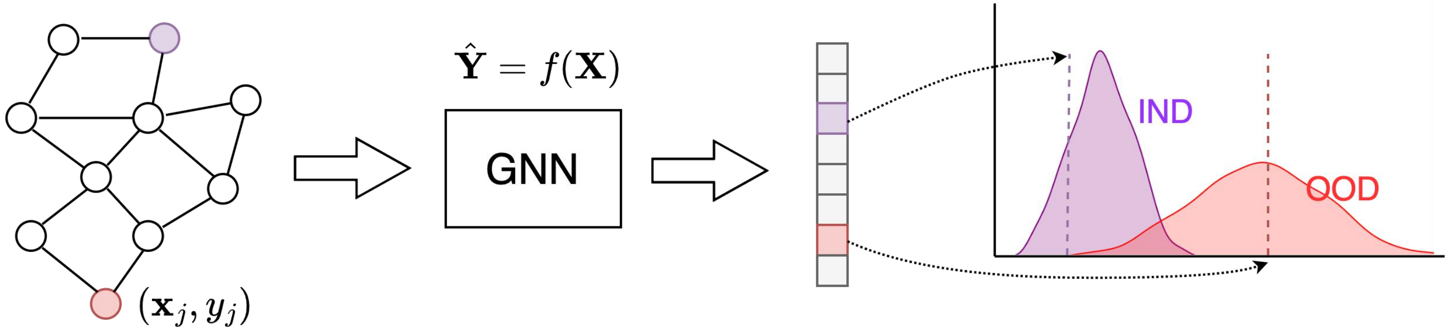
For image data:

instances are i.i.d. sampled without inter-dependence



For graph data:

instances have inter-dependence



Applications: fraud detection in financial networks, risk control in autonomous driving, etc.

OOD Detection on Graph Data

- Assume an input graph $G = (V, E)$, where V, E denotes the node and edge set. Each node has an input feature vector $\mathbf{x}_i$ and label $\mathbf{y}_i$. The node instances are divided into a labeled set $\mathcal{I}_s$ and an unlabeled set $\mathcal{I}_u$, and $\mathcal{I} = \mathcal{I}_s \cup \mathcal{I}_u$.
- Define $X = \{\mathbf{x}_i\}_{i\in\mathcal{I}}$ and $Y = \{\mathbf{y}_i\}_{i\in\mathcal{I}}$. Our goal is to learn a node-level classifier f that can predict node labels $\hat{Y} = \{\hat{y}_i\}_{i\in\mathcal{I}}$, denoted as $\hat{Y} = f(\mathbf{X}, A)$, and in the meanwhile the classifier f can induce a decision function $G(\mathbf{x}, \mathcal{G}_{\mathbf{x}}; f)$ for identifying OOD samples

$$G(\mathbf{x}, \boxed{\mathcal{G}_{\mathbf{x}}}; \boxed{f}) = \begin{cases} 1, & \mathbf{x} \ \text{ is an in-distribution instance,} \\ 0, & \mathbf{x} \ \text{ is an out-of-distribution instance,} \end{cases}$$

where $\mathcal{G}_{\mathbf{x}}$ denotes the ego-graph centered at node instance $\mathbf{x}$.

GNN-based Node-Level Prediction

- Adopt graph neural networks (GNNs) to compute node representations:

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

- The GNN classifier gives a predictive distribution for node labels:

$$p(y \mid \mathbf{x}, \mathcal{G}_\mathbf{x}) = \frac{e^{h_\theta(\mathbf{x},\mathcal{G}_\mathbf{x})_{[y]}}}{\sum_{c=1}^{C} e^{h_\theta(\mathbf{x},\mathcal{G}_\mathbf{x})_{[c]}}} \qquad \text{where} \quad \mathbf{z}_i^{(L)} = h_\theta(\mathbf{x}, \mathcal{G}_\mathbf{x})$$

