Trustworthy Learning of Graph Neural Networks

Cheng Yang
yangcheng@bupt.edu.cn
Beijing University of Posts and Telecommunications

Outline

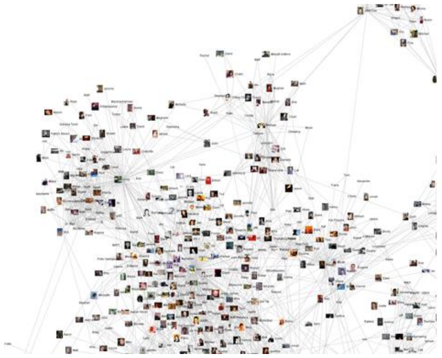
- Background
- Trustworthy GNNs
- Our Recent Attempts
- Future Directions

Outline

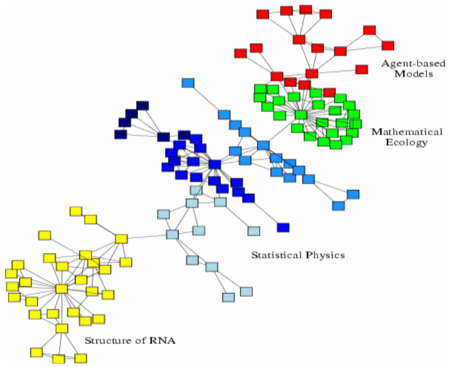
- Background
- Trustworthy GNNs
- Our Recent Attempts
- Future Directions

What & Why Graphs

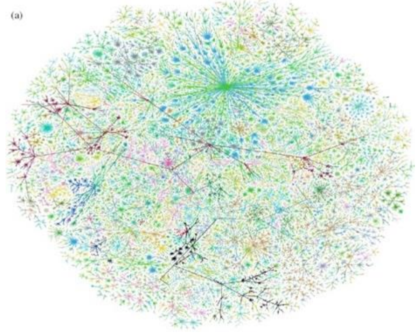
Graph (network) is a common language for describing relational data.



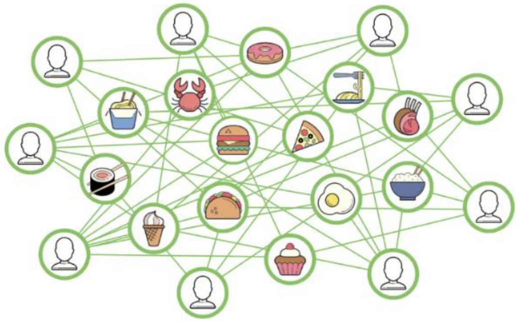
Social Network



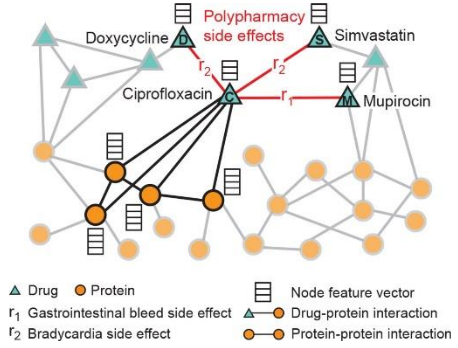
Citation Network



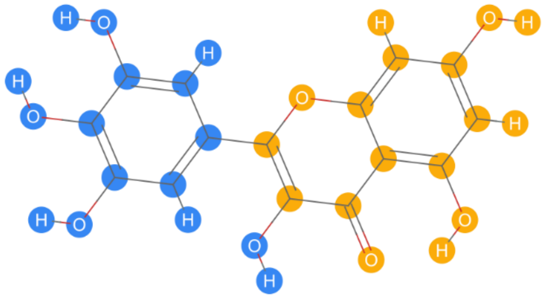
Internet



User-item Graph

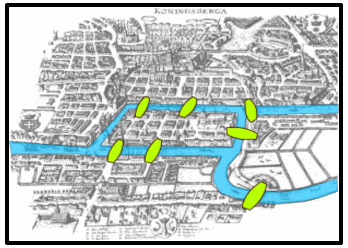


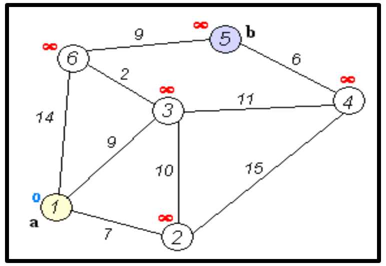
Drug Interaction Graph

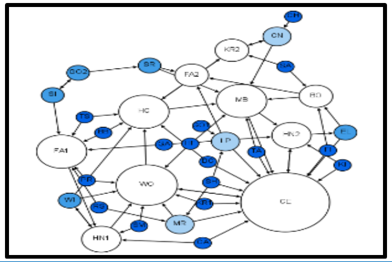


Molecule Graph

A History of Graph Theory & Learning







Graph Theory

- Euler’s seven bridges

Graph Algorithm

- Dijkstra's shortest path

Graph Models

- Random graph, Stochastic block model, Scale-free network…

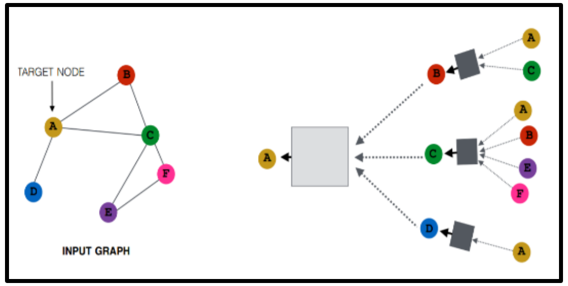
1736 — 1950s — 1990s — 2000s — 2010s — 2020s

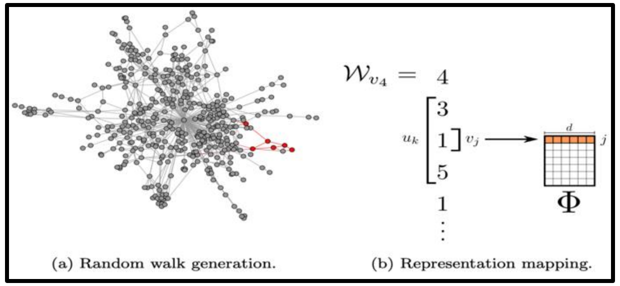
Graph Neural Network

- GCN, GAT…

Graph Embedding

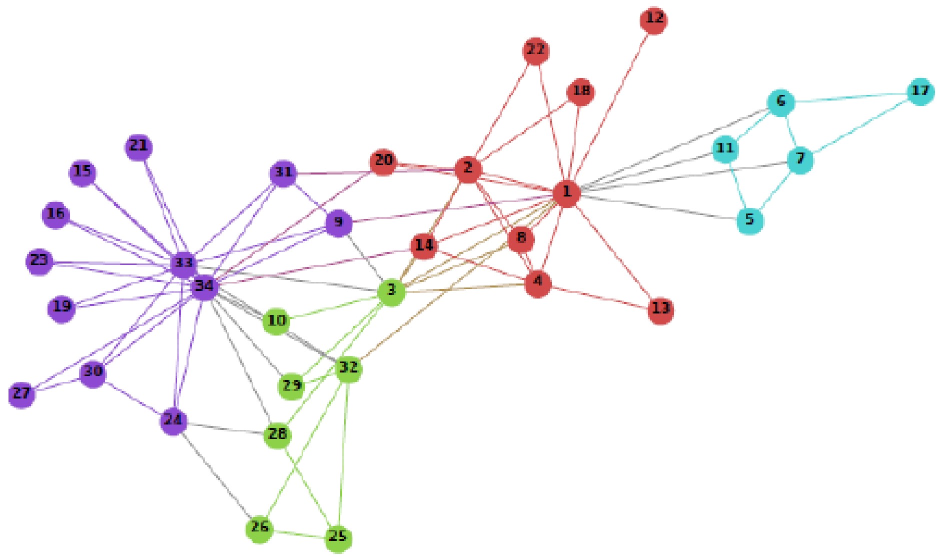
- Laplacian Eigenmap, DeepWalk…

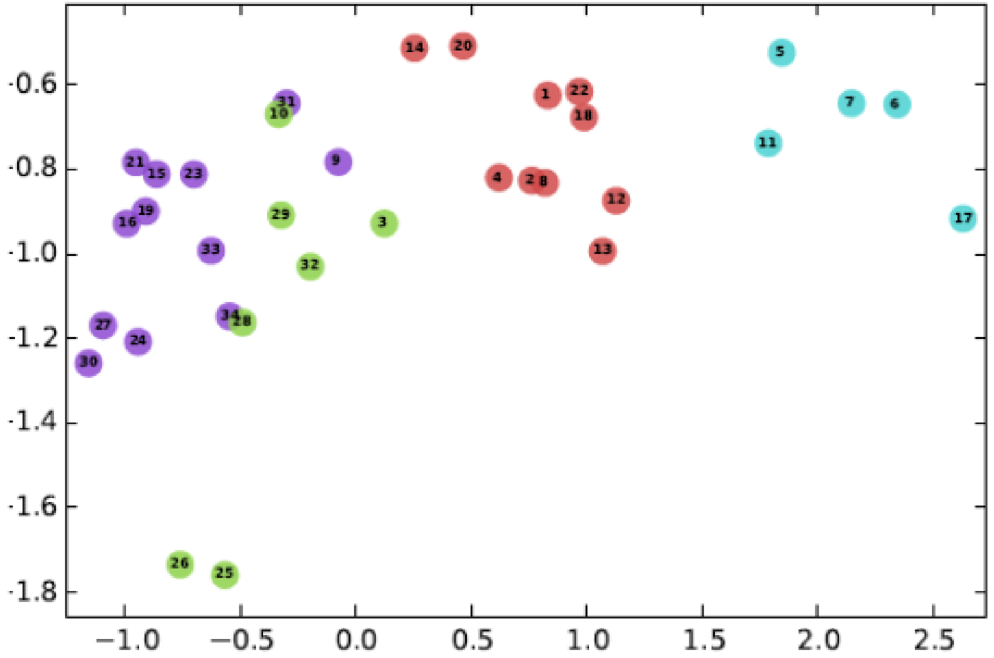




Graph Embedding

Core idea: projecting nodes in a graph into vectors in a Euclidean space.

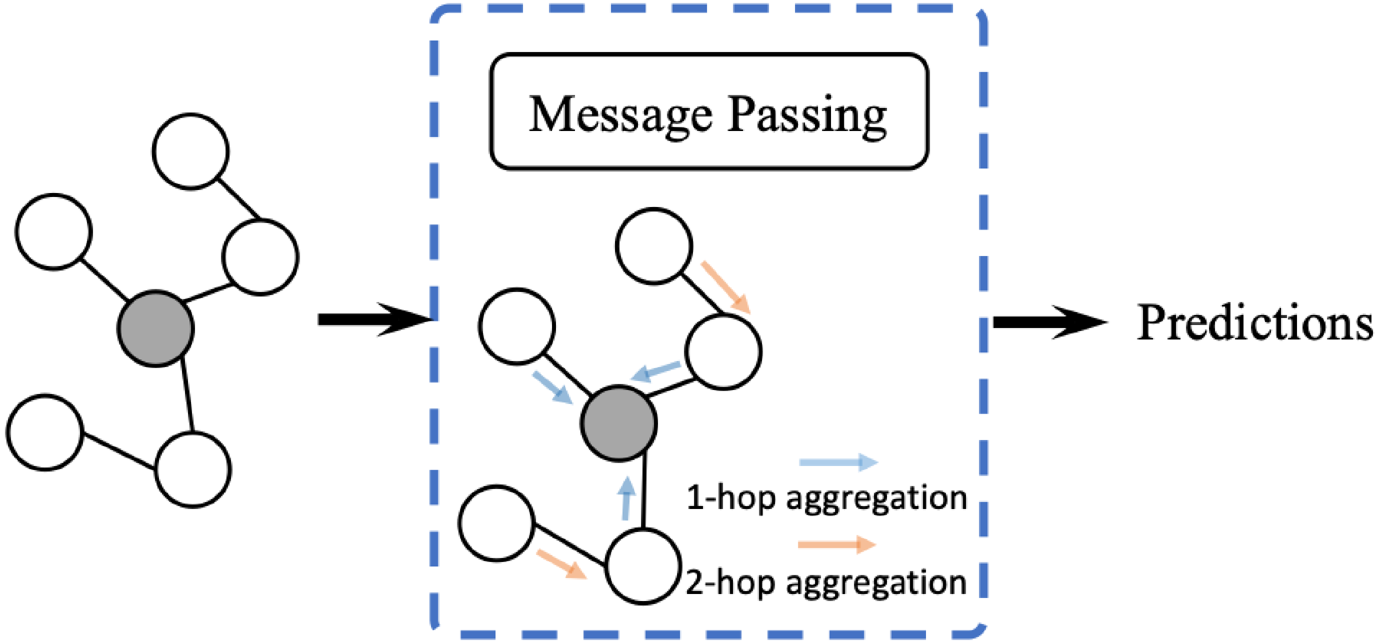




DeepWalk: Online Learning of Social Representations. KDD 2014.

Graph Neural Network (GNN)

Core idea: iteratively aggregating the embeddings of neighborhood nodes.



