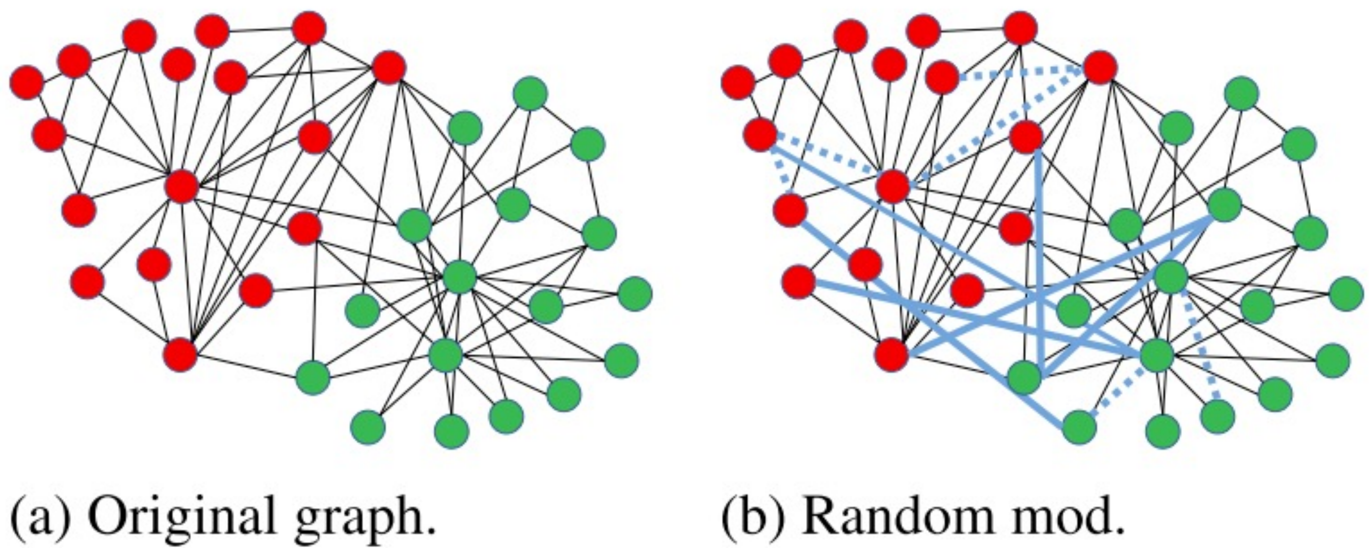
Learned Augmentation Approaches

Wei Jin, Michigan State University

Limitation of Rule-based Approaches

- Do not leverage task information and could hurt the downstream performance



F1 Score: 92.4

F1 Score: 91.0

Zhao et al. Data Augmentation for Graph Neural Networks. AAAI 2021

Learned Augmentation Approaches

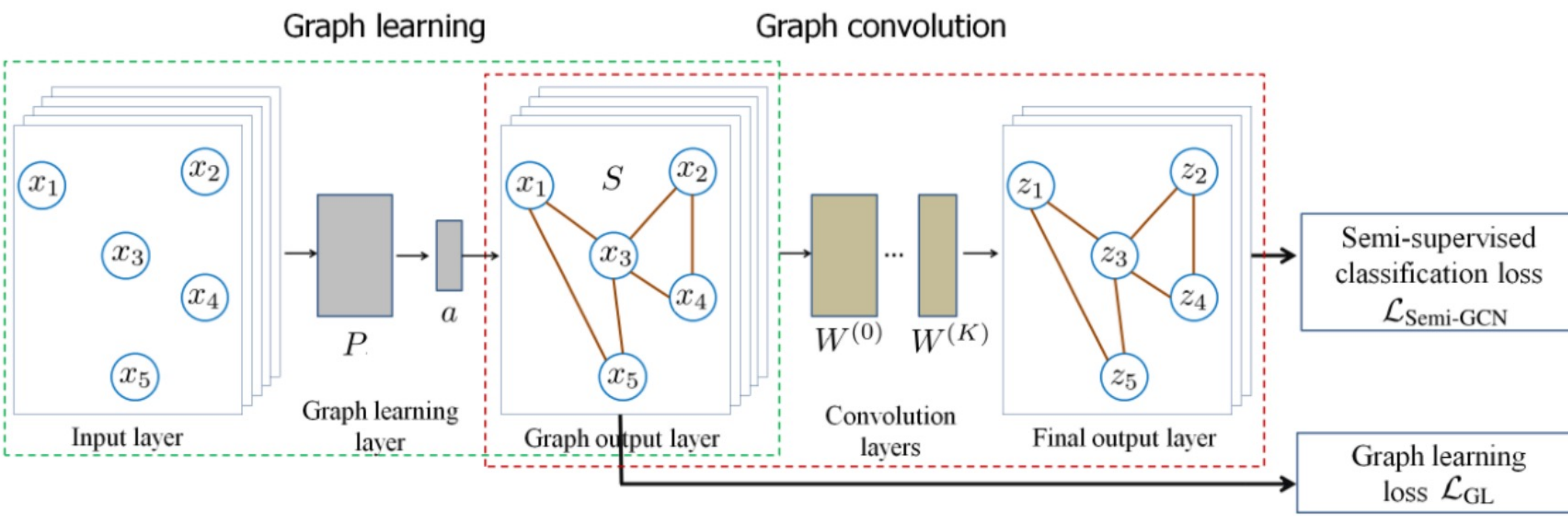
- Graph Structure Learning
  - Augment data with good graph structures
- Adversarial Training
  - Augment data with adversarial examples
- Rationalization
  - Augment data by changing graph environment
- Automated Augmentation
  - Automatically combine different augmentations

Graph Structure Learning

	Task Level			Augmented Data		
Attack Methods	**Node**	**Graph**	**Edge**	**Structure**	**Feature**	**Label**
GAug	✓			✓		
GLCN	✓			✓		
ProGNN	✓			✓		
GTrans	✓			✓	✓	