- If we assume $E(\mathbf{x}, \mathcal{G}_\mathbf{x}, y; h_\theta) = -h_\theta(\mathbf{x}, \mathcal{G}_\mathbf{x})_{[y]}$ as an energy function, we have

$$p(y|\mathbf{x}, \mathcal{G}_\mathbf{x}) = \frac{e^{-E(\mathbf{x},\mathcal{G}_\mathbf{x},y)}}{\sum_{y'} e^{-E(\mathbf{x},\mathcal{G}_\mathbf{x},y')}} = \frac{e^{-E(\mathbf{x},\mathcal{G}_\mathbf{x},y)}}{e^{-E(\mathbf{x},\mathcal{G}_\mathbf{x})}}$$

a Boltzmann distribution

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

free energy for OOD detection

Energy Models for OOD Detection

- For a given GNN classifier $h_\theta(\mathbf{x}, \mathcal{G}_\mathbf{x})$, we have the initial energy as

$$\mathbf{E}^{(0)} = [E(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i}; h_\theta)]_{i \in \mathcal{I}} \qquad \text{where } E(\mathbf{x}, \mathcal{G}_\mathbf{x}; h_\theta) = -\log \sum_{c=1}^{C} e^{h_\theta(\mathbf{x}, \mathcal{G}_\mathbf{x})_{[c]}}$$

- Then we consider propagating the energy values along graph structures

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

Intuition: connected nodes in the graph tend to be sampled from similar distributions

Proposition (informal)

The energy propagation facilitates *consensus* for the OOD estimation results between the target node and its neighboring nodes.

Loss Functions for Training

- If the training data only contains in-distribution data, use supervised loss:

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

GNN-Safe

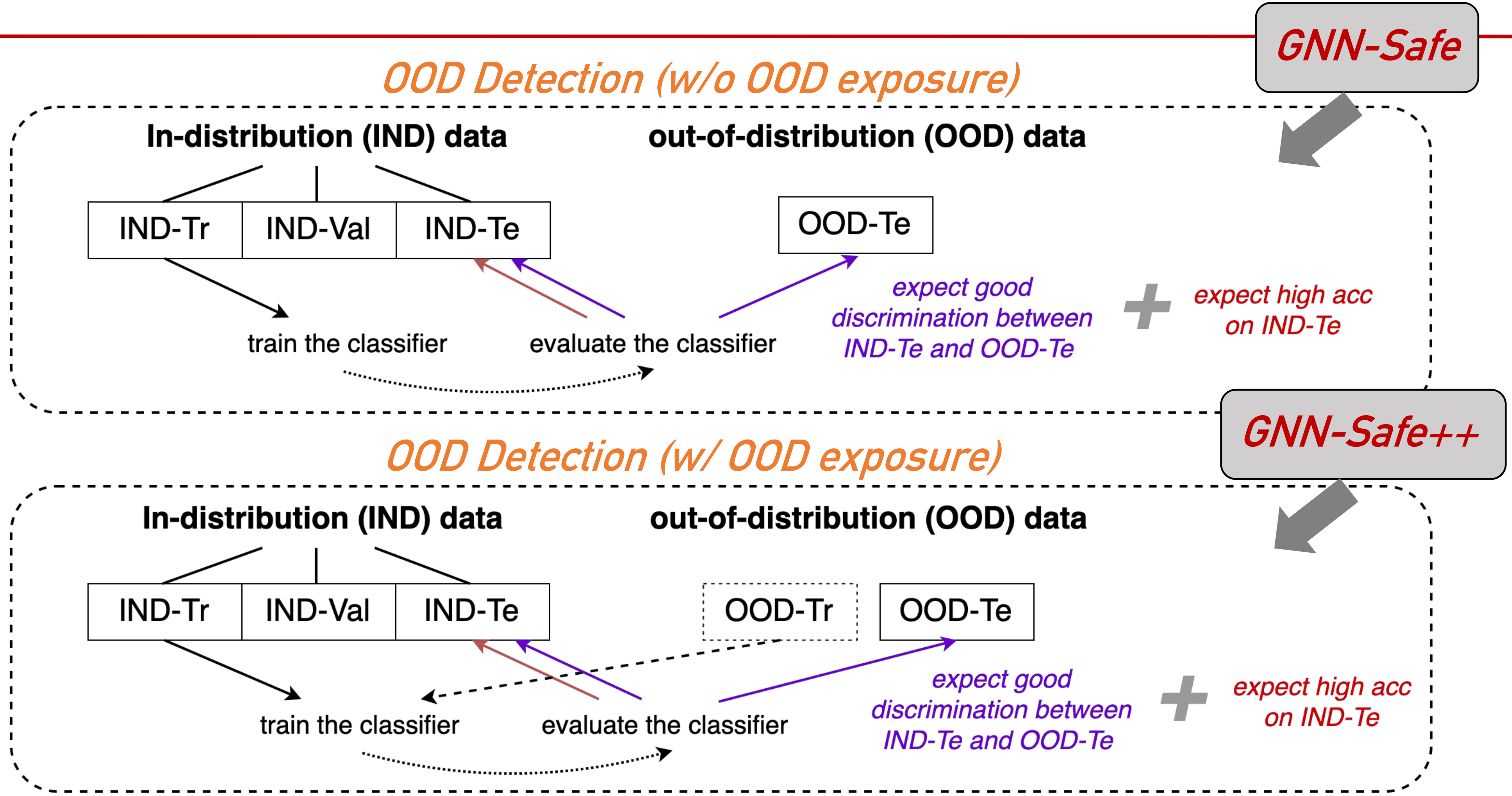
- If the training data contains extra OOD data, we additionally consider the regularization loss: $\mathcal{L}_{sup} + \lambda \mathcal{L}_{reg}$

