A **Data-centric Framework** to Endow Graph Neural Networks with **Out-Of-Distribution Detection Ability**

Presenter: Zhanke Zhou

2023. 07. 05

About the paper

A Data-centric Framework to Endow Graph Neural Networks with Out-Of-Distribution Detection Ability

Yuxin Guo
Cheng Yang*
Beijing University of Posts and Telecommunications
Beijing, China

Yuluo Chen
Jixi Liu
Beijing University of Posts and Telecommunications
Beijing, China

Chuan Shi
Junping Du
Beijing University of Posts and Telecommunications
Beijing, China

- Conference: KDD 2023
- Paper: http://shichuan.org/doc/150.pdf
- Code: https://github.com/BUPT-GAMMA/AAGOD

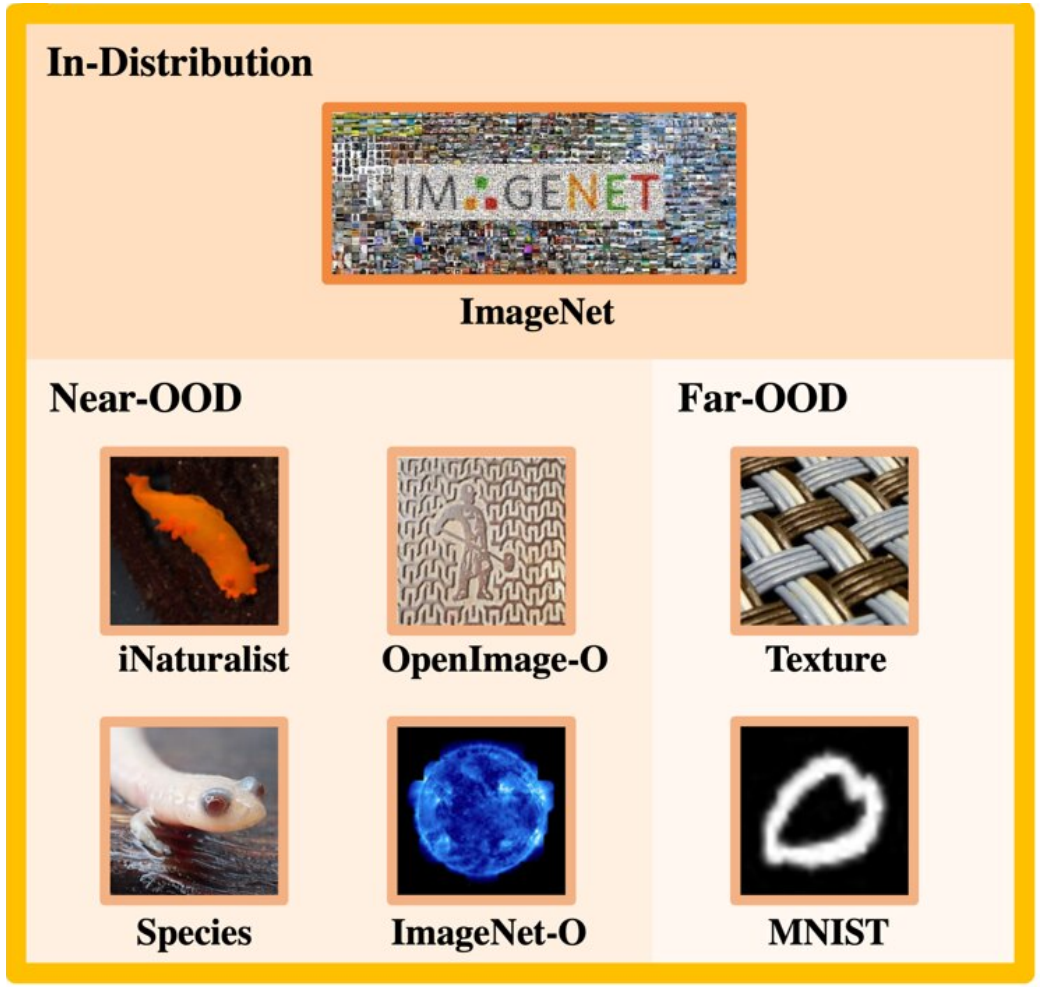
Outline

- Background
- Method
- Experiment
- Summary

Background | Visual OOD detection

In-distribution Data
(ID data)

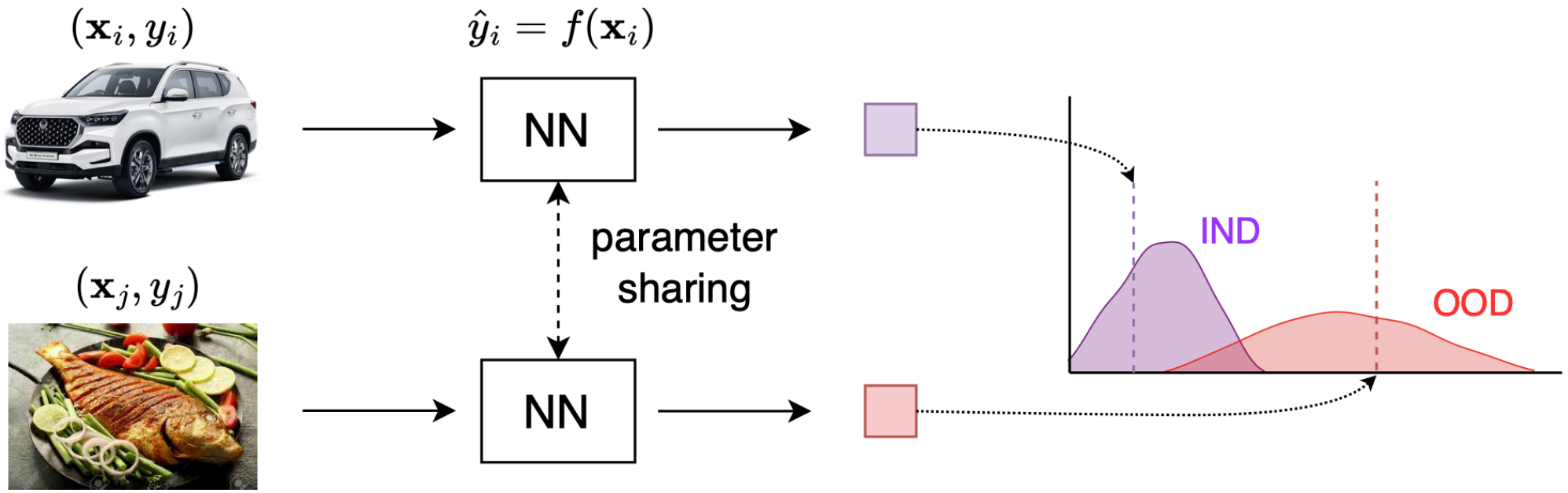
Out-of-distribution Data
(OOD data)