Graph Neural Network (GNN)

- **Graph Neural Network (GNN)**
  - **Propagation Module**
    - Convolution Operator
      - Spectral
        - SpectralCNN (ICLR 14)
        - ChebNet (NeuraIPS 16)
        - GCN (ICLR 17)
      - Spatial
        - Basic
          - GraphSAGE (NeurIPS 17)
          - MPNN (PMLR 17)
          - FastGNN (ICLR 18)
        - Attention
          - GAT (ICLR 18)
          - HGAT (IJCAI 19)
          - DAN (NeurIPS 19)
    - Recurrent Operator
      - GraphESN (IJCNN 10)
      - Graph-LSTM (TACL 17)
      - LP-GNN (TPAMI 21)
    - Skip Connection
      - Highway GCN (ACL 18)
      - JKN (ICML 18)
      - DeepGCNs (ICCV 19)
  - **Sampling Module**
    - Node
      - GraphSAGE (NeurIPS 17)
      - VR-GCN (ICML 18)
      - PinSAGE (KDD 18)
    - Layer
      - FastGNN (ICLR 18)
      - LADIES (NeurIPS 19)
    - Subgraph
      - ClusterGCN (KDD 19)
      - GraphSAINT (ICLR 20)
  - **Pooling Module**
    - Direct
      - Simple Pooling
      - Sort Pooling (AAAI 18)
    - Hierarchical
      - DiffPool (NeurIPS 18)
      - SAGPool (ICML 19)

Outline

- Background
- Trustworthy GNNs
- Our Recent Attempts
- Future Directions

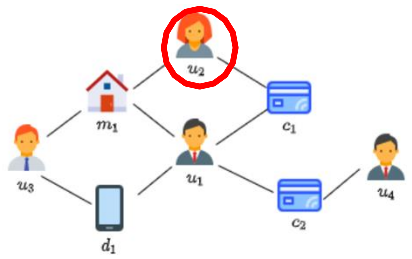
Risks in Typical GNNs

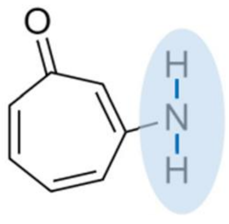
Only focusing on task performance

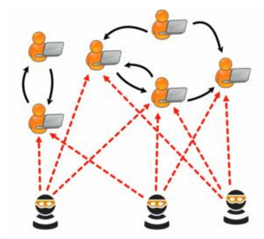
- Enhancing expressive power
- Overcoming over-smoothing issues

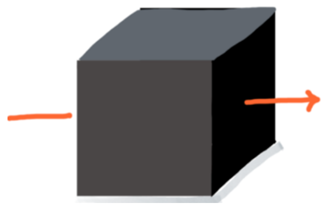
Facing **risks** of causing unintentional harm in decision-sensitive scenarios

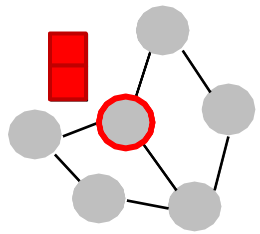
- Decision-sensitive applications
  - e.g., credit scoring systems
- Performance is not the only objective
  - Lack of fairness, robustness…

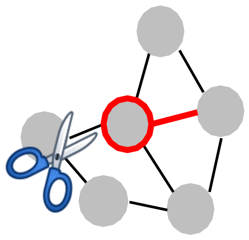












Trustworthy AI

Accuracy
How correct the prediction is?

Stability
How stable the prediction is?

Fairness
Does it treat people equally?

Explainability
Can it explain the predictions?

Privacy
Does it protect a person’s identity and data?

Robustness
How vulnerable it is to attack?

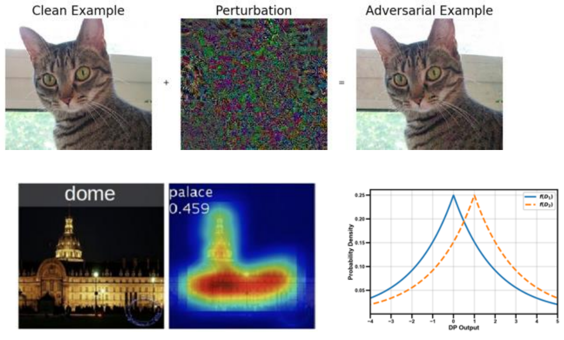
Accountability
Who is responsible when AI goes wrong?

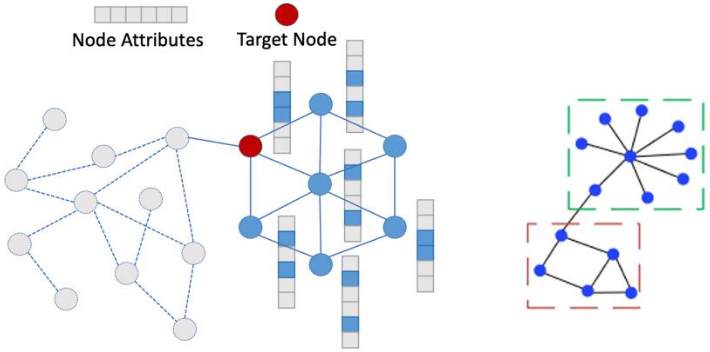
Environmental Well-being
Is it aligned to people’s expectations regarding social good?

From Trustworthy AI to Trustworthy GNNs

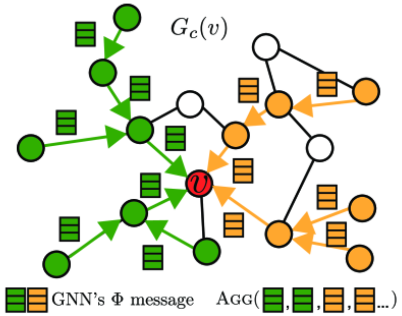
Challenges

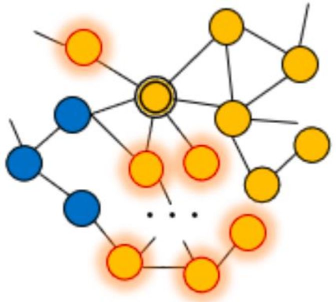
- Complex of the graph data
  - Various formats of data
  - Discreteness of graph structure

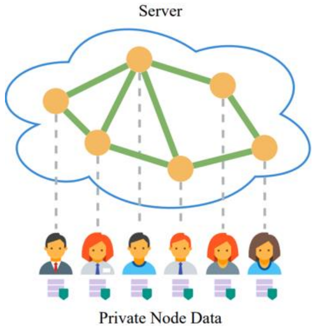




- Unique model design
  - message-passing mechanism



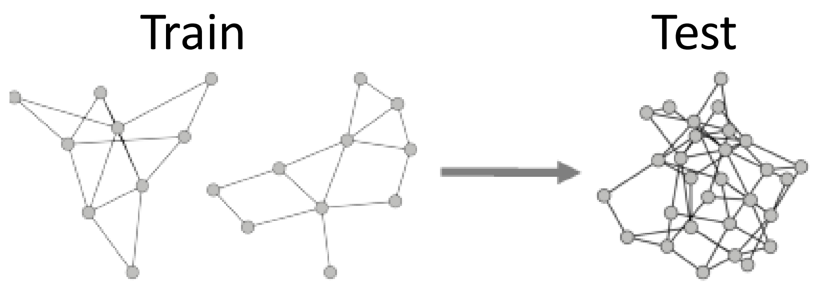




Trustworthy GNNs

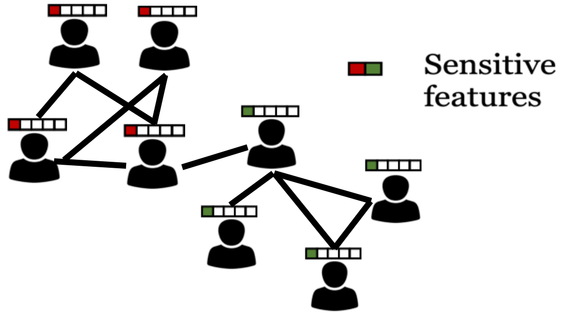
Stable GNNs

Produce stable prediction under distribution shifts



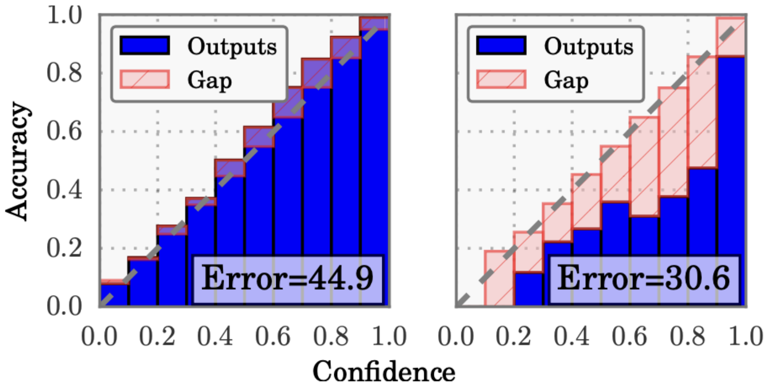
Fair GNNs

Alleviate bias in feature and topology



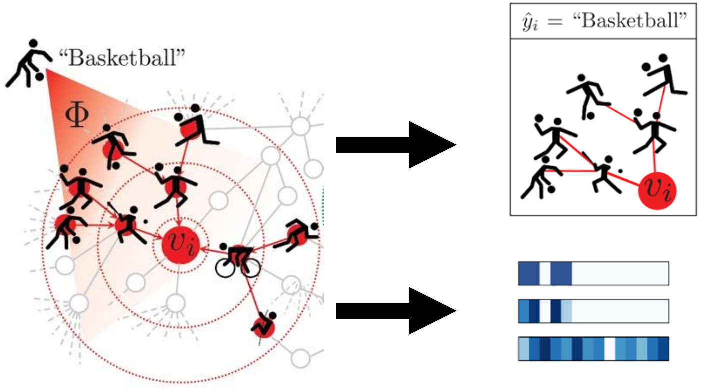
Confidence-aware GNNs

Be aware of prediction uncertainty



Explainable GNNs

Explain based on feature and topology



Outline

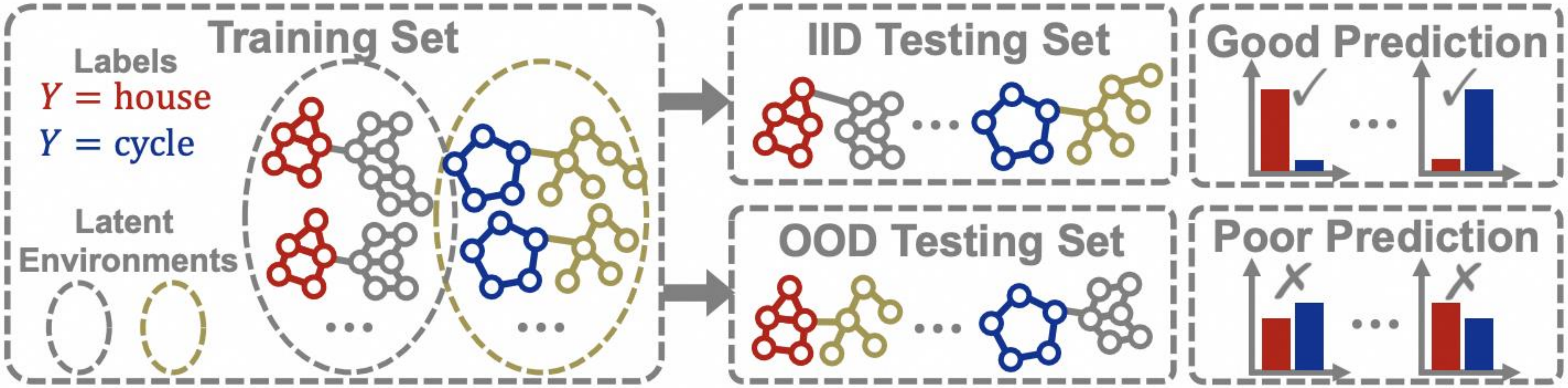
- Background
- Trustworthy GNNs
- Our Recent Attempts
- Future Directions

Our Recent Attempts

- Stable
  - A Data-centric Framework to Endow Graph Neural Networks with Out-Of-Distribution Detection Ability (AAGOD, KDD 2023)
  - Graph Invariant Learning with Subgraph Co-mixup for Out-of-distribution Generalization (IGM, AAAI 2024)
- Fair
  - FairSIN: Achieving Fairness in Graph Neural Networks through Sensitive Information Neutralization (FairSIN, AAAI 2024)
  - Endowing Pre-trained Graph Models with Provable Fairness (GraphPAR, WWW 2024)
- Confidence-aware
  - Calibrating Graph Neural Networks from a Data-centric Perspective (DCGC, WWW 2024)

Generalizing GNNs on OOD graphs

- Various forms of distribution shifts between the training and testing datasets widely exist in the real world, resulting in OOD scenarios.
  - Basic assumption (IID): Training/testing graphs are drawn from the same distribution
  - Practical situation (OOD): Training/testing graphs come from different distributions
  - Poor generalization caused by spurious correlation between subgraphs
- Approaches
  - OOD detection: identify test examples that deviate from the training distribution
  - OOD generalization: directly generalize to test examples from a different distribution