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

GNN-Safe++

extra OOD training data

Evaluation Protocols



Dataset and Splits

How to introduce distribution shifts for model evaluation?

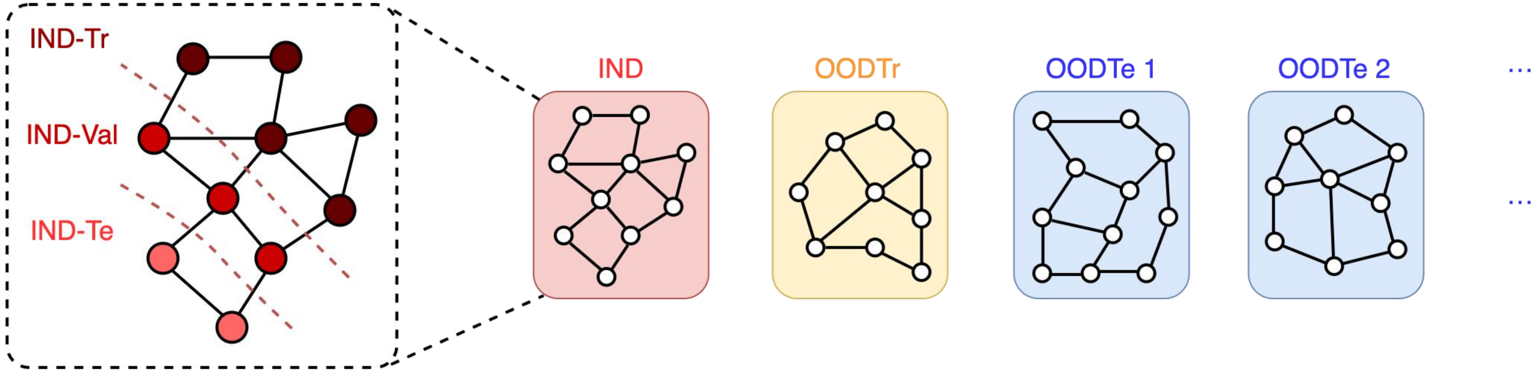
Principles for data splits

$$P_{OOD} \neq P_{IND}$$

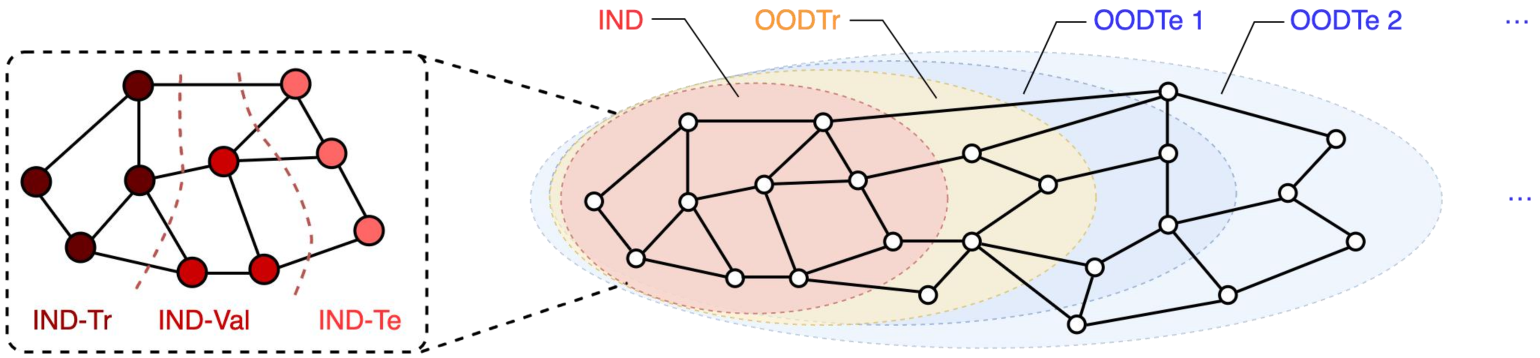
$$P_{OOD-Tr} \neq P_{OOD-Te}$$

$$P_{IND-Tr} = P_{IND-Te}$$

For multi-graph datasets:



For single-graph datasets:



Main Results on Real-World Datasets

OOD detection results on Twitch and Arxiv

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

- Metric: AUROC, AUPR, FPR for detection scores of IND-Te and OOD-Te samples
- Twitch (multi-graph dataset): use nodes in different graphs for IND/OOD
- Arxiv (a temporal graph dataset): use nodes at different times for IND/OOD

Main Results on Synthetic Datasets

OOD detection results on Cora, Amazon-Photo and Coauthor-CS

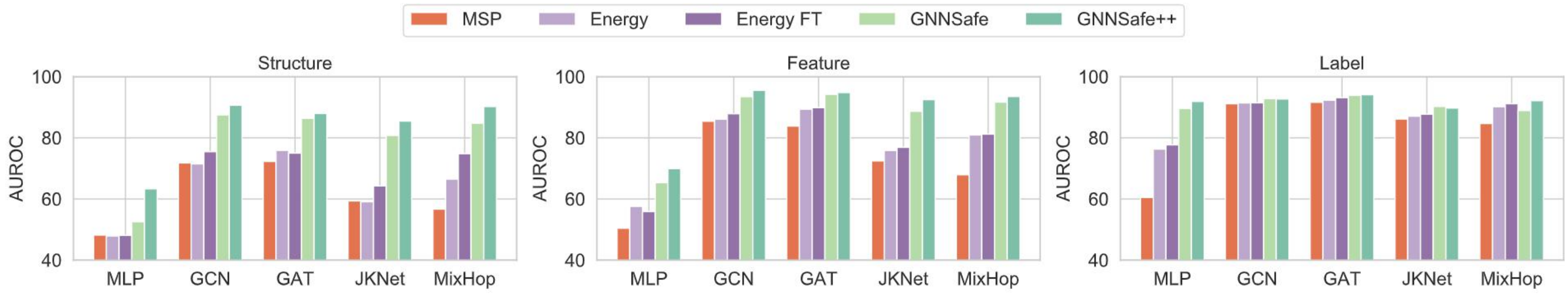
Model	OOD Expo	Cora S	Cora F	Cora L	Amazon S	Amazon F	Amazon L	Coauthor S	Coauthor F	Coauthor L