Background | Visual OOD detection

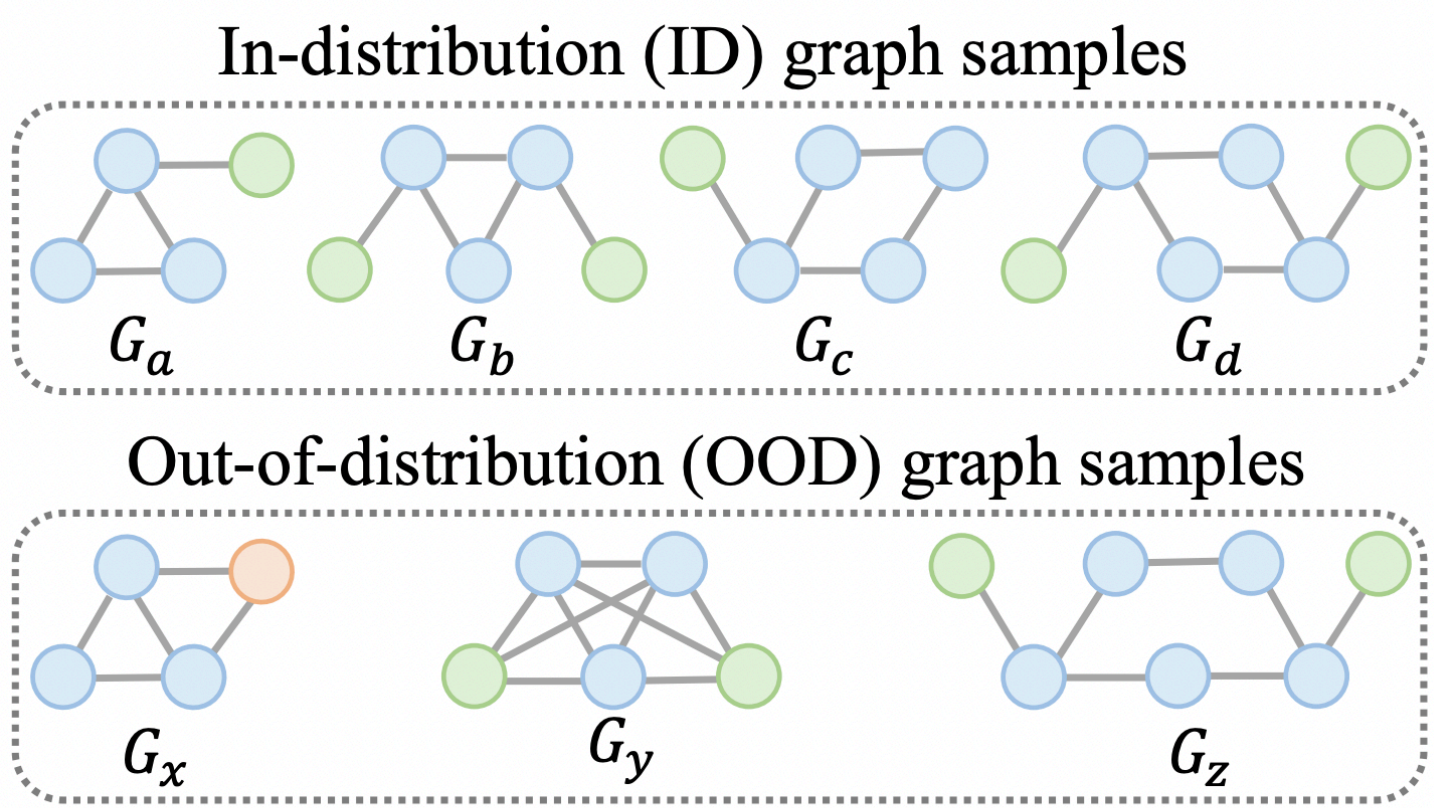
a trustworthy model should

- not only perform well on ID samples
- but also recognize what they don't know



(a) OOD detection for image data where image instances are assumed to be independent

Background | Graph OOD detection



(a) ID and OOD graph samples

Background | Graph OOD detection

Reusing-based Graph OOD Detection. In this paper, we make the first attempt to the reusing-based graph OOD detection, which performs detection by utilizing a well-trained GNN encoder in a post-hoc manner. Formally, with 0 indicating the OOD case and 1 the ID case, the reusing-based graph OOD detection aims to build a detection model detect$(\cdot)$ for distinguishing input graph G^k:

$$\text{detect}(G^k) = \begin{cases} 1, & g(G^k, f) \geq \gamma \\ 0, & g(G^k, f) < \gamma \end{cases} \tag{1}$$

where γ is a threshold, f is the well-trained GNN encoder with parameters Θ, and function g returns a score to estimate whether G^k is an ID graph or not with the help of f.

A Naive Solution. Since the GNN encoder f is sufficiently trained and can extract expressive graph representations, an intuitive idea for constructing the function g is to keep parameters Θ in encoder f untouched and directly apply a pre-defined non-parametric scoring function s on encoded representations. Formally, we can write $g(G^k, f)$ as follows:

$$g(G^k, f) = s\big(f(A^k, X^k; \Theta)\big), \tag{2}$$

where $f(A^k, X^k; \Theta)$ is the vector representation $\boldsymbol{h}^k$ of G^k encoded by GNN, and the scoring function $s(\cdot)$ measures the centrality of a sample in the representation space. Note that the function g includes no learnable parameters.

Background | Graph OOD detection

Reusing-based Graph OOD Detection. In this paper, we make the first attempt to the reusing-based graph OOD detection, which performs detection by utilizing a well-trained GNN encoder in a post-hoc manner. Formally, with 0 indicating the OOD case and 1 the ID case, the reusing-based graph OOD detection aims to build a detection model detect$(\cdot)$ for distinguishing input graph G^k:

$$\begin{cases} 1, & g(G^k, f) \geq \gamma \end{cases}$$

A Naive Solution. Since the GNN encoder f is sufficiently trained and can extract expressive graph representations, an intuitive idea for constructing the function g is to keep parameters Θ in encoder f untouched and directly apply a pre-defined non-parametric scoring function s on encoded representations. Formally, we can write $g(G^k, f)$ as follows:

$$g(G^k, f) = s\big(f(A^k, X^k; \Theta)\big), \tag{2}$$

where $f(A^k, X^k; \Theta)$ is the vector representation $\boldsymbol{h}^k$ of G^k encoded

LOF assumes the sample with low local density is more likely to be an outlier. For simplicity, it introduces a distance function $d_r(\boldsymbol{h}^k)$ to measure the distance between $\boldsymbol{h}^k$ and its r-th nearest neighbor, and calculates the local reachability density (lrd) of $\boldsymbol{h}^k$ via $d_r(\boldsymbol{h}^k)$. Then the final score of $\boldsymbol{h}^k$ is defined as:

$$s_{\text{LOF}}(\boldsymbol{h}^k) = \frac{1}{|N_r(\boldsymbol{h}^k)|} \sum_{\boldsymbol{h}^q \in N_r(\boldsymbol{h}^k)} \frac{lrd(\boldsymbol{h}^q)}{lrd(\boldsymbol{h}^k)}, \tag{3}$$

where $N_r(\boldsymbol{h}^k)$ is the collection of neighbors whose distance from $\boldsymbol{h}^k$ is less than $d_r(\boldsymbol{h}^k)$.