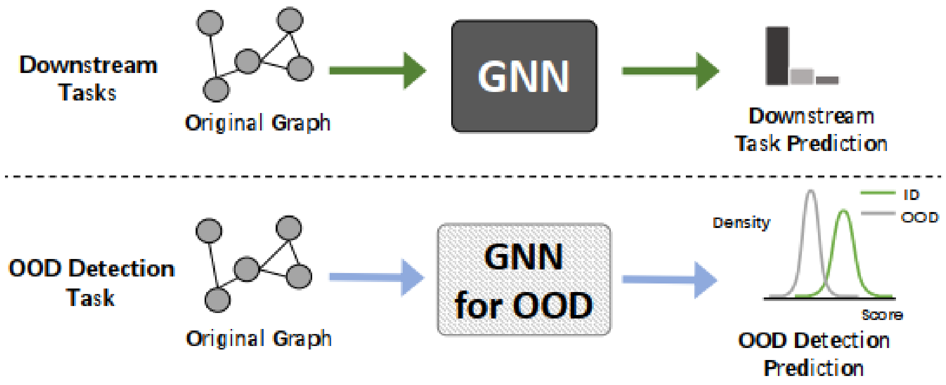
OOD scenarios

Motivation of AAGOD

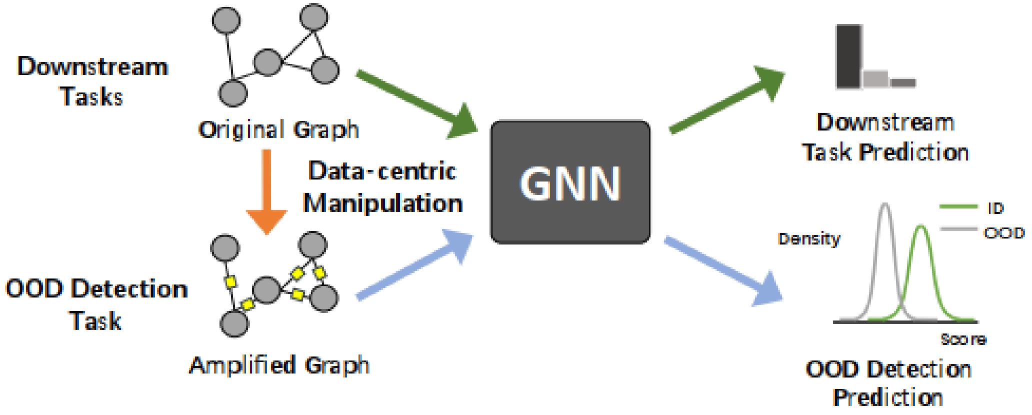
Motivation

- A reliable GNN should not only perform well on know samples (ID) but also identify graphs it has not been exposed to before (OOD) .
- Existing works proposes to train a neural network specialized for the OOD detection task.

Can we build a graph prompt that can solve OOD detection given a well-trained GNN?



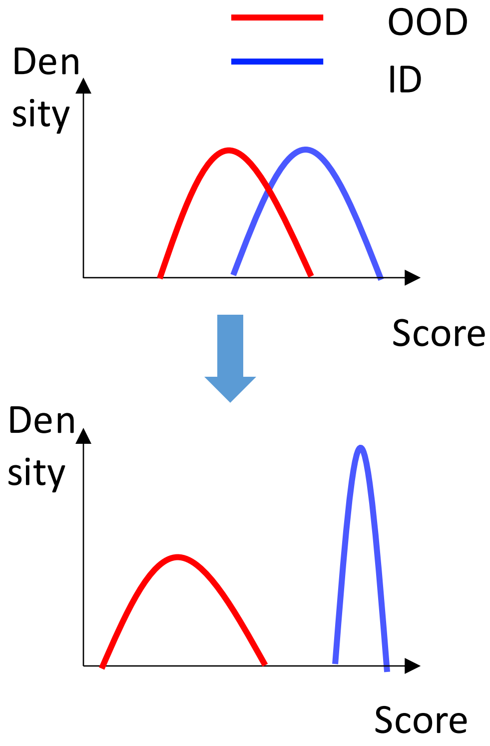
(1) Traditional works

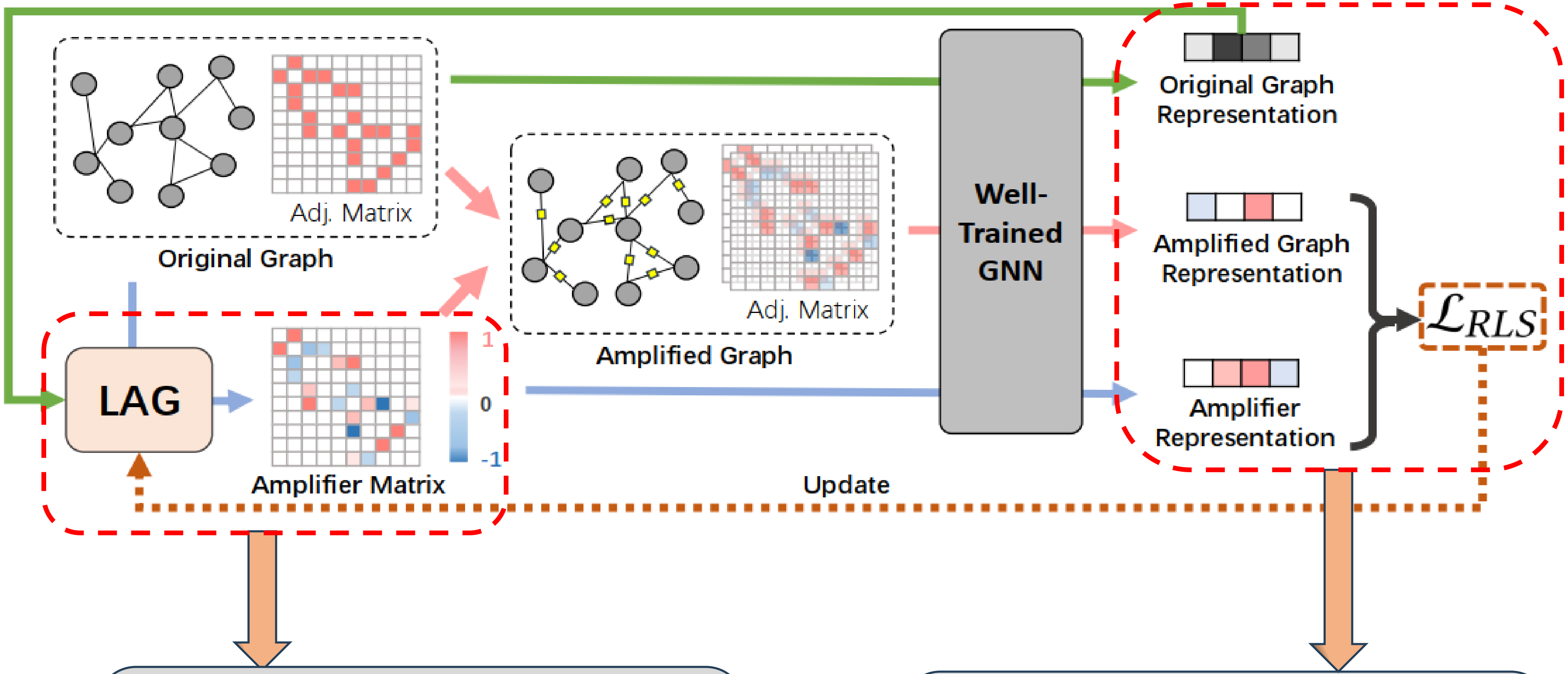


(2) Our proposed framework

AAGOD

We modify edge weights as prompts to highlight the latent pattern of ID graphs, and thus enlarge the score gap between OOD and ID graphs.





LAG adaptively generates graph-specific amplifiers by converting node representations into edge weights.

RLS encourages high scores for amplified ID graphs and expects low scores when only seeing the amplifiers.

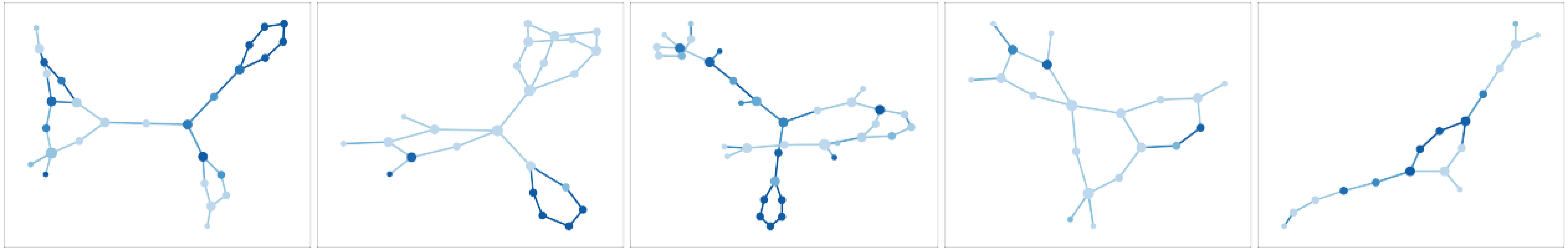
Experiments

We conducted experiments on five dataset pairs over four GNNs to verify performance.

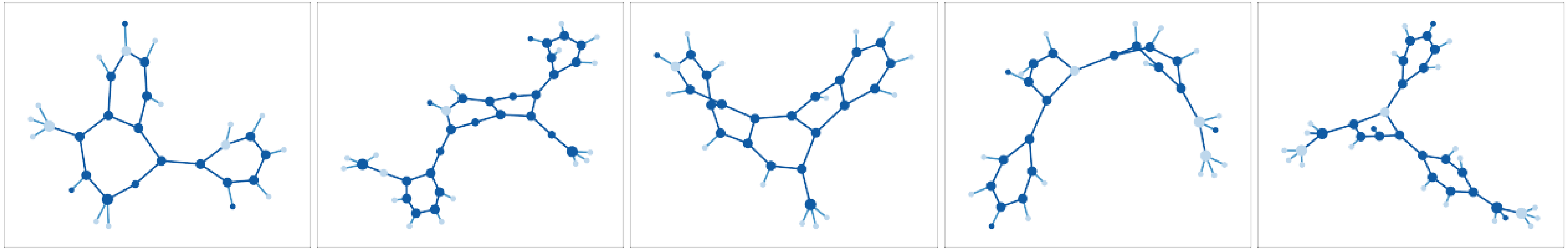
ID	OOD	Metric	GCL_S	GCL_S+	Improv.	GCL_L	GCL_L+	Improv.	$JOAO_S$	$JOAO_S$+	Improv.	$JOAO_L$	$JOAO_L$+	Improv.
ENZYMES	PROTEIN	AUC ↑	62.97	**73.76**	+17.14%	62.56	**67.15**	+7.34%	61.20	**74.19**	+21.23%	59.68	**65.11**	+9.10%
		AUPR ↑	62.47	**75.27**	+20.49%	**65.45**	65.18	-0.41%	61.30	**77.10**	+25.77%	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%

Experiments

Case study: We visualize the learned graph prompts (i.e., amplifiers) for interpretability analysis.



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



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

Motivation of IGM

- Invariant learning aims to disentangle invariant and environment parts in data.
    - combinations of invariant/environment need to be diverse enough
- Mixup may help generate data with diverse combinations!
- However, previous mixup methods operate on graph level
    - fail to reduce the spurious correlation between invariant and environment subgraphs

(a) **Inferred environment 1** *(mostly) landbirds on land, and waterbirds on water*

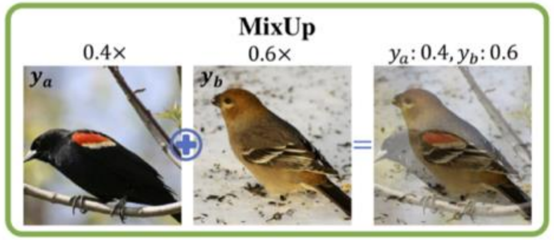
(b) **Inferred environment 2** *(mostly) landbirds on water, and waterbirds on land*

Data of different environments

Train with invariant constraints on each environment

Learned invariant feature

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j,$$
$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j,$$