Graph Structure Learning: Framework

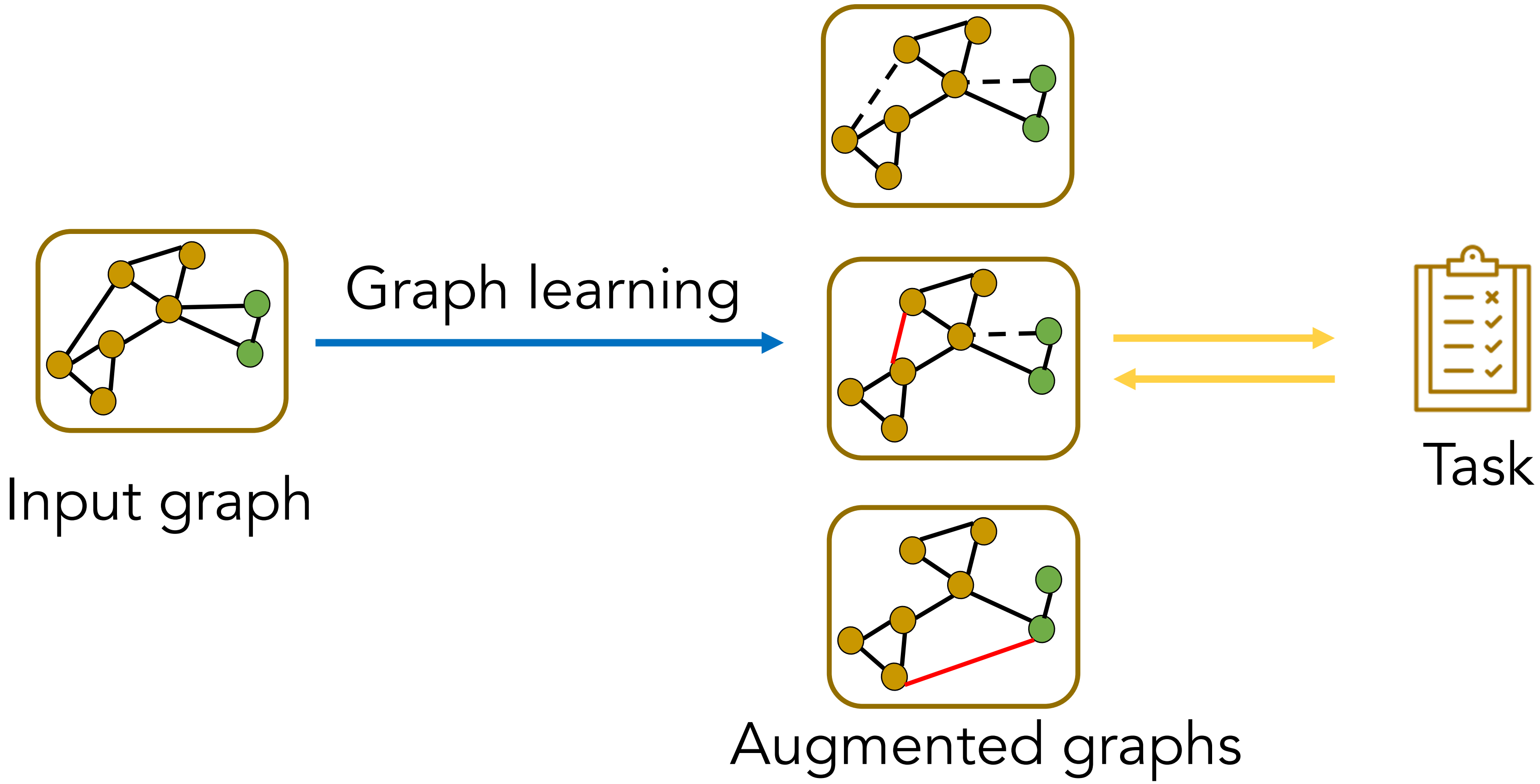
- Graph learning + graph convolution



Jiang et al. Semi-supervised Learning with Graph Learning-Convolutional Networks. CVPR 2019.

Graph Structure Learning: Core Component

- Graph learning component



GAug: Neural Edge Predictor

- What are better graph structures?
  - “Noisy” edges should be removed

    Inter-class edges

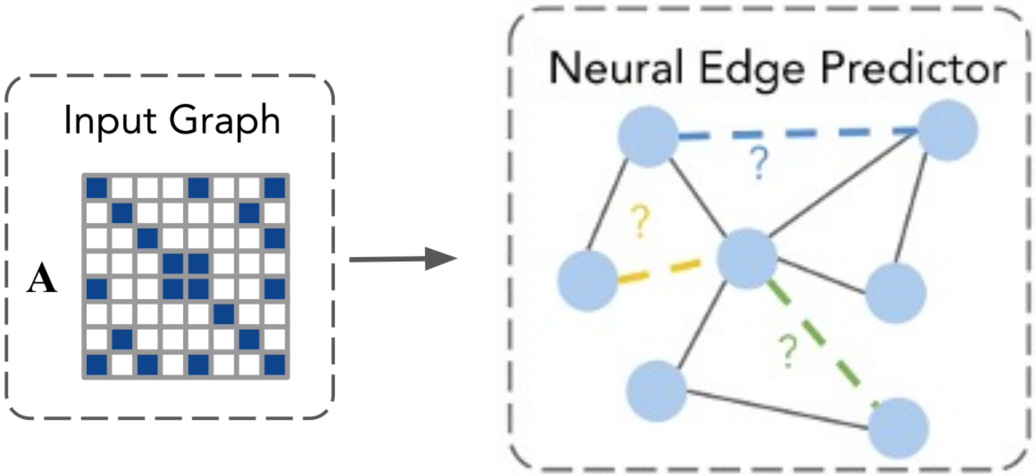
  - “Missing” edges should be added

    Intra-class edges

Zhao et al. Data Augmentation for Graph Neural Networks. AAAI 2021

GAug: Neural Edge Predictor

- GAug uses a neural edge predictor to promote intra-class edges and demote inter-class edges