SSD leverages K-means clustering to separate the representations into T clusters, and then use the Mahalanobis distance between $\boldsymbol{h}^k$ and the corresponding cluster center to compute the score of $\boldsymbol{h}^k$. Formally, the scoring function of SSD is:

$$s_{\text{SSD}}(\boldsymbol{h}^k) = \frac{1}{\min_t (\boldsymbol{h}^k - \boldsymbol{\mu}_t)^\top \Sigma_t^{-1} (\boldsymbol{h}^k - \boldsymbol{\mu}_t)}, \tag{4}$$

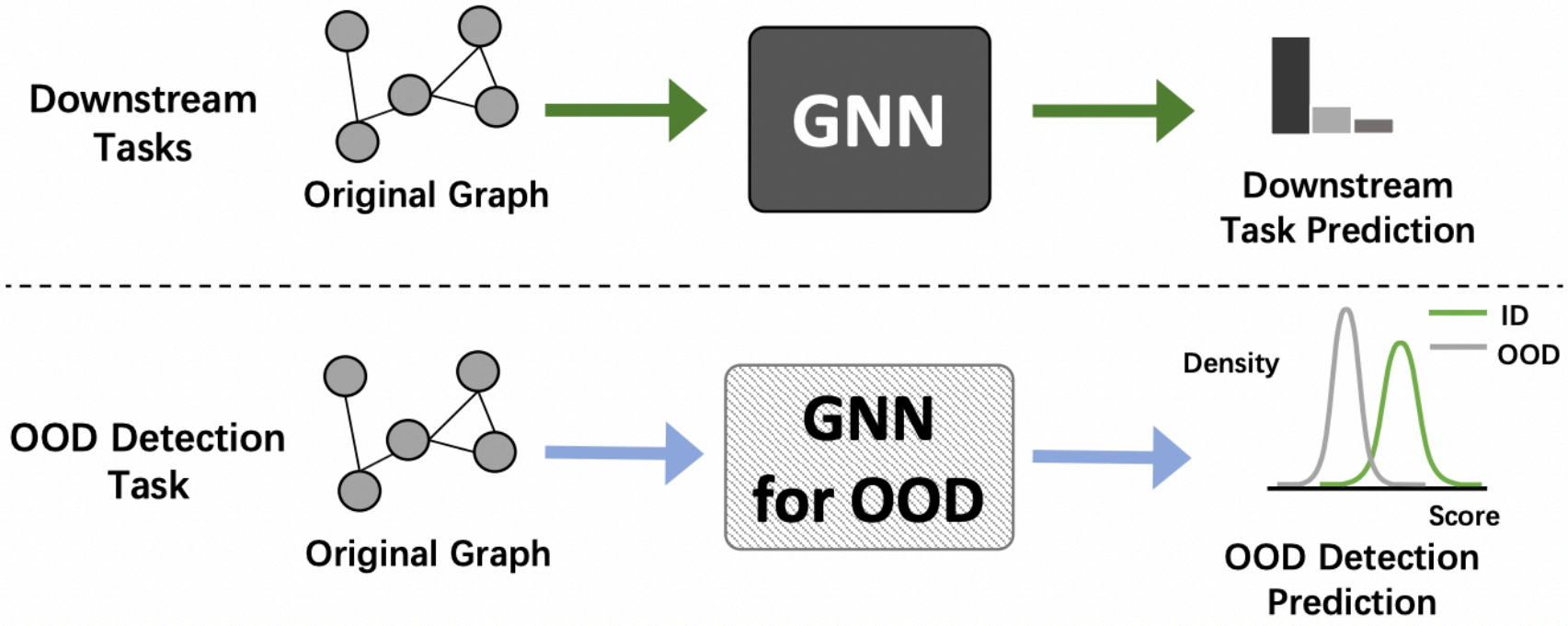
where $\boldsymbol{h}^k$ belongs to the cluster t, $\boldsymbol{\mu}_t$ and Σ_t are the mean and covariance of representations in cluster t. Since all ID graphs are assumed to be drawn from the same distribution $\mathcal{P}_{\text{IN}}$, we simply set $T = 1$ in practice.

SSD/LOF: two scoring functions

Background | Graph OOD detection

Train-from-scratch pipeline

- problem: high computation cost

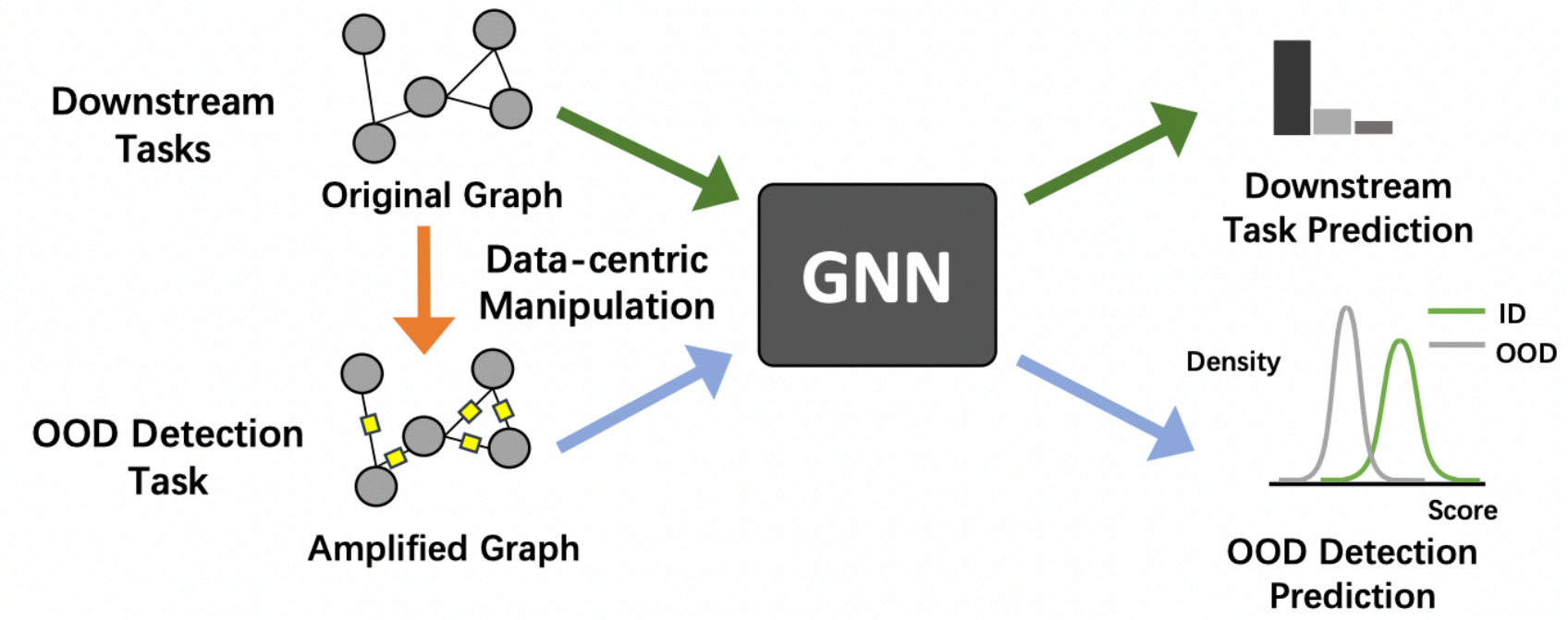


(a) Typical retraining-based graph OOD detection methods [24, 39, 58].