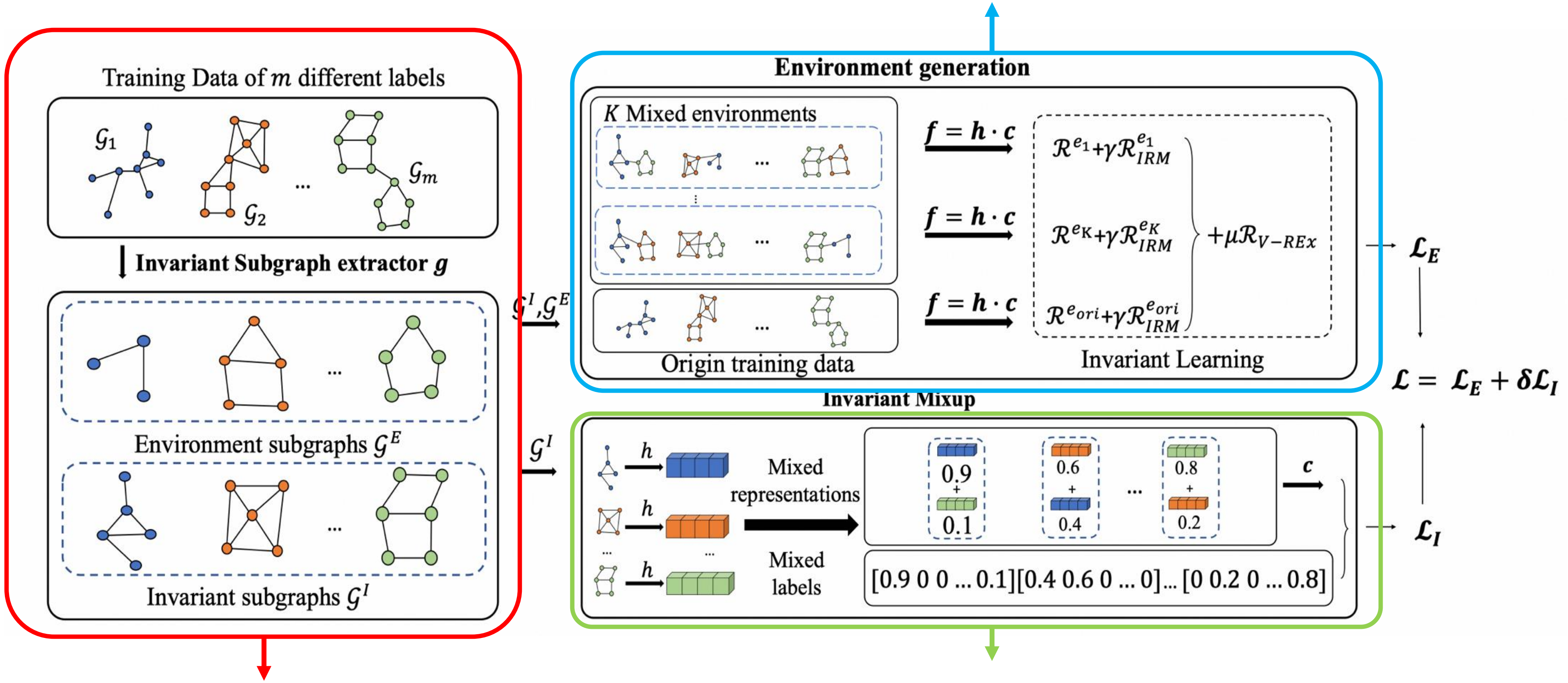


Mixup

Can we introduce subgraph-level mixup to help disentangle invariant/environment information?

IGM

Environment Mixup: generate environments with enough difference for IL (Invariant Learning)



Subgraph extractor: Learnable subgraph extractor

Invariant Mixup: conduct Mixup on extracted invariant subgraphs

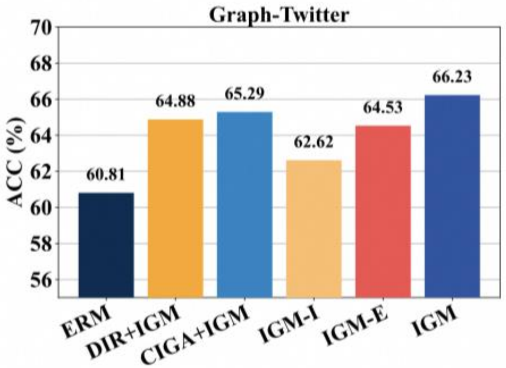
Experiments

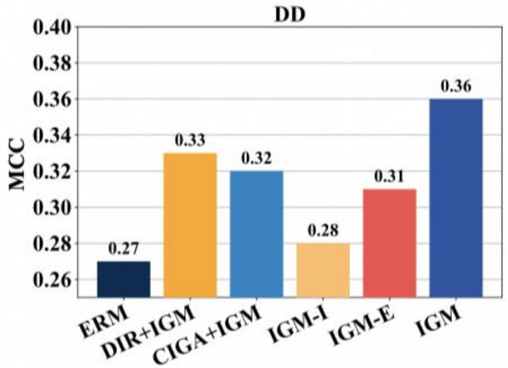
Experiments on real-world datasets and synthetic datasets

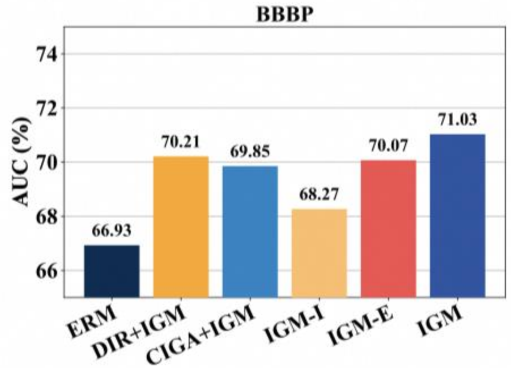
Shift Type	Degree		Size		Structure(Assay, Scaffold)			
Dataset	Graph-SST5	Graph-Twitter	PROTEINS	DD	$\text{DrugOOD}_{\text{Assay}}$	$\text{DrugOOD}_{\text{Scaffold}}$	BACE	BBBP
Metric	ACC (%)		MCC		AUC (%)			
ERM	43.89 ± 1.73	60.81 ± 2.05	0.22 ± 0.09	0.27 ± 0.09	76.41 ± 0.73	66.83 ± 0.93	77.83 ± 3.49	66.93 ± 2.31
G-Mixup	43.75 ± 1.34	63.91 ± 3.01	0.24 ± 0.03	0.29 ± 0.04	76.53 ± 2.20	66.01 ± 1.35	79.12 ± 2.75	68.44 ± 2.08
Manifold-Mixup	43.11 ± 0.65	62.60 ± 1.87	0.23 ± 0.04	0.28 ± 0.06	77.02 ± 1.15	65.56 ± 0.44	78.85 ± 1.26	68.67 ± 1.38
IRM	43.69 ± 1.26	63.50 ± 1.23	0.21 ± 0.09	0.22 ± 0.08	74.03 ± 0.58	66.32 ± 0.27	77.51 ± 2.46	69.13 ± 1.45
V-REx	43.28 ± 0.52	63.21 ± 1.57	0.22 ± 0.06	0.21 ± 0.07	75.85 ± 0.78	65.37 ± 0.42	76.96 ± 1.88	64.86 ± 2.13
EIIL	42.98 ± 1.03	62.76 ± 1.72	0.20 ± 0.05	0.23 ± 0.10	76.93 ± 1.44	64.13 ± 0.89	79.36 ± 2.72	65.77 ± 3.36
DIR	41.12 ± 1.96	59.85 ± 2.98	0.25 ± 0.14	0.20 ± 0.10	74.11 ± 3.10	64.45 ± 1.69	79.93 ± 2.03	69.73 ± 1.54
GSAT	43.72 ± 0.87	62.50 ± 1.44	0.21 ± 0.06	0.28 ± 0.04	76.64 ± 2.82	66.02 ± 1.13	79.63 ± 1.87	68.48 ± 2.01
CIGA	44.71 ± 1.14	64.45 ± 1.99	0.40 ± 0.06	0.29 ± 0.08	76.15 ± 1.21	67.11 ± 0.33	80.98 ± 1.25	69.65 ± 1.32
IGM	**46.69 ± 0.52**	**66.23 ± 1.58**	**0.43 ± 0.05**	**0.36 ± 0.04**	**78.16 ± 0.65**	**68.32 ± 0.48**	**82.65 ± 1.17**	**71.03 ± 0.79**

Dataset	SPMotif-0.33	SPMotif-0.6
ERM	59.49 ± 3.50	55.48 ± 4.84
G-mixup	60.31 ± 2.89	58.74 ± 5.58
Manifold-mixup	58.33 ± 4.05	56.63 ± 2.96
IRM	57.15 ± 3.98	61.74 ± 1.32
V-REx	54.64 ± 3.05	53.60 ± 3.74
EIIL	56.48 ± 2.56	60.07 ± 4.47
DIR	58.73 ± 11.9	48.72 ± 14.8
GSAT	56.21 ± 7.08	55.32 ± 6.35
CIGA	77.33 ± 9.13	69.29 ± 3.06
IGM	**82.36 ± 7.39**	**78.09 ± 5.63**

Ablation study

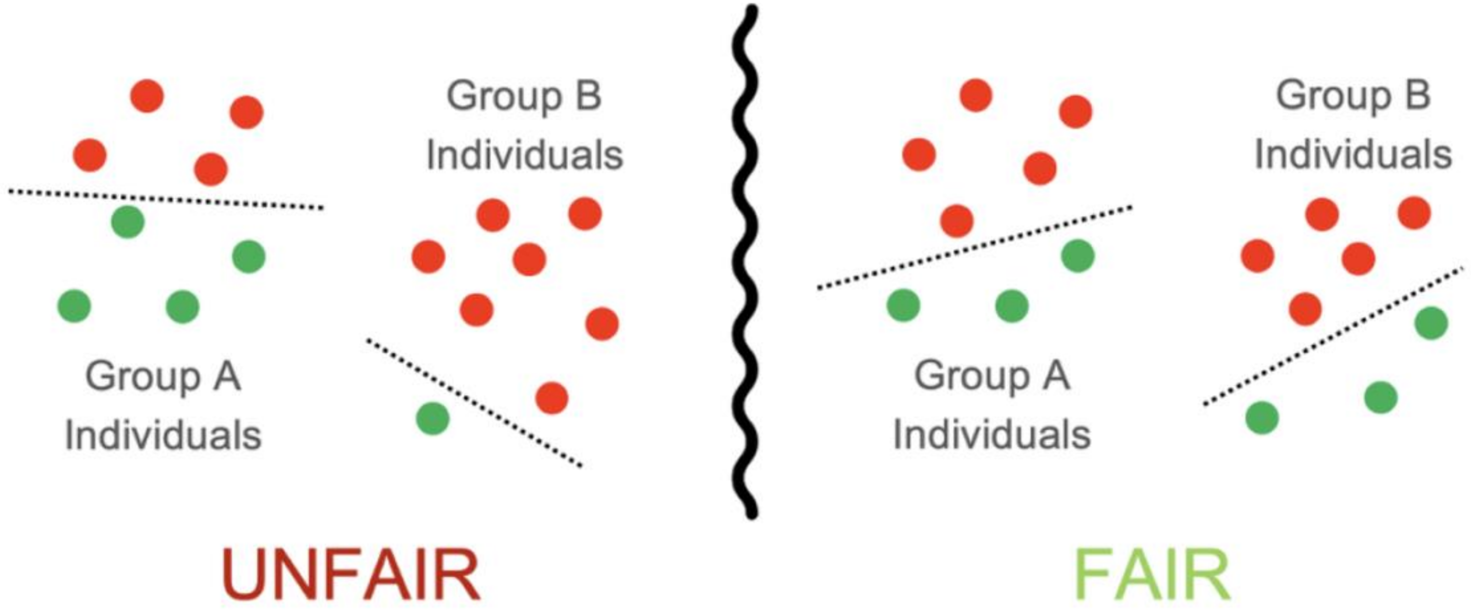






Improving GNNs for Fair Predictions

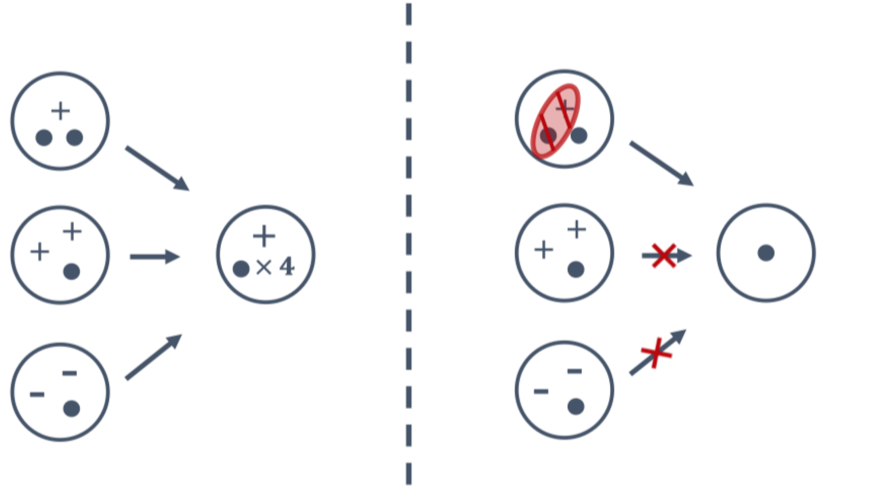
- Fairness issue: the predictions of GNNs could be biased towards some demographic groups defined by sensitive attributes, e.g., age or gender.
  - may bring about severe societal concerns in applications such as credit evaluation
- Reasons behind…
  - raw node features could be statistically correlated to the sensitive attribute
  - nodes with the same sensitive attribute tend to link with each other, making representations in the same sensitive group more similar during message passing