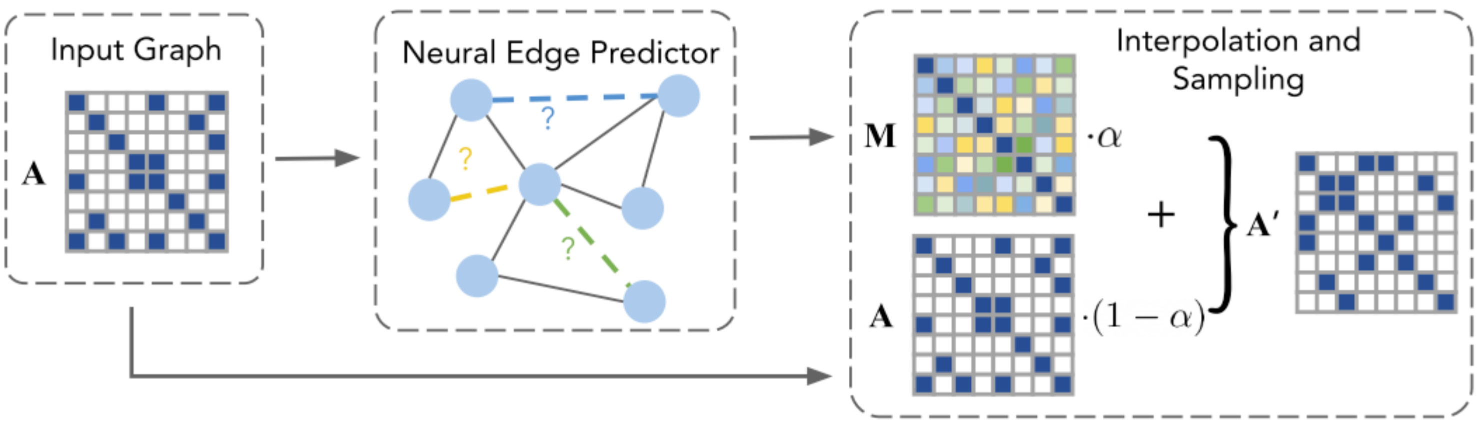
$$\underbrace{\mathbf{M} = \sigma\left(\mathbf{Z}\mathbf{Z}^T\right)}_{M \text{ models node similarities}}, \text{ where } \mathbf{Z} = f_{GCL}^{(1)}\left(\mathbf{A}, f_{GCL}^{(0)}\left(\mathbf{A}, \mathbf{X}\right)\right)$$

M models node similarities

Zhao et al. Data Augmentation for Graph Neural Networks. AAAI 2021

GAug: Framework

- Interpolation and sampling



$$\mathbf{P}_{ij} = \alpha\mathbf{M}_{ij} + (1-\alpha)\mathbf{A}_{ij}$$

Bernoulli sampling

$$\mathbf{A}'_{ij}$$

Zhao et al. Data Augmentation for Graph Neural Networks. AAAI 2021

GLCN: Graph Learning Layer

- Modeling the relationship between nodes as a function of their node features

$$S_{ij} = g(x_i, x_j) = \frac{\exp(\mathrm{\mathbf{ReLU}}(a^T|x_i - x_j|))}{\sum_{j=1}^{n} \exp(\mathrm{\mathbf{ReLU}}(a^T|x_i - x_j|))}$$

$$S_{ij} = g(x_i, x_j) = \frac{A_{ij}\exp(\mathrm{\mathbf{ReLU}}(a^T|x_i - x_j|))}{\sum_{j=1}^{n} A_{ij}\exp(\mathrm{\mathbf{ReLU}}(a^T|x_i - x_j|))}$$

Matrix Estimation

$$\mathcal{L}_{\mathrm{GL}} = \sum_{i,j=1}^{n} \|x_i - x_j\|_2^2 S_{ij} + \gamma\|S\|_F^2 + \beta\|S - A\|_F^2$$

Jiang et al. Semi-supervised Learning with Graph Learning-Convolutional Networks. CVPR 2019.

GLCN: Graph Learning Layer

- Modeling the relationship between nodes as a function of their node features

$$S_{ij} = g(x_i, x_j) = \frac{\exp(\mathbf{ReLU}(a^T|x_i - x_j|))}{\sum_{j=1}^{n} \exp(\mathbf{ReLU}(a^T|x_i - x_j|))}$$

$$S_{ij} = g(x_i, x_j) = \frac{A_{ij}\exp(\mathbf{ReLU}(a^T|x_i - x_j|))}{\sum_{j=1}^{n} A_{ij}\exp(\mathbf{ReLU}(a^T|x_i - x_j|))}$$