Background | Graph OOD detection

This work: post-hoc pipeline

- endow a well-trained GNN with the OOD detection ability
- without modifying its parameters



(b) Our proposed data-centric framework for graph OOD detection.

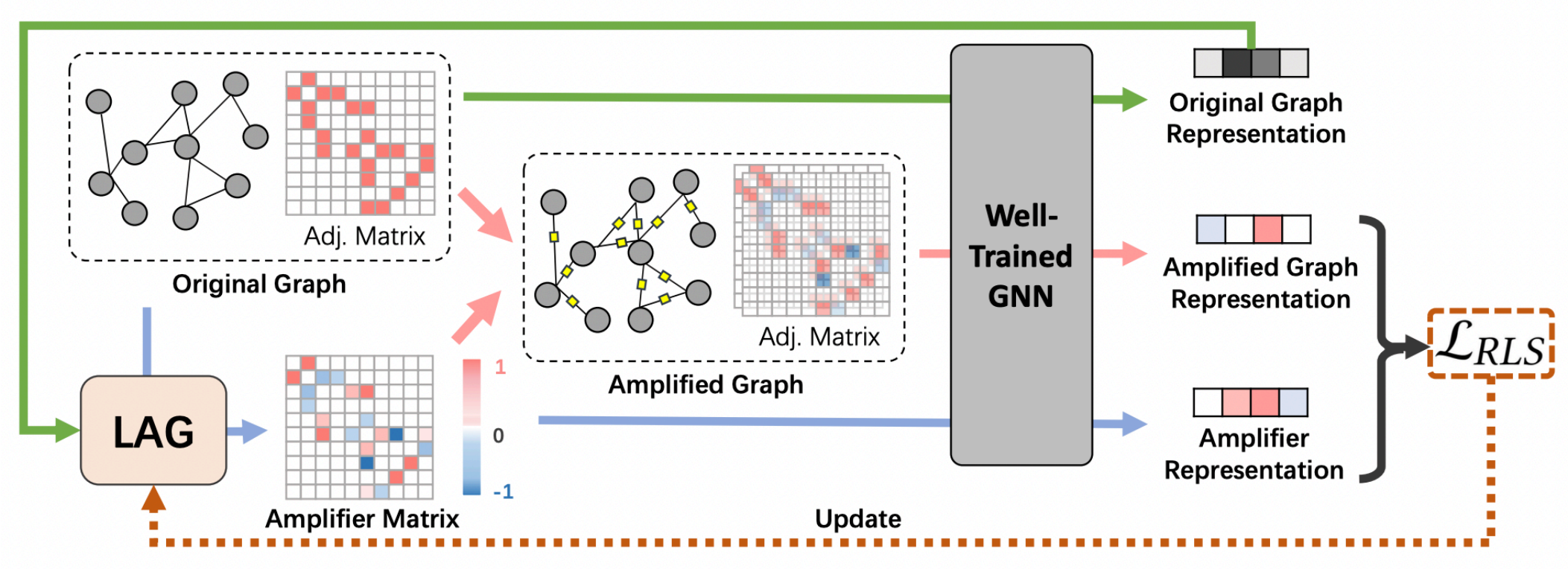
Outline

- Background
- Method
- Experiment
- Summary

Method

AAGOD: a post-hoc framework with **Adaptive Amplifier** for Graph OOD Detection

- concentrating on **data-centric** manipulation
- help **highlight** the latent pattern of ID graphs
- **enlarge the score gap** between OOD and ID graphs



Method

AAGOD: a post-hoc framework with **Adaptive Amplifier** for Graph OOD Detection

- concentrating on **data-centric** manipulation
- help **highlight** the latent pattern of ID graphs
- **enlarge the score gap** between OOD and ID graphs

Difficulties

- **The number and order of nodes in graph data are uncertain**
  - infeasible to design a global static amplifier suitable for all graphs
- **OOD graphs are not available during training**
  - hard to design a learning objective for distinguishing ID and OOD data

Method

AAGOD: a post-hoc framework with **Adaptive Amplifier** for Graph OOD Detection

- concentrating on **data-centric** manipulation
- help **highlight** the latent pattern of ID graphs
- **enlarge the score gap** between OOD and ID graphs

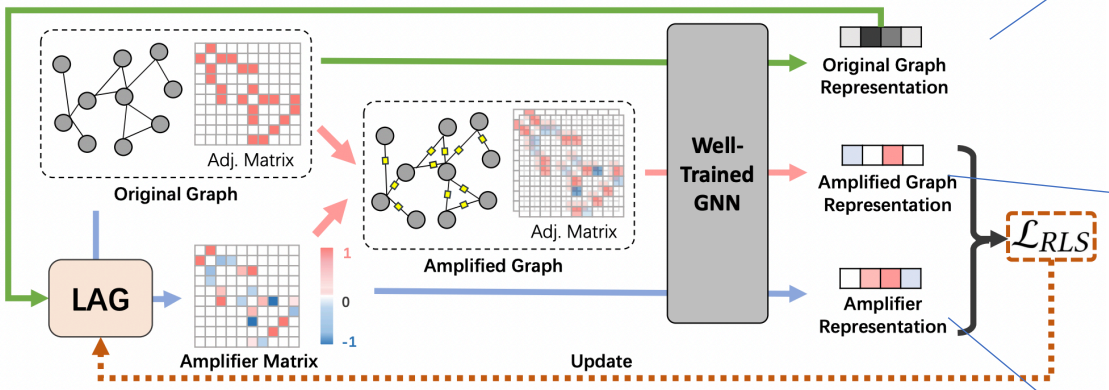
Difficulties

- **The number and order of nodes in graph data are uncertain**
  - infeasible to design a global static amplifier suitable for all graphs
- **OOD graphs are not available during training**
  - hard to design a learning objective for distinguishing ID and OOD data

Solutions

- **a Learnable Amplifier Generator (LAG)**
  - to adaptively generate graph-specific amplifiers
  - utilize the node representations encoded by the well-trained GNN
  - to generate weights for the edges in the original graph
- **a Regularized Learning Strategy (RLS)**
  - encourages high scores for amplified ID graphs
  - expects low scores when only seeing the amplifiers

Method



$$\mathbf{M}_{\Omega}^{k}$$

$$m_{ij;\Omega}^{k} = \mathrm{MLP}\left(\mathrm{Concat}(\boldsymbol{p}_i^k, \boldsymbol{p}_j^k); \Omega\right), \tag{8}$$

$$\begin{aligned} \boldsymbol{p}_i^k &= \mathrm{MP}(v_i|\mathbf{A}^k, \mathbf{X}^k; \Theta), \\ \boldsymbol{h}^k &= \mathrm{Pool}(\{\boldsymbol{p}_i^k | v_i \in \mathcal{V}^k\}), \end{aligned} \tag{7}$$

original graph representation

$$\boldsymbol{h}_{\Omega}^{k} = \mathrm{Pool}\left(\{\mathrm{MP}(v_i|\mathbf{A}^k + \mathbf{M}_{\Omega}^k, \mathbf{X}^k; \Theta) | v_i \in \mathcal{V}^k\}\right). \tag{9}$$