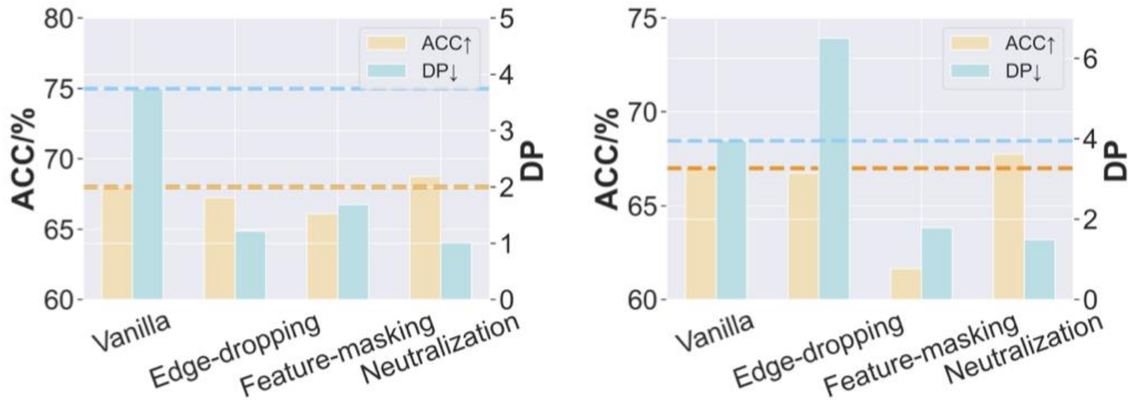
Motivation of FairSIN

Motivation

- Previous fair GNNs are usually filtering-based
  - e.g., masking features or dropping edges that could cause sensitive information leakage
  - may lose much non-sensitive information as well
  - leading to a decline in prediction performance



(a) Original Message Passing **(b) Filtering-Based Method**

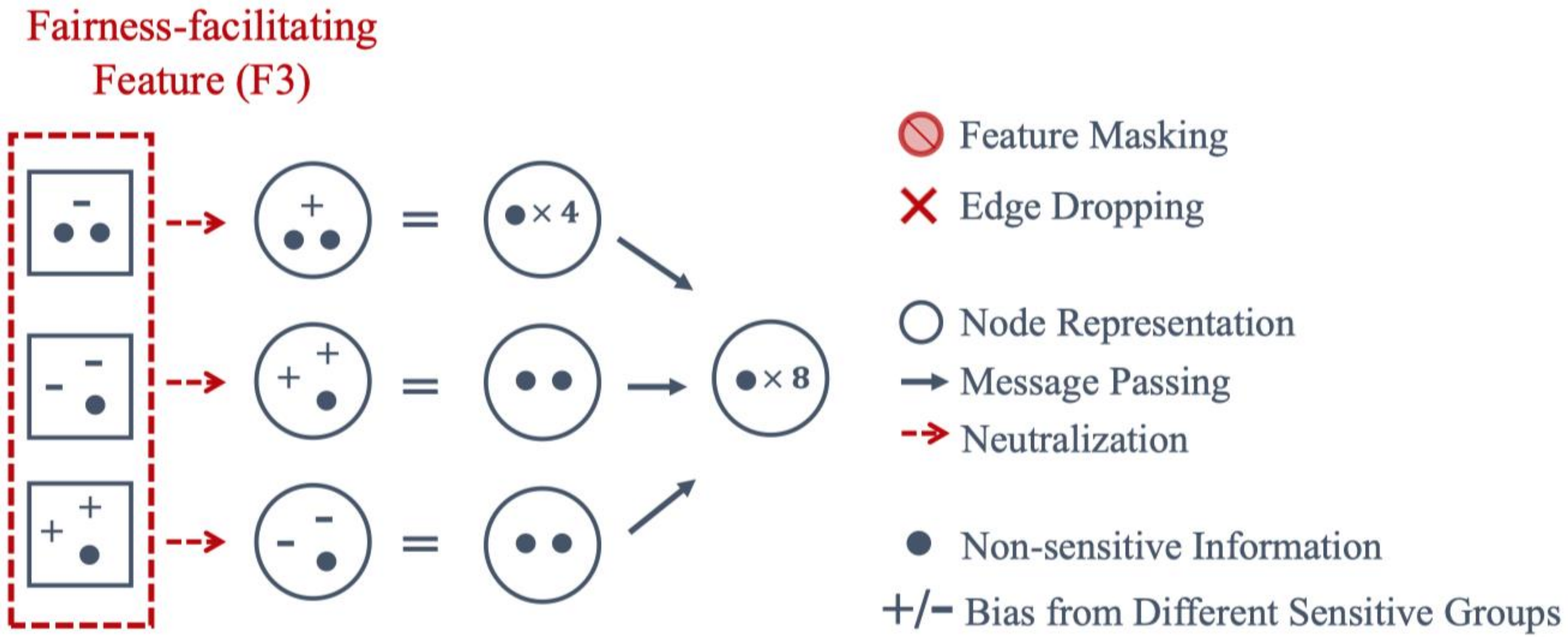


(a) **Pokec-n** (b) **Pokec-z**

Can we go beyond the filtering-based paradigm for fair GNNs?

FairSIN

- We propose a novel neutralization-based paradigm
  - introducing extra features or edges to statistically neutralize sensitive bias and provide additional non-sensitive information.

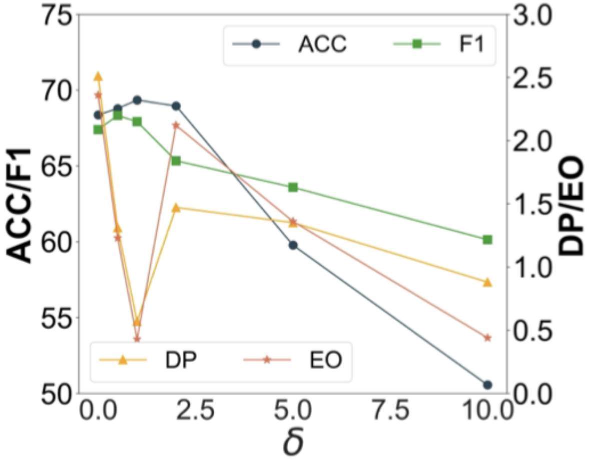


(c) Neutralization-based Method (Ours)

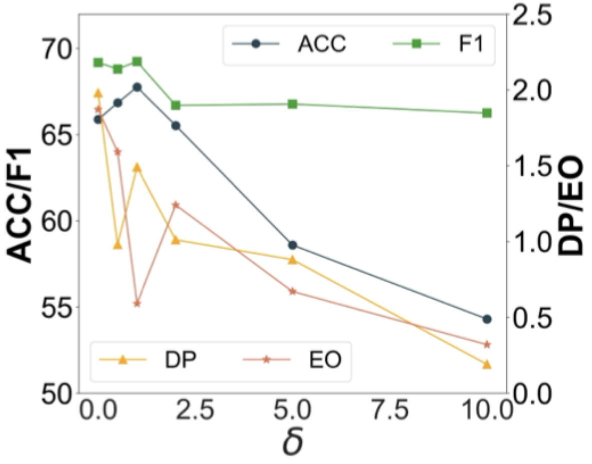
Experiments

Encoder	Method	Bail				Pokec_n				Pokec_z			
		F1↑	ACC↑	DP↓	EO↓	F1↑	ACC↑	DP↓	EO↓	F1↑	ACC↑	DP↓	EO↓
GCN	vanilla	82.04±0.74	87.55±0.54	6.85±0.47	5.26±0.78	67.74±0.41	68.55±0.51	3.75±0.94	2.93±1.15	69.99±0.41	66.78±1.09	3.95±1.03	2.76±0.95
	FairGNN	77.50±1.69	82.94±1.67	6.90±0.17	4.65±0.14	65.62±2.03	67.36±2.06	3.29±2.95	2.46±2.64	70.86±2.36	67.65±1.65	1.87±1.95	1.32±1.42
	EDITS	75.58±3.77	84.49±2.27	6.64±0.39	7.51±1.20	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	NIFTY	74.76±3.91	82.36±3.91	5.78±1.29	4.72±1.08	64.02±1.26	67.24±0.49	1.22±0.94	2.79±1.24	69.96±0.71	66.74±0.93	6.50±2.16	7.64±1.77
	FairVGNN	79.11±0.33	84.73±0.46	6.53±0.67	4.95±1.22	64.85±1.17	66.10±1.45	1.69±0.79	1.78±0.70	67.31±1.72	61.64±4.72	1.79±1.22	1.25±1.01
	FairSIN-G	79.61±1.29	85.57±1.08	6.57±0.29	5.55±0.84	67.80±0.63	68.22±0.39	2.56±0.60	1.69±1.29	69.68±0.86	65.73±1.76	3.53±1.20	2.42±1.43
	FairSIN-F	82.23±0.63	87.61±0.83	5.54±0.40	3.47±1.03	66.30±0.56	67.96±1.54	1.16±0.90	0.98±0.70	**69.74±0.85**	66.38±1.39	2.53±0.97	2.03±1.23
	FairSIN w/o Neutral.	81.51±0.33	87.26±0.17	5.93±0.04	4.30±0.20	67.39±0.70	68.35±0.62	2.51±1.99	2.36±1.89	69.18±0.51	65.87±1.34	1.98±1.01	1.87±0.64
	FairSIN w/o Discri.	82.05±0.41	87.40±0.15	5.65±0.40	4.63±0.52	**67.94±0.38**	68.74±0.33	2.22±1.47	1.67±1.70	69.31±0.63	66.42±1.52	2.73±1.08	2.37±0.69
	FairSIN	**82.30±0.63**	**87.67±0.26**	**4.56±0.75**	**2.79±0.89**	67.91±0.45	**69.34±0.32**	**0.57±0.19**	**0.43±0.41**	69.24±0.30	**67.76±0.71**	**1.49±0.74**	**0.59±0.50**
GIN	vanilla	77.89±1.09	83.52±0.87	7.55±0.51	6.17±0.69	67.87±0.70	69.25±1.75	3.71±1.20	2.55±1.52	69.49±0.34	65.83±1.31	1.97±1.12	2.17±0.48
	FairGNN	73.67±1.17	77.90±2.21	6.33±1.49	4.74±1.64	64.73±1.86	67.10±3.25	3.82±2.44	3.62±2.78	**69.50±2.38**	66.49±1.54	3.53±3.90	3.17±3.52
	EDITS	68.07±5.30	73.74±5.12	6.71±2.35	5.98±3.66	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	NIFTY	70.64±6.73	74.46±9.98	5.57±1.11	**3.41±1.43**	61.82±3.25	66.37±1.51	3.84±1.05	3.24±1.60	67.61±2.23	65.57±1.34	2.70±1.28	3.23±1.92
	FairVGNN	76.36±2.20	83.86±1.57	5.67±0.76	5.77±0.76	68.01±1.08	68.37±0.97	1.88±0.99	1.24±1.06	68.70±0.89	65.46±1.22	1.45±1.13	1.21±1.06
	FairSIN-G	79.69±0.62	86.10±1.39	6.93±0.16	6.75±0.66	67.16±1.03	67.73±1.67	1.98±1.54	1.50±1.15	68.84±1.96	65.09±2.69	1.55±1.23	1.74±0.80
	FairSIN-F	80.37±0.84	86.48±0.75	5.95±1.85	5.97±2.07	68.36±0.55	68.92±1.08	1.51±1.11	**0.82±0.79**	68.96±1.08	65.97±0.82	1.45±1.15	1.14±0.73
	FairSIN w/o Neutral.	79.33±0.64	85.27±0.70	7.21±0.39	6.75±0.55	68.30±1.12	68.92±1.13	2.81±1.91	2.12±1.30	69.38±1.28	65.04±1.56	2.19±1.96	1.23±0.92
	FairSIN w/o Discri.	80.14±1.06	86.44±0.80	4.38±1.48	4.23±1.88	67.32±0.36	**70.04±0.80**	2.44±1.50	1.63±1.24	69.21±0.25	65.58±0.71	1.40±0.67	1.12±0.24
	FairSIN	**80.44±1.14**	**86.52±0.48**	**4.35±0.71**	4.17±0.96	**68.43±0.64**	69.58±0.57	**1.11±0.31**	0.97±0.59	69.06±0.54	**66.74±1.56**	**0.64±0.47**	**1.01±0.64**
SAGE	vanilla	83.03±0.42	88.13±1.12	1.13±0.48	2.61±1.16	67.15±0.88	69.03±0.77	3.09±1.29	2.21±1.60	70.24±0.46	66.55±0.69	4.71±1.05	2.72±0.85
	FairGNN	82.55±0.98	87.68±0.73	1.94±0.82	1.72±0.70	65.75±1.89	67.03±2.61	2.97±1.28	2.06±3.02	69.49±2.15	67.68±1.49	2.86±1.39	2.30±1.33
	EDITS	77.83±3.79	84.42±2.87	3.74±3.54	4.46±3.50	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	NIFTY	77.81±6.03	84.11±5.49	5.74±0.38	4.07±1.28	61.70±1.47	68.48±1.11	3.84±1.05	3.90±2.18	66.86±2.51	66.68±1.45	6.75±1.84	8.15±0.97
	FairVGNN	83.58±1.88	88.41±1.29	1.14±0.67	1.69±1.13	67.40±1.20	68.50±0.71	1.12±0.98	1.13±1.02	69.91±0.95	66.39±1.95	4.15±1.30	2.31±1.57
	FairSIN-G	83.96±1.78	**88.79±1.08**	3.97±0.92	1.70±0.66	68.08±1.10	69.11±0.62	2.00±1.13	1.66±0.70	**71.05±0.73**	66.19±1.49	4.96±0.25	2.90±1.21
	FairSIN-F	83.82±0.26	88.51±0.16	0.67±0.33	1.85±0.50	67.21±0.84	69.28±0.98	1.80±0.46	1.62±0.84	70.25±0.40	66.99±1.06	3.25±1.00	1.89±0.79
	FairSIN w/o Neutral.	82.95±0.46	87.70±0.28	0.64±0.40	2.21±0.22	67.38±0.81	68.77±0.62	2.35±0.99	1.71±0.99	69.87±1.70	67.39±1.05	2.92±1.69	1.79±1.16
	FairSIN w/o Discri.	83.49±0.34	88.46±0.19	0.82±0.51	2.12±0.55	67.14±1.09	**69.65±0.32**	1.91±0.82	1.09±1.12	70.10±0.93	66.78±0.83	3.92±1.02	1.62±0.68
	FairSIN	**83.97±0.43**	88.74±0.42	**0.58±0.60**	**1.49±0.34**	**68.38±0.83**	69.12±1.16	**1.04±0.83**	**1.04±0.42**	70.70±0.99	**67.95±0.79**	**1.74±0.73**	**0.68±0.65**