Original Graph

Matrix Estimation

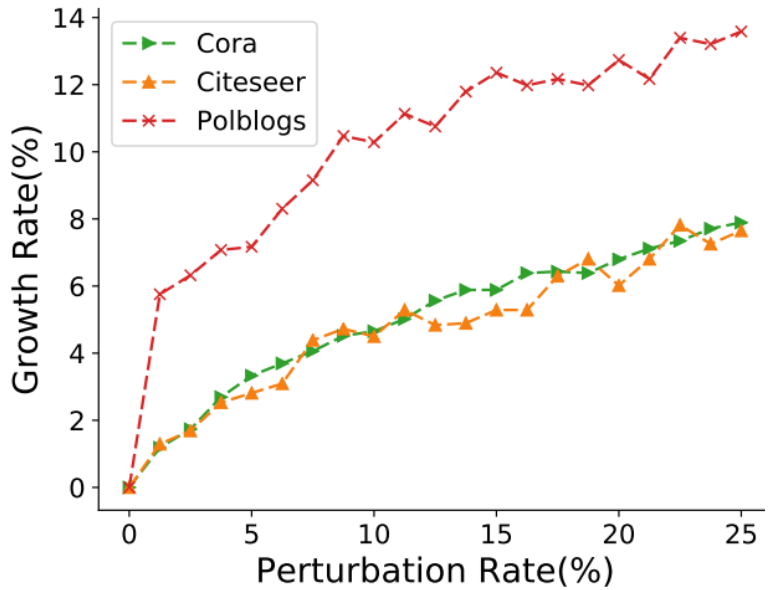
$$\mathcal{L}_{\mathrm{GL}} = \sum_{i,j=1}^{n} \underbrace{\|x_i - x_j\|_2^2 S_{ij}}_{\text{Connected nodes should be similar}} + \gamma\|S\|_F^2 + \beta\|S - A\|_F^2$$

New graph

Jiang et al. Semi-supervised Learning with Graph Learning-Convolutional Networks. CVPR 2019.

ProGNN: What Are Better Graph Structures?

- A closer look at graphs perturbed by adversarial attacks



Low rank

Dataset	r(%)	edge+	edge-	edges	ranks	clustering coefficients
Cora	0	0	0	5069	2192	0.2376
	5	226	27	5268	2263	0.2228
	10	408	98	5380	2278	0.2132
	15	604	156	5518	2300	0.2071
	20	788	245	5633	2305	0.1983
	25	981	287	5763	2321	0.1943
Citeseer	0	0	0	3668	1778	0.1711
	5	181	2	3847	1850	0.1616
	1	341	25	3985	1874	0.1565
	15	485	65	4089	1890	0.1523
	20	614	119	4164	1902	0.1483
	25	743	174	4236	1888	0.1467
Polblogs	0	0	0	16714	1060	0.3203
	5	732	103	17343	1133	0.2719
	10	1347	324	17737	1170	0.2825
	15	1915	592	18038	1193	0.2851
	20	2304	1038	17980	1193	0.2877
	25	2500	1678	17536	1197	0.2723

Sparse

Jin et al. Graph Structure Learning for Robust Graph Neural Networks. KDD 2020.

ProGNN: Graph Learning Component

- Model the new graph structure as free parameters

Original Graph

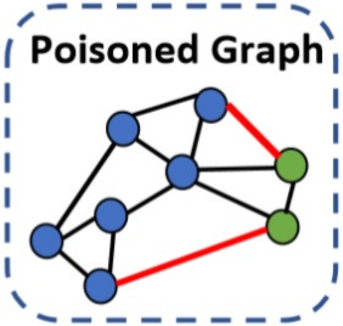
$$\arg\min_{\mathbf{S}\in\mathcal{S}} \mathcal{L}_0 = ||\mathbf{A} - \mathbf{S}||_F^2 + \underbrace{\alpha||\mathbf{S}||_1}_{\text{Sparse}} + \underbrace{\beta||\mathbf{S}||_*}_{\text{Low rank}}, \ s.t., \mathbf{S} = \mathbf{S}^\top$$

New graph

$$||\mathbf{S}||_1 = \Sigma_{ij}|\mathbf{S}_{ij}| \quad ||\mathbf{S}||_* = \Sigma_{i=1}^{rank(\mathbf{S})}\sigma_i$$

Jin et al. Graph Structure Learning for Robust Graph Neural Networks. KDD 2020.

ProGNN: Framework



Jin et al. Graph Structure Learning for Robust Graph Neural Networks. KDD 2020.