"ID graph" representation

$$\boldsymbol{h}_{\mathrm{OOD};\Omega}^{k} = \mathrm{Pool}\left(\{\mathrm{MP}(v_i|\mathbf{M}_{\Omega}^k, \mathbf{X}^k; \Theta) | v_i \in \mathcal{V}^k\}\right), \tag{11}$$

"OOD graph" representation

Method

AAGOD: a post-hoc framework with **Adaptive Amplifier** for Graph OOD Detection

- concentrating on **data-centric** manipulation
- help **highlight** the latent pattern of ID graphs
- **enlarge the score gap** between OOD and ID graphs

Difficulties

- **The number and order of nodes in graph data are uncertain**
  - infeasible to design a global static amplifier suitable for all graphs
- **OOD graphs are not available during training**
  - hard to design a learning objective for distinguishing ID and OOD data

Solutions

- **a Learnable Amplifier Generator (LAG)**
  - to adaptively generate graph-specific amplifiers
  - utilize the node representations encoded by the well-trained GNN
  - to generate weights for the edges in the original graph
- **a Regularized Learning Strategy (RLS)**
  - encourages high scores for amplified ID graphs
  - expects low scores when only seeing the amplifiers

Method

$$\boldsymbol{h}_{\Omega}^{k} = \text{Pool}\left(\{\text{MP}(v_i|\mathbf{A}^k + \mathbf{M}_{\Omega}^k, \mathbf{X}^k; \Theta)|v_i \in \mathcal{V}^k\}\right). \tag{9}$$

$$\mathcal{L}_{\text{ID}} = \sum_{k=1}^{n} 1/s(\boldsymbol{h}_{\text{ID};\Omega}^{k}), \tag{10}$$

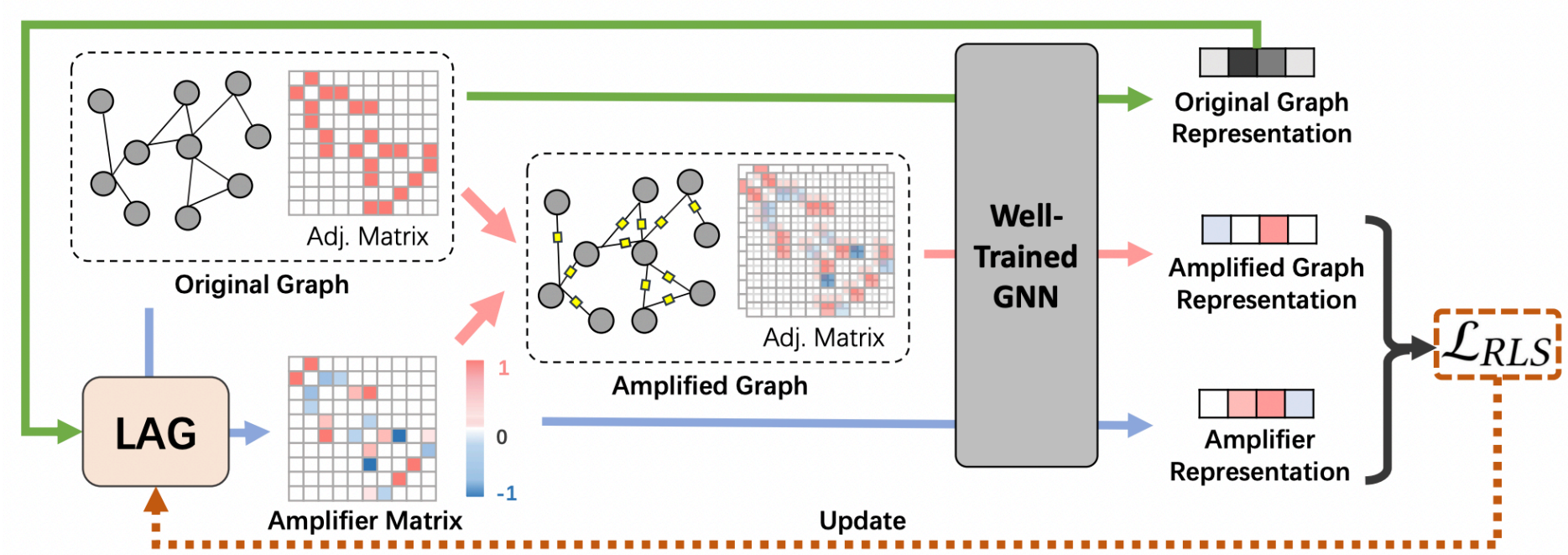
$$\boldsymbol{h}_{\text{OOD};\Omega}^{k} = \text{Pool}\left(\{\text{MP}(v_i|\mathbf{M}_{\Omega}^k, \mathbf{X}^k; \Theta)|v_i \in \mathcal{V}^k\}\right), \tag{11}$$

$$\mathcal{L}_{\text{OOD}} = \sum_{k=1}^{n} s(\boldsymbol{h}_{\text{OOD};\Omega}^{k}), \tag{12}$$

Finally, the overall learning objective can be written as follows:

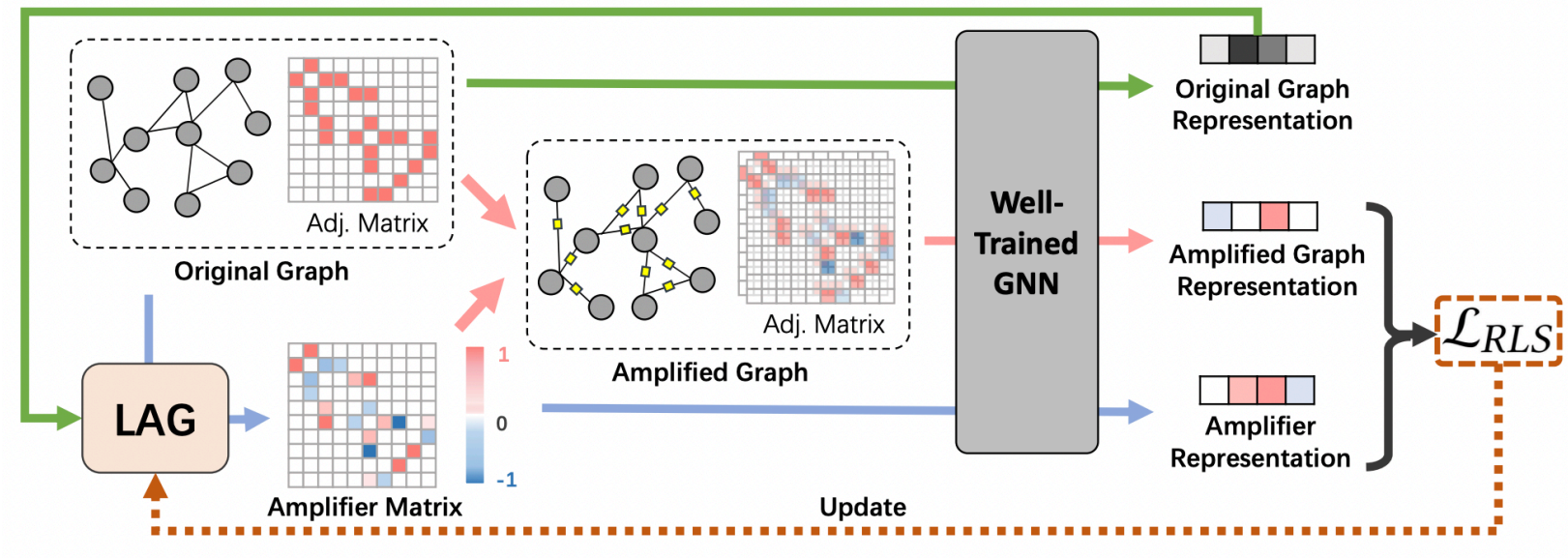
$$\mathcal{L}_{\text{RLS}} = \mathcal{L}_{\text{ID}} + \lambda\mathcal{L}_{\text{OOD}} + \tau\sum_{k=1}^{n}||\mathbf{M}_{\Omega}^k||, \tag{13}$$