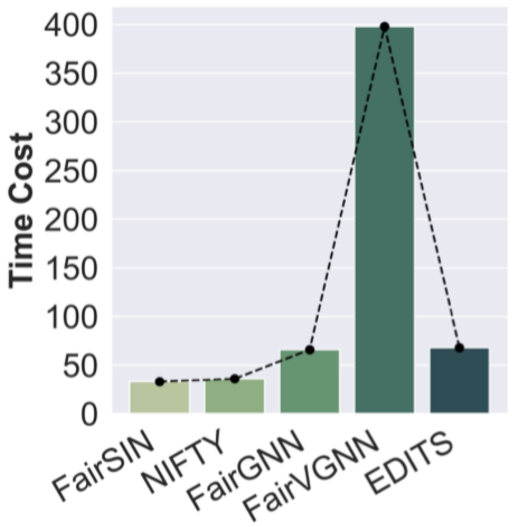
Experiments



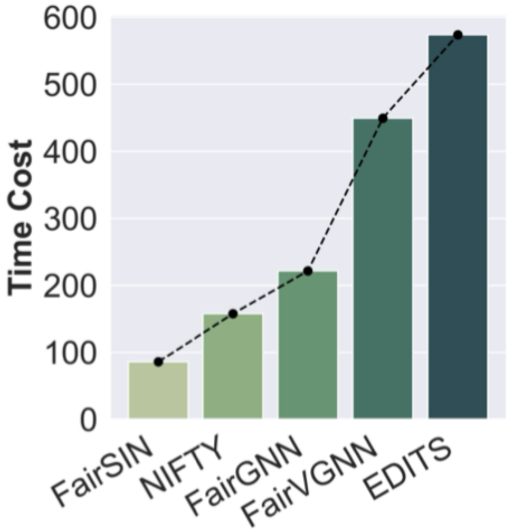
(a) Pokec-n



(b) Pokec-z



(a) Bail



(b) Credit

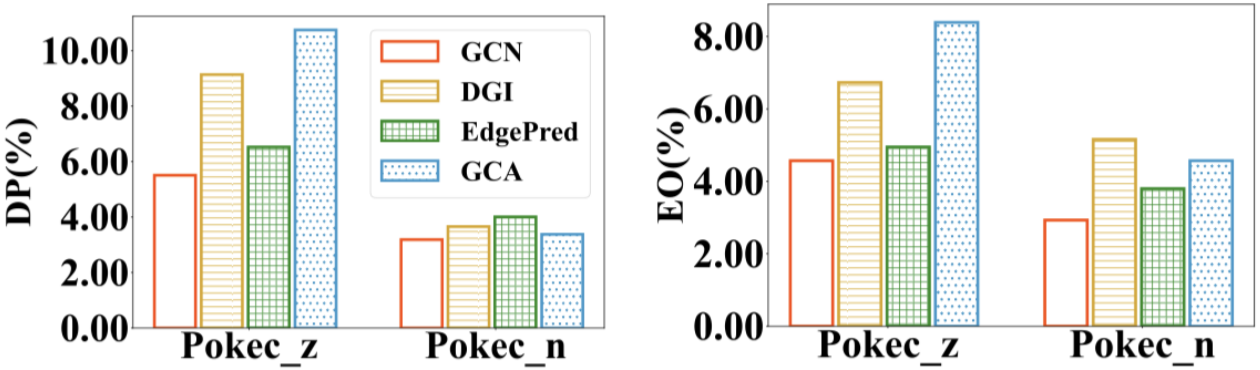
(1) Classification performance and group fairness under different values of hyper-parameter δ.

(2) Training time cost on Bail and Credit with GCN backbone (in seconds).

Motivation of GraphPAR

Do pre-trained graph models (PGMs) also inherit bias from graphs?

- Recent work [1] have demonstrated that pre-trained language models tend to inherit bias from pre-training corpora.



(a) Demographic Parity (DP). (b) Equality Opportunity (EO).

- PGMs can well capture semantic information on graphs during the pre-training phase, which inevitably contains sensitive attribute semantics.

[1] Nicholas Meade, Elinor Poole-Dayan, and Siva Reddy. 2022. An Empirical Survey of the Effectiveness of Debiasing Techniques for Pre-trained Language Models. ACL

Motivation of GraphPAR

Existing fair methods is inflexible and inefficient.

- Existing works generally train a fair GNN for a specific task.
- Debiasing for a specific task in the pre-training phase is inflexible
- Maintaining a specific PGM for each task is inefficient

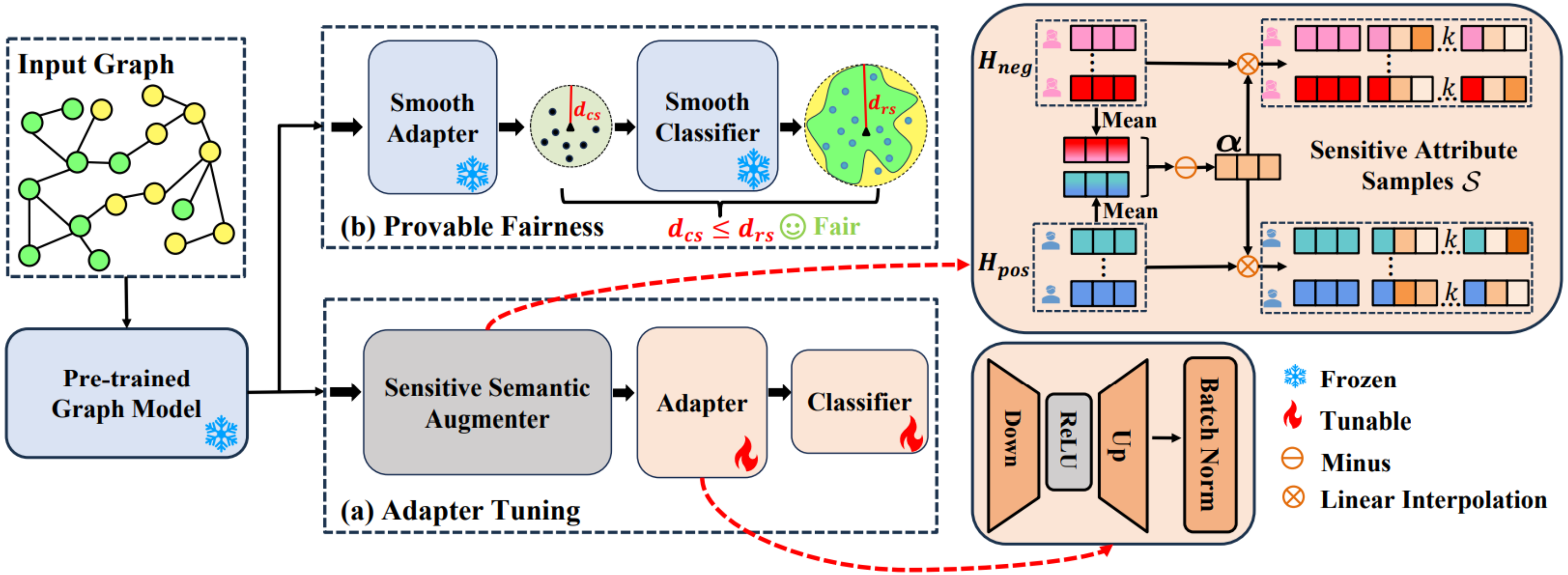
Existing fair GNN methods lack theoretical guarantees.

- No provable lower bounds on the fairness of model prediction.

How to efficiently and flexibly endow PGMs fairness with practical guarantee?

GraphPAR

Core idea: tuning an adapter so that the adapter-processed node representations are independent of sensitive attribute semantics, preventing the propagation of sensitive attribute semantics from PGMs to task predictions.



Augmenting sensitive attribute semantics

$$\boldsymbol{\alpha} = \mathbf{h}_{pos} - \mathbf{h}_{neg},$$

$$\mathrm{h}_{pos} = \frac{1}{n_{pos}} \sum_{i=1}^{n_{pos}} \mathrm{H}_{pos,i} \,, \mathrm{h}_{neg} = \frac{1}{n_{neg}} \sum_{i=1}^{n_{neg}} \mathrm{H}_{neg,i}$$

$$\mathcal{S}_i := \{\mathbf{h}_i + t \cdot \boldsymbol{\alpha} \mid |t| \leq \epsilon\} \subseteq \mathbb{R}^p,$$

Training an adapter for PGMs fairness

$$\mathcal{L}_{\text{RandAT}} = \mathbb{E}_{i \in \mathcal{V}_L} \left[\mathbb{E}_{\mathbf{h}'_i \in \hat{\mathcal{S}}_i} \left[\ell(d \circ g(\mathbf{h}'_i), y_i) \right] \right],$$

$$\mathcal{L}_{MinMax}(\mathbf{h}_i) \approx \max_{\mathbf{h}'_i \in \hat{\mathcal{S}}_i} \left\| g(\mathbf{h}_i) - g(\mathbf{h}'_i) \right\|_2 .$$

Experiments

How effective is GraphPAR compared to existing graph fairness methods?

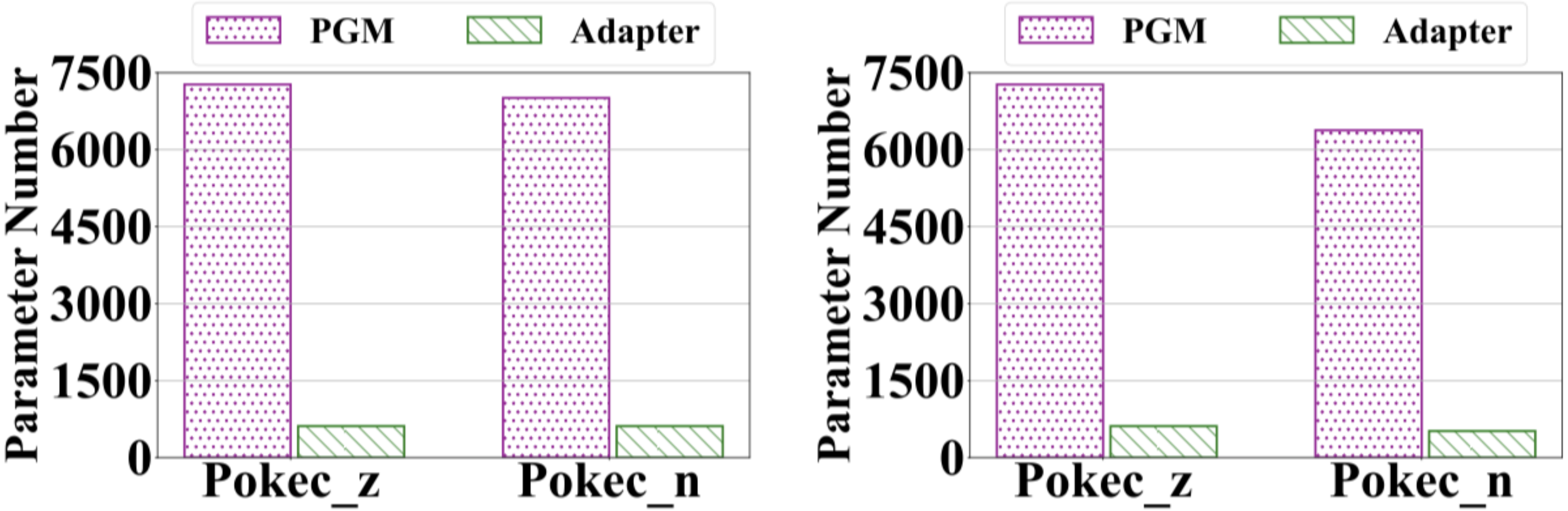
- GraphPAR outperforms baseline models both in classification and fairness performance.
- Performance of GraphPAR varies among different PGMs.
- RandAT and MinMax perform well but in different ways.