GTrans: Test-time Graph Augmentation

Why test-time augmentation?

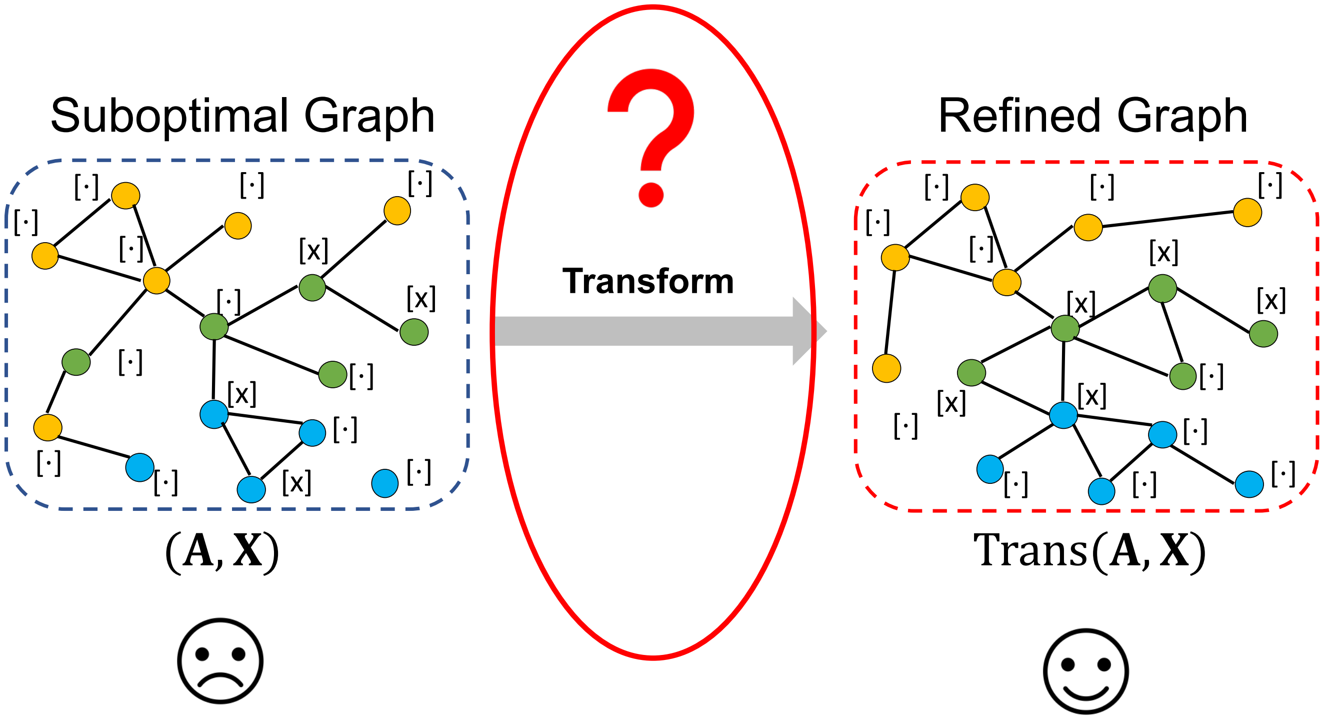
When the models are too large and we do not want to modify the model architecture/parameters, how do we augment the graph data to improve the performance?

Transform the graph data at test time while keeping these models unchanged.

Jin et al., "Empowering Graph Neural Networks with Test-Time Graph Transformation". ICLR 2023