where λ and τ are the balance hyper-parameters, $||\cdot||$ is the L1-norm.



Method

Full algorithm



Algorithm 1 Adaptive Amplifier for Graph OOD Detection

Require: Training set of ID graphs $\mathcal{D}_{\text{ID}}$, well-trained GNN model f with parameters $\boldsymbol{\Theta}$;

Ensure: LAG with learned parameters $\boldsymbol{\Omega}$;

```
Randomly initialize Ω;
while not converge do
    for each graph G^k ∈ D_ID do
        Compute p_i^k for each node v_i ∈ V^k by Eq. (7);
        Compute the amplifier matrix M_Ω^k by Eq. (8);
        Compute h^k_{ID;Ω} and h^k_{OOD;Ω} by Eq. (9) and Eq. (11) with
        M_Ω^k, respectively;
    end for
    Update Ω by minimizing Eq. (13);
end while
return Learned parameters Ω for LAG.
```

Outline

- Background
- Method
- Experiment
- Summary

Experiment

- 6.21% relative enhancement in AUC
- a 34 times faster training speed

Table 1: Graph OOD detection performance with unsupervised GNNs as GCL [54] and JOAO [53]. Here the subscript S/L indicates the SSD/LOF scoring function in Section 3.1. $GNN_{S/L}$ is the non-parametric naive solution, while $GNN_{S/L}$+ is our AAGOD.

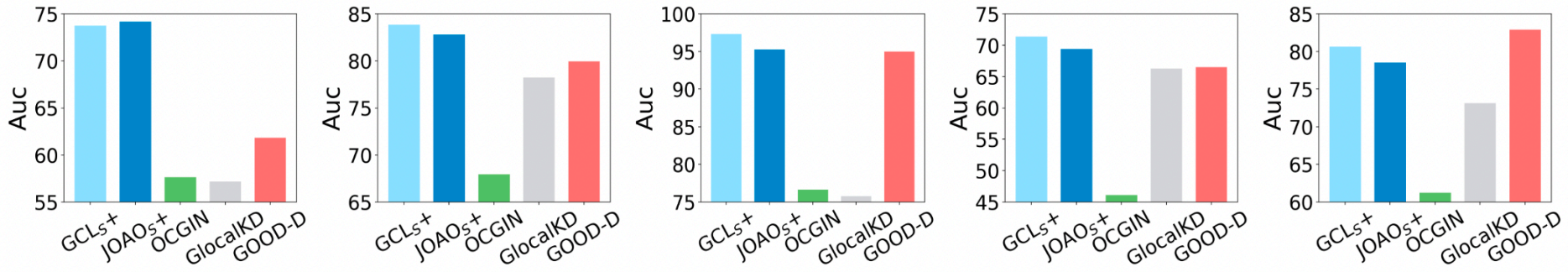
ID	OOD	Metric	GCL_S	GCL_S+	Improv.	GCL_L	GCL_L+	Improv.	$JOAO_S$	$JOAO_S$+	Improv.	$JOAO_L$	$JOAO_L$+	Improv.
ENZYMES	PROTEIN	AUC ↑	62.97	**73.76**	+17.14%	62.56	**67.15**	+7.34%	61.20	**74.19**	+21.23%	59.68	**65.11**	+9.10%
		AUPR ↑	62.47	**75.27**	+20.49%	**65.45**	65.18	-0.41%	61.30	**77.10**	+25.77%	64.16	**64.49**	+0.51%
		FPR95 ↓	93.33	**88.33**	-5.36%	93.30	**85.00**	-8.90%	90.00	**81.67**	-9.26%	96.67	**85.00**	-12.07%
IMDBM	IMDBB	AUC ↑	80.52	**83.84**	+4.12%	61.08	**68.64**	+12.38%	80.40	**82.80**	+2.99%	48.25	**64.32**	+33.31%
		AUPR ↑	74.43	**80.16**	+7.70%	59.52	**68.03**	+14.30%	74.70	**77.77**	+4.11%	47.88	**61.62**	+28.70%
		FPR95 ↓	38.67	**38.33**	-0.88%	96.67	**91.33**	-5.52%	44.70	**42.00**	-6.04%	98.00	**94.00**	-4.08%
BZR	COX2	AUC ↑	75.00	**97.31**	+29.75%	34.69	**65.00**	+87.37%	80.00	**95.25**	+19.06%	41.80	**65.62**	+56.99%
		AUPR ↑	62.41	**97.17**	+55.70%	39.07	**62.89**	+60.97%	67.10	**94.34**	+40.60%	56.70	**67.22**	+18.55%
		FPR95 ↓	47.50	**15.00**	-68.42%	92.50	**80.00**	-13.51%	37.50	**12.50**	-66.67%	**97.50**	**97.50**	0.00%
TOX21	SIDER	AUC ↑	68.04	**71.27**	+4.75%	53.44	**58.25**	+9.00%	53.46	**69.39**	+29.80%	53.64	**55.67**	+3.78%
		AUPR ↑	69.28	**73.52**	+6.12%	56.81	**59.58**	+4.88%	56.02	**71.01**	+26.76%	**56.02**	**56.02**	0.00%
		FPR95 ↓	90.42	**89.53**	-0.98%	94.25	**92.72**	-1.62%	95.66	**90.55**	-5.34%	95.66	**89.66**	-6.27%
BBBP	BACE	AUC ↑	77.07	**80.64**	+4.63%	46.74	**50.53**	+8.11%	75.48	**78.54**	+4.05%	43.96	**51.28**	+16.65%
		AUPR ↑	68.41	**72.60**	+6.12%	45.35	**46.49**	+2.51%	69.32	**74.06**	+6.84%	44.77	**48.32**	+7.93%
		FPR95 ↓	71.92	**60.59**	-15.75%	92.12	**86.70**	-5.88%	76.85	**69.46**	-9.62%	94.09	**92.61**	-1.57%

Experiment

Table 2: Graph OOD detection performance with supervised GNNs as GIN [50] and PPGN [29]. Here the subscript S/L indicates the SSD/LOF scoring function in Section 3.1. $GNN_{S/L}$ is the non-parametric naive solution, while $GNN_{S/L}$+ is our AAGOD.