Method		Credit				Pokec_z				Pokec_n			
		ACC (↑)	F1 (↑)	DP (↓)	EO (↓)	ACC (↑)	F1 (↑)	DP (↓)	EO (↓)	ACC (↑)	F1 (↑)	DP (↓)	EO (↓)
	GCN	69.73±0.04	79.14±0.02	13.28±0.15	12.66±0.24	67.54±0.48	68.93±0.39	5.51±0.67	4.57±0.29	**70.11±0.34**	67.37±0.38	3.19±0.86	2.93±0.95
	FairGNN	72.50±4.09	81.80±3.86	9.20±3.35	7.64±3.58	67.47±1.12	69.35±3.14	1.91±1.01	1.04±1.11	68.42±2.04	64.34±2.32	1.41±1.30	1.50±1.23
	NIFTY	70.89±0.59	80.23±0.54	9.93±0.59	8.79±0.71	65.83±3.90	66.99±4.26	5.47±2.13	2.64±1.02	68.97±1.21	66.77±1.27	1.68±0.90	1.38±0.91
	EDITS	66.80±1.03	76.64±1.13	10.21±1.14	8.78±1.15	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
DGI	Naive	75.72±2.18	84.73±2.00	7.87±2.22	6.51±2.79	67.87±0.51	70.23±0.80	4.69±1.95	3.03±1.34	68.58±1.22	65.66±1.37	3.58±3.09	4.99±3.68
	GraphPAR_{RandAT}	**76.88±1.33**	**85.85±1.36**	5.93±2.91	4.44±3.34	67.05±1.33	**70.50±0.69**	1.90±1.22	0.84±0.28	68.92±1.55	65.61±1.33	**1.19±0.65**	2.11±1.60
	GraphPAR_{MinMax}	74.37±2.91	83.46±2.64	**3.81±2.37**	**2.60±2.48**	**68.32±0.55**	68.35±2.38	1.64±0.78	**0.53±0.39**	68.43±0.55	**68.20±2.22**	1.73±0.76	1.11±0.88
EdgePred	Naive	69.66±1.74	79.30±1.63	7.89±2.28	6.67±2.42	67.33±0.44	69.17±0.52	6.00±3.04	3.95±2.52	68.60±0.53	65.56±0.79	2.48±0.86	5.29±2.71
	GraphPAR_{RandAT}	69.97±2.35	79.55±2.24	6.36±2.19	4.83±2.70	66.87±1.12	68.86±0.46	1.99±1.12	2.27±1.23	68.49±1.41	65.45±1.02	1.79±0.85	3.69±0.68
	GraphPAR_{MinMax}	68.53±1.23	78.19±1.14	5.10±2.31	4.52±2.17	67.51±0.55	69.03±0.82	**1.45±1.40**	1.15±0.85	69.10±0.91	65.00±1.10	1.28±0.97	3.31±2.06
GCA	Naive	75.28±0.51	84.35±0.47	8.56±0.97	6.21±0.90	67.63±0.44	70.24±0.98	7.68±2.19	4.82±1.43	67.85±1.23	65.81±1.35	2.90±2.61	3.23±1.05
	GraphPAR_{RandAT}	75.50±1.29	84.66±1.27	5.51±2.44	3.98±1.96	66.73±2.22	70.32±0.73	4.23±2.50	2.94±1.84	68.11±0.44	64.43±1.05	2.35±1.12	2.42±1.62
	GraphPAR_{MinMax}	73.74±2.01	82.96±1.74	4.90±1.90	2.96±1.66	66.59±1.28	68.74±1.17	2.33±2.28	2.42±1.72	68.11±0.70	65.49±1.57	1.41±0.86	**0.94±0.59**

Experiments

How parameter-efficient is GraphPAR?

- The number of tuned parameters in GraphPAR is 91% smaller than in the PGM.



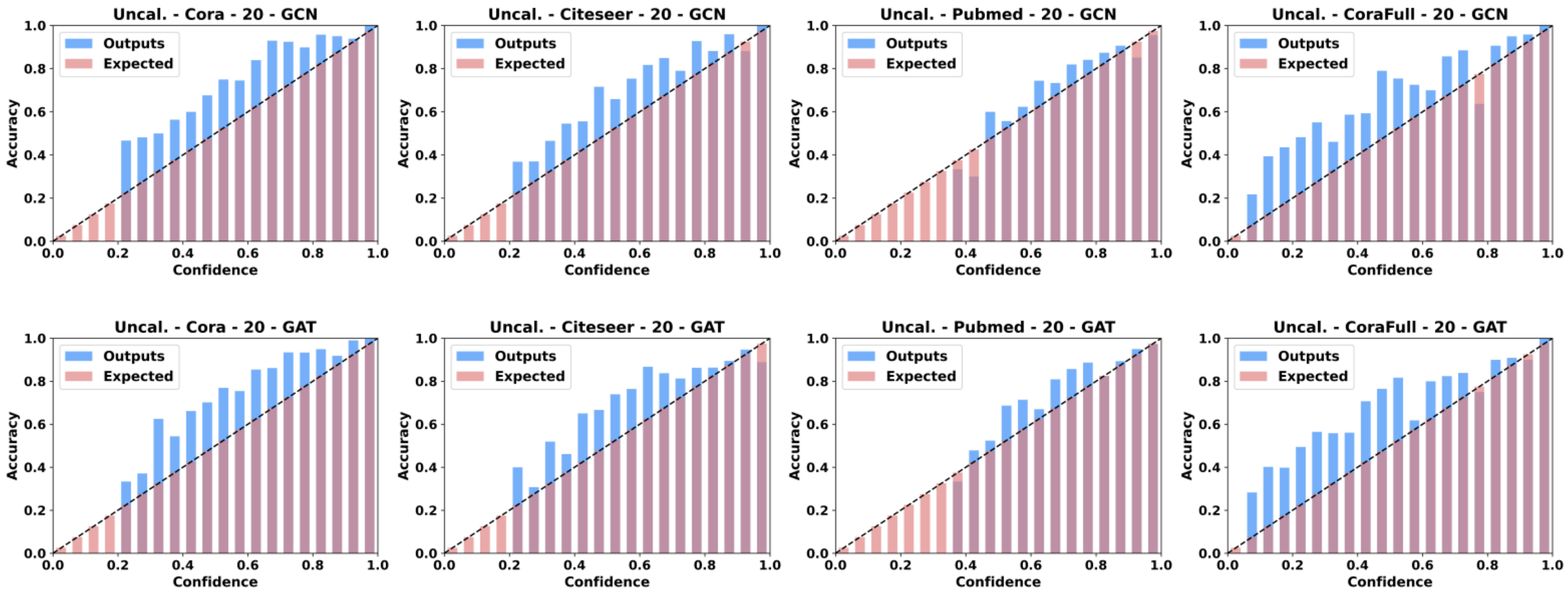
(a) Infomax.

(b) EdgePred.

Calibrating GNNs for Uncertainty Awareness

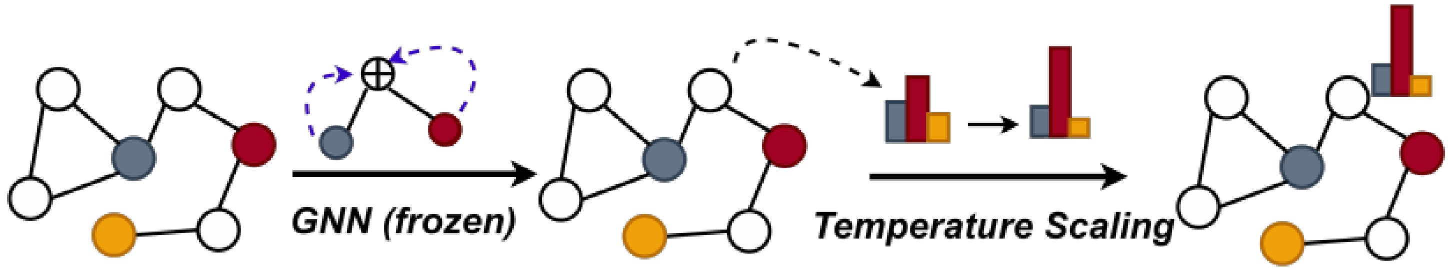
A trustworthy model should know when it is likely to be incorrect

- The confidence probability associated with the predicted class label should reflect its ground truth correctness likelihood
- Recent works show that GNNs tend to be under-confident in their predictions



Motivation of DCGC

- Existing calibration methods focus on improving GNN models. Recent work has shown that the post-hoc methods, such as temperature scaling-based calibration, can achieve a better trade-off between accuracy and calibration.

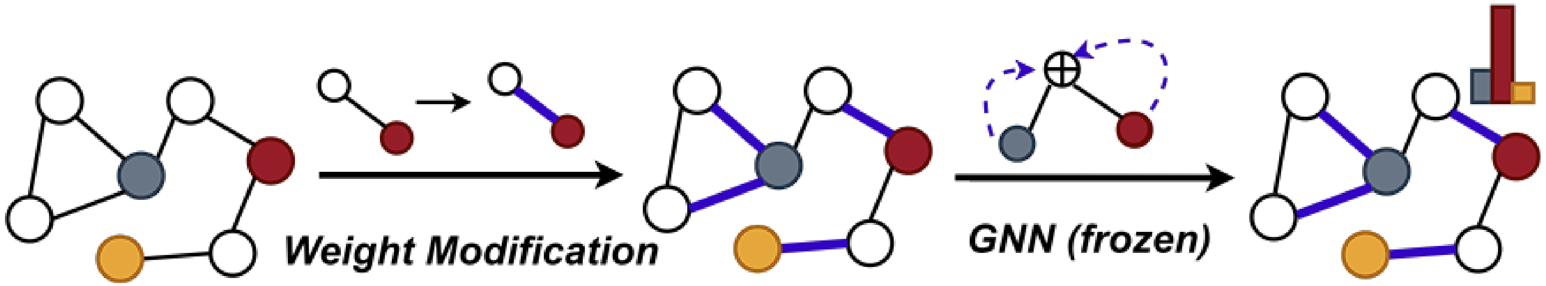


(a) Temperature scaling-based calibration

- Through evaluating the expected calibration error (ECE) on Cora and Photo datasets with five different GNNs, we find that the ECEs on Cora (10.25%-18.02%) are always larger than those on Photo (4.38%-8.27%), indicating that the calibration performance depends more on the datasets instead of GNN model.

Motivation of DCGC

- Inspired by this phenomenon, we innovatively propose to calibrate GNNs from a data-centric perspective: *can we modify the graph data instead for better calibration performance without losing accuracy?*



(b) Data-centric calibration

Observation of DCGC

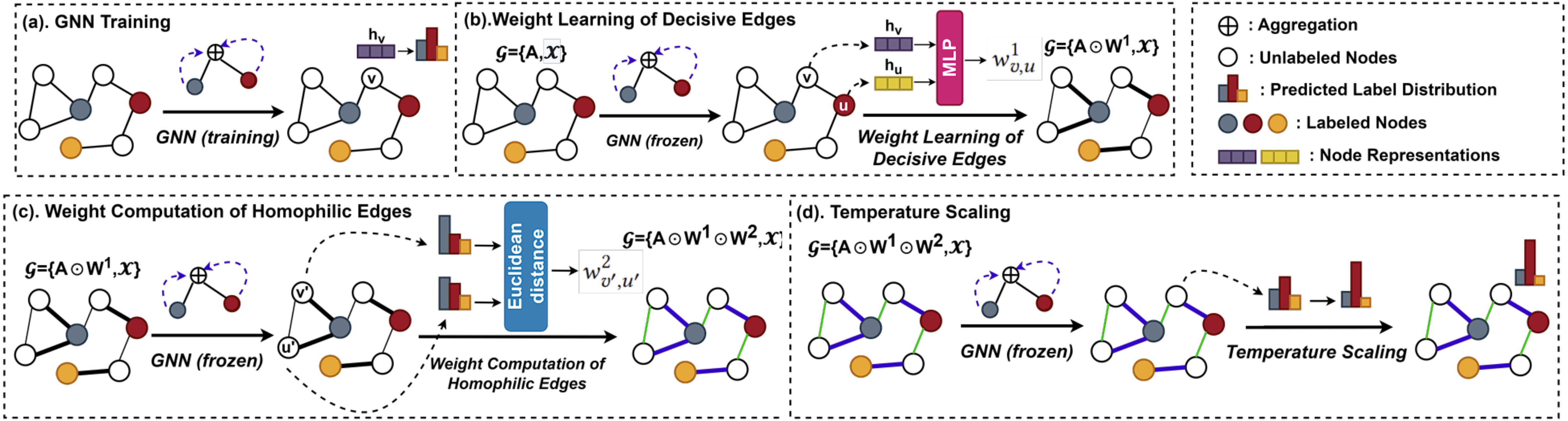
- To support the data-centric motivation, we further conduct data observations by analyzing the impacts of decisive and homophilic edges on calibration performance.

Table 1: Calibration performance with original/modified graphs on 8 datasets. Here Modified-D and Modified-H represent the modified graphs based on decisive and homophilic edges, respectively. Decisive/homophilic edges are assigned with larger weights than unimportant/heterophilic ones. ECE scores (%) are the lower the better.