GTrans: Test-time Graph Augmentation

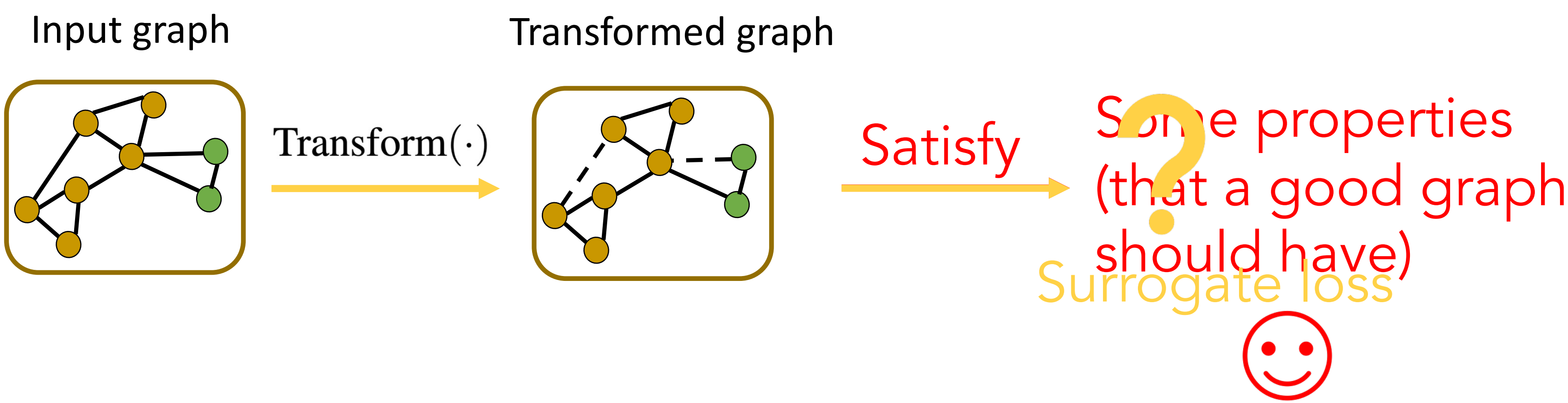
Goal: Transform the test graph to improve trained GNN models *while keeping these models unchanged*.