ID	OOD	Metric	GIN_S	GIN_S+	Improv.	GIN_L	GIN_L+	Improv.	$PPGN_S$	$PPGN_S$+	Improv.	$PPGN_L$	$PPGN_L$+	Improv.
ENZYMES	PROTEIN	AUC ↑	52.22	**66.22**	+26.81%	58.44	**65.89**	+12.75%	53.89	**66.67**	+23.71%	52.56	**63.22**	+20.28%
		AUPR ↑	50.41	**58.81**	+16.66%	53.82	**59.03**	+9.68%	54.06	**65.70**	+21.53%	51.21	**57.56**	+12.40%
		FPR95 ↓	93.33	**73.33**	-21.43%	90.00	**83.33**	-7.41%	**80.00**	**80.00**	0.00%	100.00	**83.33**	-16.67%
IMDBM	IMDBB	AUC ↑	42.05	**59.00**	+40.31%	57.24	**62.70**	+9.54%	40.62	**59.25**	+45.86%	47.90	**55.64**	+16.16%
		AUPR ↑	44.43	**57.82**	+30.14%	54.41	**62.21**	+14.34%	43.41	**55.04**	+26.79%	50.06	**52.76**	+5.39%
		FPR95 ↓	100.00	**90.67**	-9.33%	**87.17**	95.00	+8.98%	96.43	**85.17**	-11.68%	89.67	**89.33**	-0.38%
BZR	COX2	AUC ↑	35.25	**76.75**	+117.73%	60.75	**76.00**	+25.10%	62.75	**71.75**	+14.34%	65.00	**72.25**	+11.15%
		AUPR ↑	39.61	**66.32**	+67.43%	53.71	**63.18**	+17.63%	57.15	**78.94**	+38.13%	62.14	**77.59**	+24.86%
		FPR95 ↓	100.00	**70.00**	-30.00%	95.00	**45.00**	-52.63%	**65.00**	90.00	+38.46%	**80.00**	95.00	+18.75%
TOX21	SIDER	AUC ↑	63.73	**64.26**	+0.83%	51.47	**57.59**	+11.89%	36.98	**61.24**	+65.60%	54.61	**55.00**	+0.71%
		AUPR ↑	63.79	**67.35**	+5.58%	52.33	**56.55**	+8.06%	43.55	**58.16**	+33.55%	53.91	**57.91**	+7.42%
		FPR95 ↓	**83.78**	93.87	+12.04%	96.93	**92.34**	-4.74%	97.45	**82.63**	-15.21%	**94.38**	97.57	+3.38%
BBBP	BACE	AUC ↑	64.58	**67.80**	+4.99%	43.54	**57.13**	+31.21%	30.79	**70.20**	+128.00%	47.55	**56.96**	+19.79%
		AUPR ↑	58.39	**61.91**	+6.03%	43.80	**54.12**	+23.56%	44.06	**74.56**	+69.22%	49.71	**57.96**	+16.60%
		FPR95 ↓	**87.68**	90.64	+3.38%	**91.63**	92.12	+0.53%	97.56	**78.05**	-20.00%	100.00	**88.78**	-11.22%

Experiment



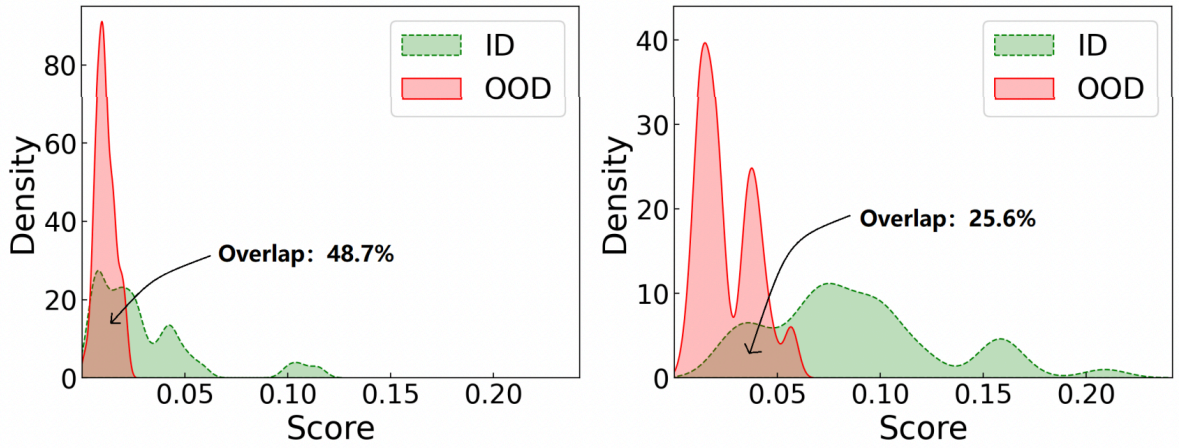
(a) ENZYMES-PROTEIN (b) IMDBM-IMDBB (c) BZR-COX2 (d) TOX21-SIDER (e) BBBP-BACE

Figure 3: AUC of different methods on graph OOD detection. Higher AUC indicates better performance.

Table 3: Training time comparison between AAGOD and the corresponding well-trained GNNs. "Ratio" means the percentage of the running time of AAGOD to that of the original GNNs.

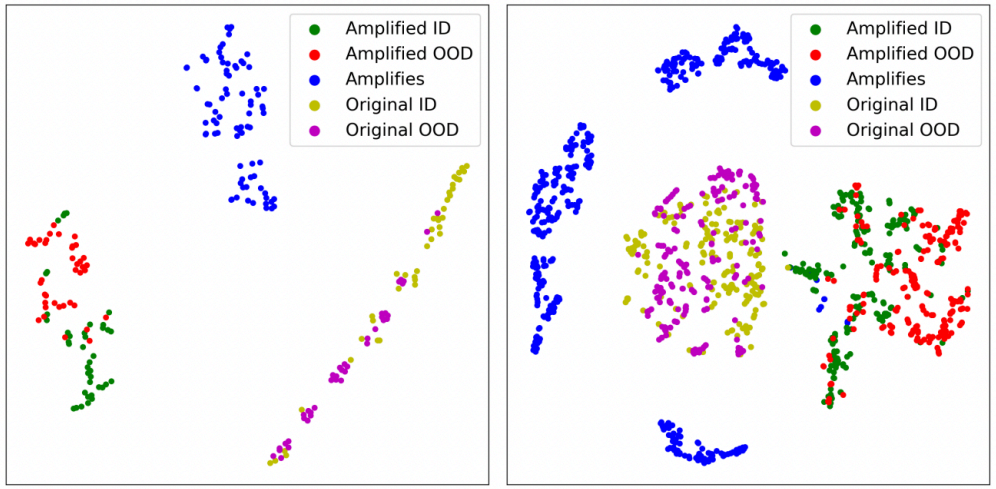
Dataset		Unsupervised						Supervised					
ID	OOD	GCL	GCL_S+	Ratio	JOAO	$JOAO_S+$	Ratio	GIN	GIN_S+	Ratio	PPGN	$PPGN_S+$	Ratio
ENZYMES	PROTEIN	30.65	1.99	6.48%	49.16	2.44	4.97%	47.48	4.99	10.52%	375.31	270.38	72.04%
IMDBM	IMDBB	44.57	6.39	14.33%	110.29	5.07	4.60%	43.45	12.68	29.19%	713.46	519.22	72.78%
BZR	COX2	17.78	1.84	10.34%	39.07	0.96	2.45%	39.87	6.35	15.92%	305.51	197.49	64.64%
TOX21	SIDER	460.76	16.66	3.62%	482.79	15.84	3.28%	509.40	12.76	2.50%	3474.90	2702.61	77.78%
BBBP	BACE	175.10	6.86	3.92%	134.74	5.84	4.34%	137.24	4.24	3.09%	1015.99	668.81	65.83%