Model	Structure	Cora	Citeseer	Pubmed	Photo	Computers	CoraFull	Arxiv	Reddit
GCN	Original	14.43±4.52	14.42±4.17	8.41±1.29	7.49±1.14	5.92±0.29	14.31±0.54	8.00±0.15	5.18±0.23
	Modified-D	14.01±3.54	13.97±3.24	7.06±1.20	4.29±0.56	4.35±0.18	12.84±0.41	7.10±0.13	3.45±0.19
	Modified-H	13.61±3.92	14.35±3.66	8.29±1.01	6.22±1.01	5.07±0.51	13.95±0.51	7.70±0.12	2.37±0.21
GraphSAGE	Original	10.25±5.27	10.82±4.74	7.43±2.23	8.27±2.60	7.22±0.78	13.92±1.21	8.79±1.52	9.67±0.31
	Modified-D	8.22±1.61	9.65±3.52	6.85±1.45	4.53±1.00	6.41±0.76	9.95±0.73	8.42±1.39	5.74±0.27
	Modified-H	4.22±1.86	5.80±1.08	4.00±0.78	2.00±1.00	2.93±0.95	4.17±1.14	2.02±1.12	4.93±0.24

DCGC

- Motivated by our observations, we propose Data-centric Graph Calibration (DCGC). Given a well-trained GNN, we design two modules to improve the weights of decisive and homophilic edges.



Experiments

We conducted experiments on 8 datasets with GCN and GraphSAGE.

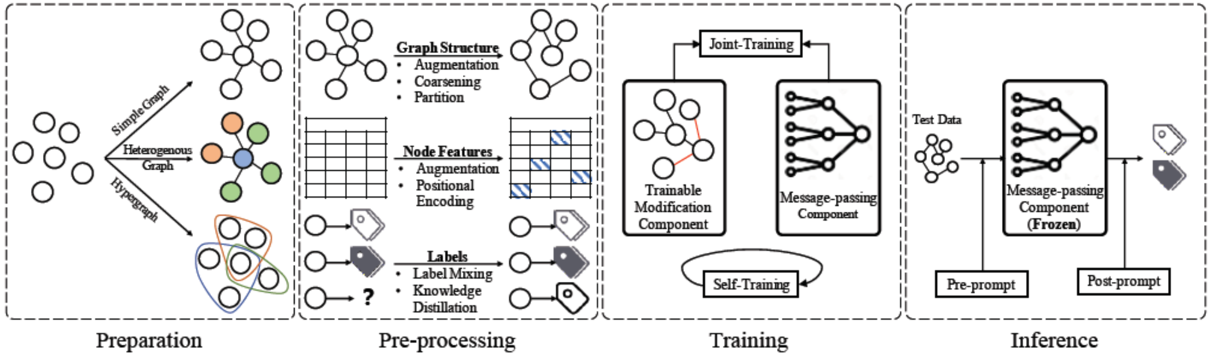
Model	Method	Cora	Citeseer	Pubmed	Photo	Computers	CoraFull	Arxiv	Reddit
GCN	Original	14.43±4.52	14.42±4.17	8.41±1.29	7.49±1.14	5.92±0.29	14.31±0.54	8.00±0.15	5.18±0.23
	TS	6.60±1.83	10.22±1.92	4.43±0.58	3.16±1.02	3.92±1.56	11.00±0.78	6.39±0.31	5.12±0.22
	DCGC+TS	4.89±1.41	8.13±2.36	2.18±0.71	1.72±0.62	1.93±0.50	5.63±0.78	4.26±0.37	4.17±0.32
	VS	8.26±1.80	10.86±1.38	5.02±0.68	4.54±0.96	4.46±1.31	13.68±0.37	7.68±0.21	4.36±0.05
	DCGC+VS	6.04±1.67	8.86±1.69	2.50±0.85	1.77±0.49	1.67±0.70	8.32±0.85	4.60±0.27	3.84±0.27
	CaGCN	6.88±1.29	8.41±1.87	3.52±0.56	1.75±0.72	2.94±3.33	7.09±0.58	3.87±0.39	2.92±0.14
	DCGC+CaGCN	5.42±1.25	6.68±1.85	1.68±0.54	1.11±0.24	2.55±2.84	4.52±0.47	2.86±0.37	1.23±0.26
	GATS	5.27±1.86	9.09±2.03	3.69±0.51	1.41±0.41	1.61±0.85	9.07±0.61	4.42±0.31	-
	DCGC+GATS	4.23±1.24	7.17±2.30	1.66±0.47	1.30±0.26	1.58±0.41	4.21±0.56	3.87±0.33	-
GraphSAGE	Original	10.25±5.27	10.82±4.74	7.43±2.23	8.27±2.60	7.22±0.78	13.92±1.21	8.79±1.52	9.67±0.31
	TS	9.68±3.83	9.42±1.68	5.15±0.80	2.76±0.79	2.85±0.69	10.54±1.33	7.77±0.99	9.05±0.20
	DCGC+TS	6.03±1.19	5.00±0.68	3.54±1.06	1.45±0.50	2.26±0.66	5.39±1.25	4.14±1.21	4.04±0.47
	VS	9.91±3.75	9.18±3.19	5.14±0.35	4.11±0.89	4.25±0.68	14.47±1.66	8.55±1.18	9.87±0.26
	DCGC+VS	5.14±0.72	5.91±0.76	2.19±0.63	1.62±0.71	2.14±0.55	8.28±1.63	5.10±1.36	8.16±0.36
	CaGCN	9.49±2.29	8.67±1.64	4.63±1.74	2.05±0.63	2.38±0.36	6.91±1.35	4.13±1.22	5.02±0.22
	DCGC+CaGCN	5.26±1.35	5.38±3.10	2.30±0.69	1.31±0.36	2.13±0.43	4.29±0.84	3.83±1.15	2.15±0.17
	GATS	9.68±3.38	8.86±2.05	5.04±1.33	2.44±0.77	2.76±0.58	8.69±1.27	5.96±1.21	-
	DCGC+GATS	6.99±1.61	6.18±1.73	3.70±1.25	1.43±0.40	2.31±0.67	4.50±0.99	2.92±1.16	-

Outline

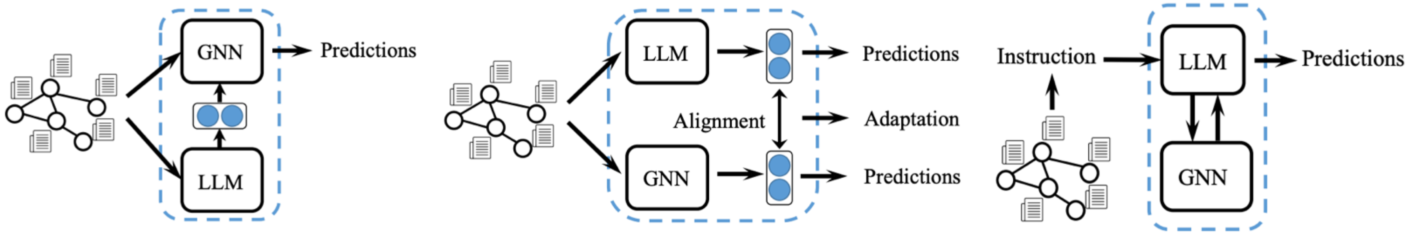
- Background
- Trustworthy GNNs
- Our Recent Attempts
- Future Directions

Future Directions

1. Data-centric Learning
   - Data quantity and quality
   - Structure/Feature/Label Augmentation



2. Integration with LLMs
   - World knowledge for trustworthiness
   - Graph foundation models



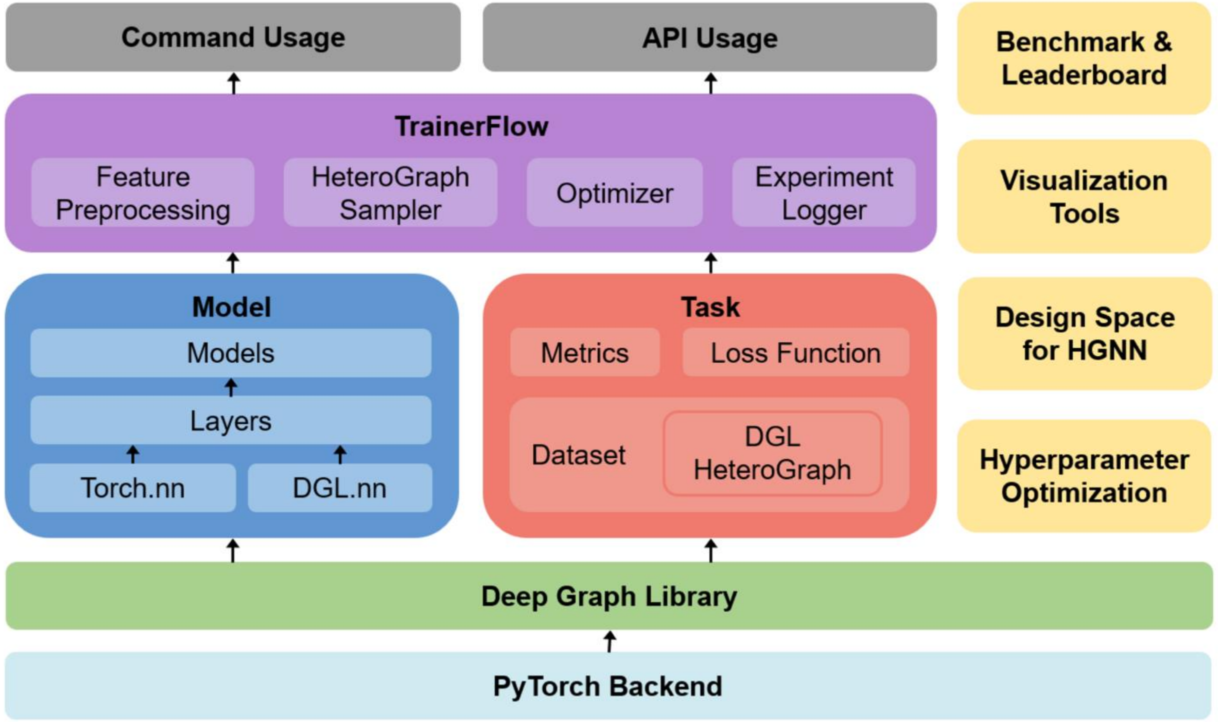
(a) GNN-centric methods.

(b) Symmetric methods, where the aligned embeddings can be further utilized for downstream tasks.

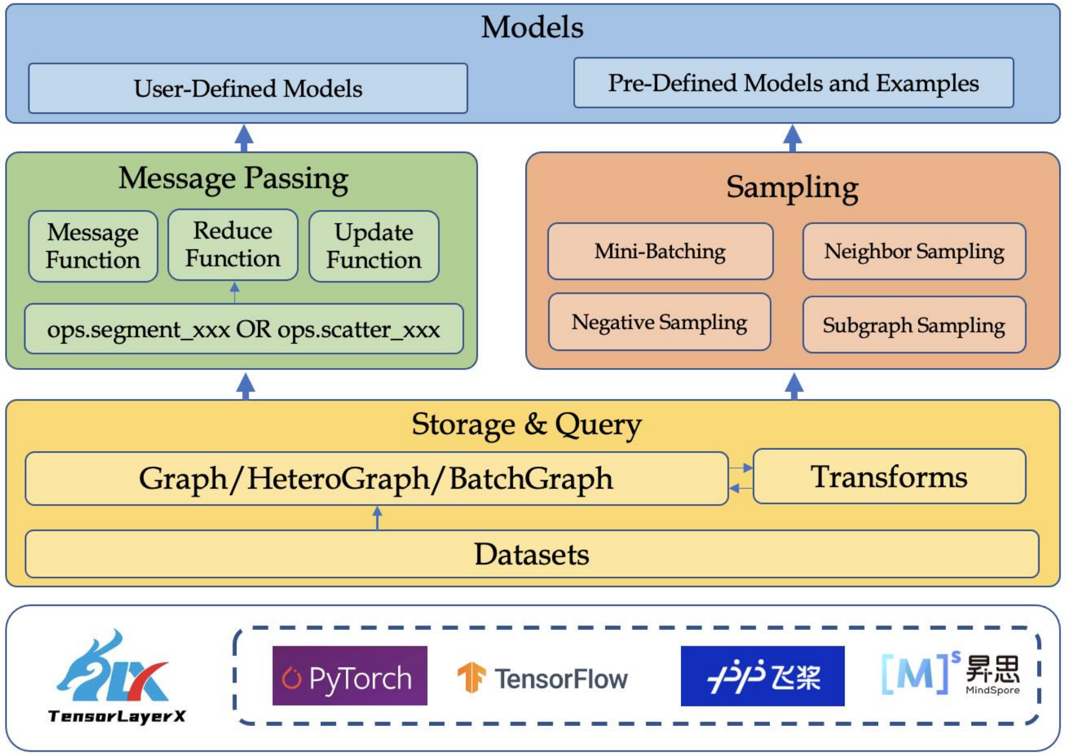
(c) LLM-centric methods, which take an instruction as input and output an answer.

Open-source Graph Learning Platforms

OpenHGNN: The first heterogeneous graph neural network library



GammaGL: A GNN library supporting multiple deep learning backends



Yaoqi Liu, Cheng Yang, Tianyu Zhao, Hui Han, Siyuan Zhang, Jing Wu, Guangyu Zhou, Hai Huang, Hui Wang, Chuan Shi. GammaGL: A Multi-Backend Library for Graph Neural Networks. SIGIR 2023

Han H, Zhao T, Yang C, et al. OpenHGNN: An Open Source Toolkit for Heterogeneous Graph Neural Network. CIKM 2022

图数据挖掘和机器学习

▶ 扫码关注我们 ◀

Thanks
Q&A