MSP	No	70.90	85.39	91.36	98.27	97.31	**93.97**	95.30	97.05	94.88
ODIN	No	49.92	49.88	49.80	93.24	81.15	65.97	52.14	51.54	51.44
Mahalanobis	No	46.68	49.93	67.62	71.69	76.50	73.25	80.46	93.23	85.36
Energy	No	71.73	**86.15**	**91.40**	**98.51**	**97.87**	93.81	**96.18**	**97.88**	**95.87**
GKDE	No	68.61	82.79	57.23	76.39	58.96	65.58	65.87	80.69	61.15
GPN	No	**77.47**	85.88	90.34	97.17	87.91	92.72	34.67	72.56	83.65
GNNSAFE	No	**87.52**	**93.44**	**92.80**	**99.58**	**98.55**	**97.35**	**99.60**	**99.64**	**97.23**
OE	Yes	67.98	81.83	89.47	**99.60**	98.39	95.39	97.86	99.04	96.04
Energy FT	Yes	**75.88**	**88.15**	**91.36**	98.83	**98.55**	**97.35**	**98.84**	**99.43**	**96.23**
GNNSAFE++	Yes	**90.62**	**95.56**	**92.75**	**99.82**	**99.64**	**97.51**	**99.99**	**99.97**	**97.89**

synthetic OOD data

- S: randomly generate edges with stochastic block model
- F: modify node features via the mix of arbitrary node pairs
- L: use label classes to divide IND/OOD