Experiment



(a) Before amplifying (b) After amplifying

Figure 5: Scoring distributions before/after amplifying on BZR-COX2. Graphs with high (low) scores are regarded as ID (OOD) graphs, and AAGOD enlarges the distribution gap between ID and OOD graphs.



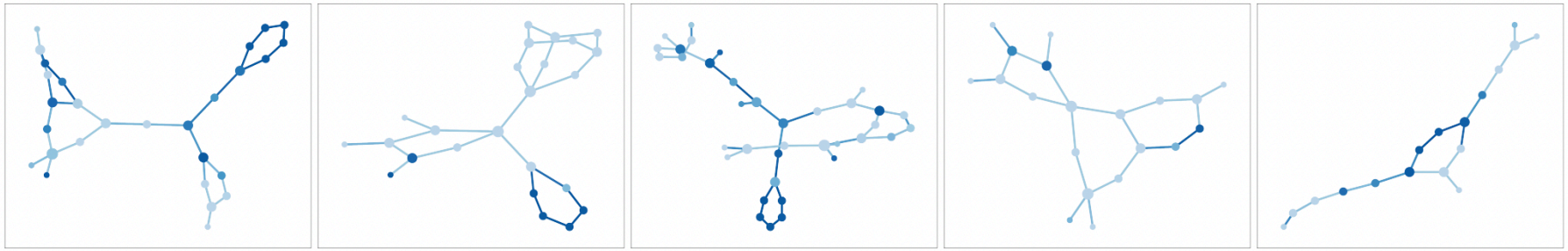
(a) BZR-COX2 (b) BBBP-BACE

Figure 6: Visualization of graph representations on BZR-COX2 and BBBP-BACE. The representations of ID and OOD graphs (yellow v.s. purple) becomes more separable after amplification (green v.s. red).

Experiment

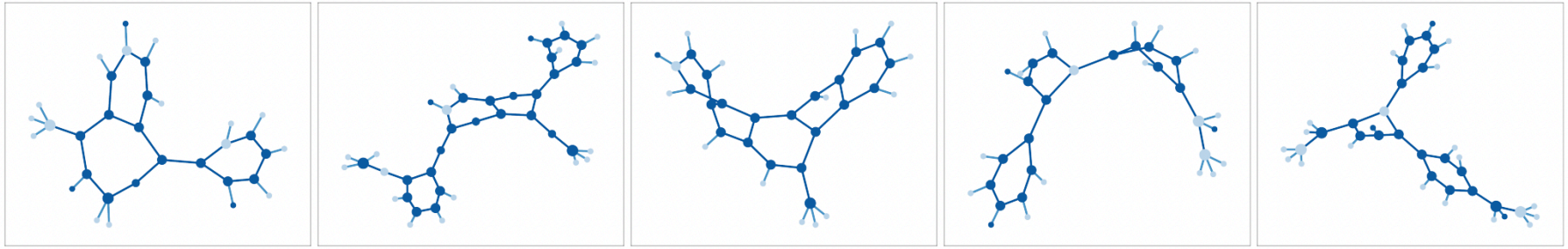
The amplifiers emphasize **the functional group**

- with **a hexagonal structure (e.g., benzene ring)** in ID graphs
- which is the **key structure** making ID graphs different from OOD ones



(a) ID (b) ID (c) ID (d) OOD (e) OOD

Figure 7: Visualization of different amplified graphs on BBBP-BACE. Here dark (light) edges/nodes have large (small) edge/self-loop weights, and the size of a node depends on its degree. The amplifiers emphasize the functional group with a hexagonal structure (*e.g.,* benzene ring) at the end of the backbone in ID graphs, which is the key structure making ID graphs different from OOD ones.



(a) ID (b) ID (c) ID (d) OOD (e) OOD

Figure 8: Visualization of different amplified graphs on BZR-COX2. The amplifiers highlight the backbone and ignore the side chains, since ID graphs usually have adjacent rings sharing common edges in the backbone, which is the critical characteristic making BZR dataset different from others.

Outline

- Background
- Method
- Experiment
- Summary

Summary

Main contributions:

- the first work proposed for reusing-based graph OOD detection problem
  - endow a well-trained GNN with the OOD detection ability

- an effective framework named AAGOD
  - Learnable Amplifier Generator (LAG) and Regularize Learning Strategy (RLS)
  - produce graph-specific amplifiers
  - enlarge the gap between amplified ID and OOD graphs

- Extensive experiments on real-world datasets
  - successfully applied on diverse GNNs
  - outperforms SOTA baselines in graph OOD detection

Related works | Graph OOD Detection

Graph-level OOD detection

1. KDD 2023. A Data-centric Framework to Endow Graph Neural Networks with Out-Of-Distribution Detection Ability.
2. WSDM 2023. GOOD-D: On Unsupervised Graph Out-Of-Distribution Detection
3. NeurIPS 2022. GraphDE: A Generative Framework for Debiased Learning and Out-of-Distribution Detection on Graphs.
4. ICML workshop 2022. Towards OOD Detection in Graph Classification from Uncertainty Estimation Perspective.

Node-level OOD detection

1. ICLR 2023. Energy-based out-of-distribution detection for graph neural networks.
2. KDD 2022. Learning on Graphs with Out-of-Distribution Nodes.
3. NeurIPS 2021. Graph Posterior Network: Bayesian Predictive Uncertainty for Node Classification.
4. NeurIPS 2020. Uncertainty Aware Semi-Supervised Learning on Graph Data.

Anomaly Detection

1. NeurIPS 2022. Dual-discriminative Graph Neural Network for Imbalanced Graph-level Anomaly Detection.
2. ICML 2022. Rethinking Graph Neural Networks for Anomaly Detection.

Q&A

Thanks for your attention!

Background | graph learning

Graph data $D \rightarrow$ GNN $f \rightarrow$ representation $Z \rightarrow \tilde{Y} \longleftrightarrow Y$