Jin et al., "Empowering Graph Neural Networks with Test-Time Graph Transformation". ICLR 2023

GTrans: What Objective To Use?

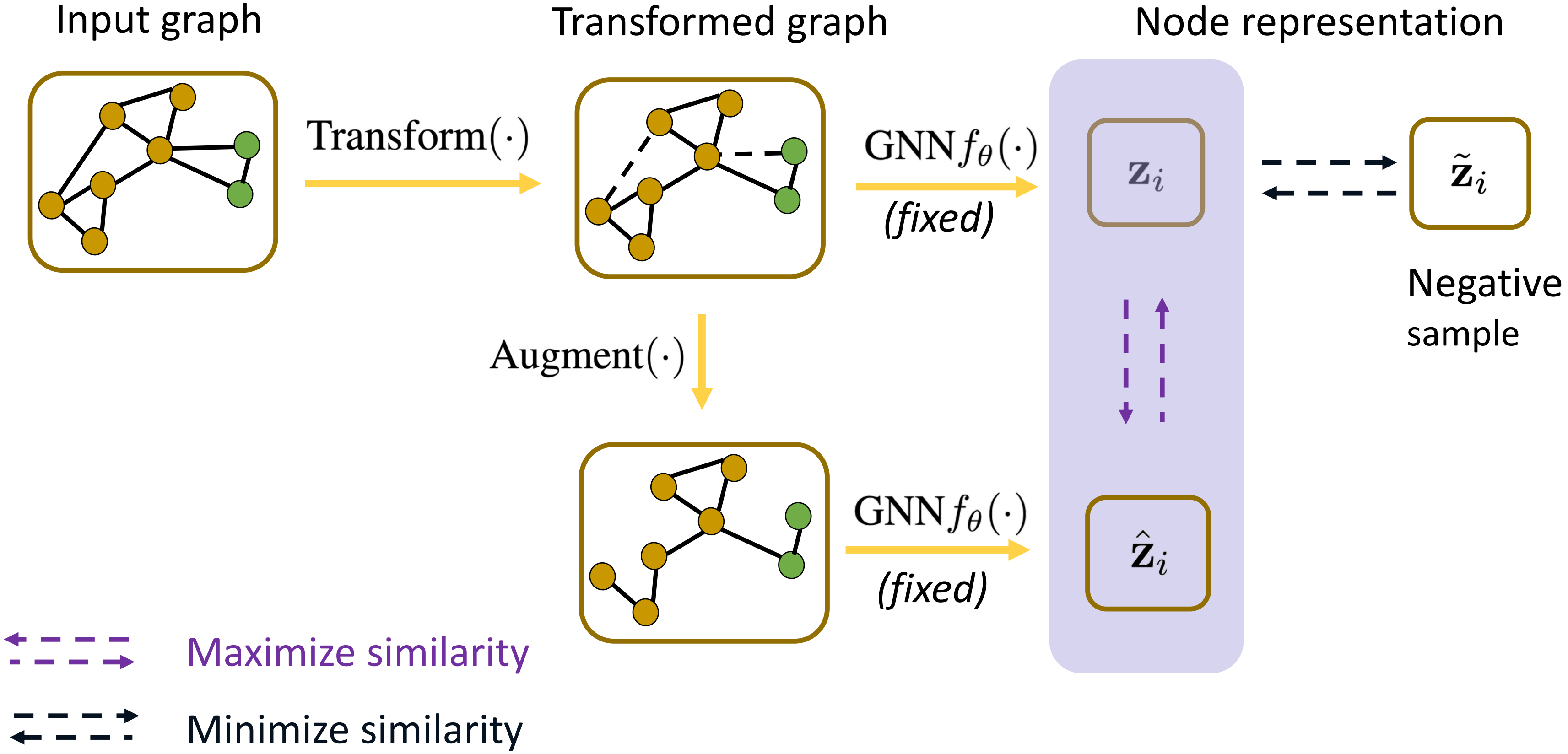
We do not have the label information for test data ☹