Comparison of GNN Backbones

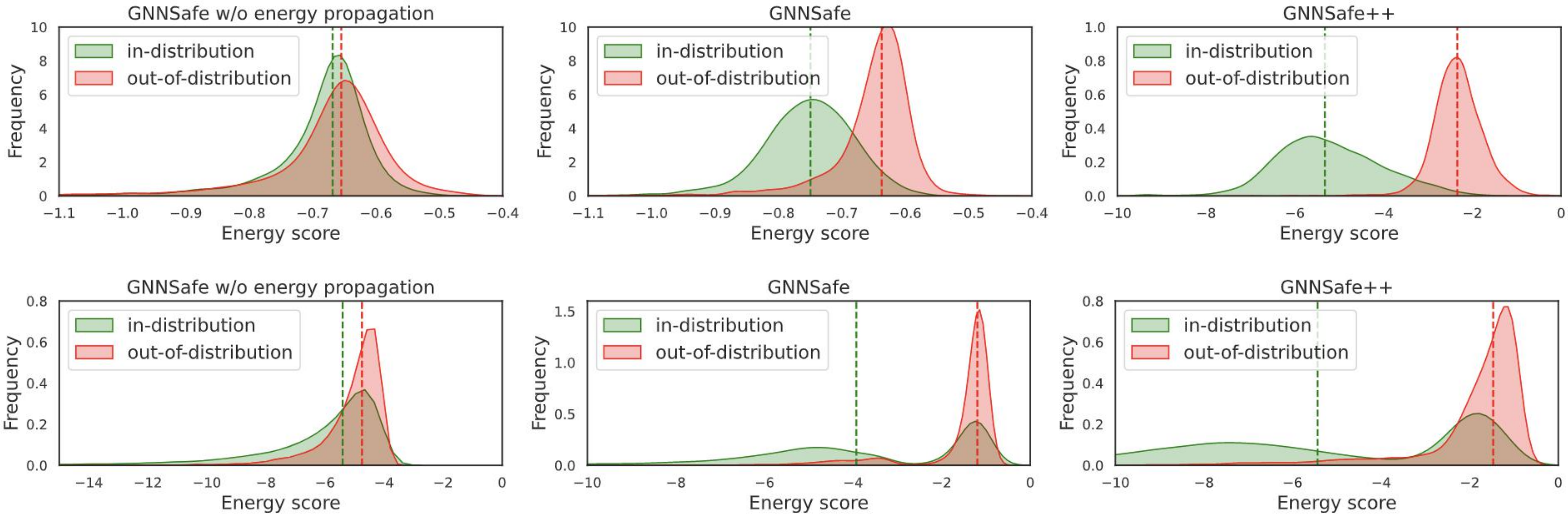
OOD detection results with different GNN classifier backbones



Performance comparison:

- Energy < GNNSafe, Energy FT < GNNSafe++ ➡ *energy propagation is effective*
- GNNSafe < Energy FT ➡ *energy propagation contributes to more performance gain than energy regularization*

Energy Score Visualization



Energy propagation and regularization can both help to enlarge the discrimination gap

Resources and Related Materials

code

blog

https://github.com/qitianwu/GraphOOD-GNNSafe

https://zhuanlan.zhihu.com/p/609178151

知乎专栏：分布外泛化与检测

Out-of-Distribution Detection:

[1] Energy-based Out-of-Distribution Detection for Graph Neural Networks, in ICLR'23

[2] GraphDE: A Generative Framework for Debiased Learning and Out-of-Distribution Detection on Graphs, in NeurIPS'22

Out-of-Distribution Generalization:

[3] Handling Distribution Shifts on Graphs: An Invariance Perspective, in ICLR'22

[4] Learning Substructure Invariance for Out-of-Distribution Molecular Representations, in NeurIPS'22

[5] Towards out-of-distribution sequential event prediction: A causal treatment, in NeurIPS'22

Email: echo740@sjtu.edu.cn, sjtucyt@sjtu.edu.cn