GTrans: Designing A Surrogate Loss

A good graph should satisfy some property:

data augmentation on the nodes should create similar node samples



Graph Adversarial Training

	Task Level			Augmented Data		
Attack Methods	**Node**	**Graph**	**Edge**	**Structure**	**Feature**	**Label**
RobustTraining	✓			✓		
FLAG	✓	✓	✓		✓	
LAT	✓			✓	✓	

Graph Adversarial Training

Motivation

Augment the training set with adversarial data to improve model robustness

Main Idea

$$\min_{\theta} \max_{\substack{\Delta_A \in \mathcal{P}_A \\ \Delta_X \in \mathcal{P}_X}} \mathcal{L}\left(f_\theta(A + \Delta_A, X + \Delta_X), Y\right)$$

RobustTraining: Structure Augmentation

- Projected gradient descent (PGD)

$$\min_{\theta} \max_{\substack{(\Delta_A)_{ij} \in \{0,1\} \\ \Sigma_{ij}\Delta_A \leq B}} \mathcal{L}(f_\theta(A + \Delta_A, X), Y)$$

$$\Delta_{\mathbf{A}} \leftarrow \bigcirc(\Delta_{\mathbf{A}} - \eta \nabla_{\Delta_{\mathbf{A}}} \mathcal{L}(\Delta_{\mathbf{A}}))$$ Projected Gradient descent

$$(\Delta_{\mathbf{A}})_{ij} \in [0, 1],\ \sum_{i,j} \Delta_{\mathbf{A}_{ij}} \leq B$$

Xu et al. Topology Attack and Defense for Graph Neural Networks: An Optimization Perspective. IJCAI 2019.

FLAG: Feature Augmentation

- Consider the problem of augmenting features

$$\min_{\theta} \max_{||\delta||_\infty < \epsilon} \mathcal{L}(f_\theta(x+\delta), y)$$

- The final perturbation is obtained by iterative updates:

$$\boldsymbol{\delta}_{t+1} = \Pi_{||\boldsymbol{\delta}||_\infty \leq \epsilon} \left(\boldsymbol{\delta}_t + \alpha \cdot \operatorname{sign} \left(\nabla_{\boldsymbol{\delta}} L \left(f_{\boldsymbol{\theta}}(x + \boldsymbol{\delta}_t), y \right) \right) \right)$$

Kong et al. Robust Optimization as Data Augmentation for Large-scale Graphs. CVPR 2022.

FLAG: Feature Augmentation

- Diverse types of data augmentations are beneficial!

We should take advantage of the intermediate perturbations $(\delta_1, \delta_2, \ldots, \delta_M)$ to form diverse augmentation

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \frac{\tau}{M} \sum_{t=1}^{M} \nabla_{\boldsymbol{\theta}} L\left(f_{\boldsymbol{\theta}}(x + \boldsymbol{\delta_t}), y\right)$$

Kong et al. Robust Optimization as Data Augmentation for Large-scale Graphs. CVPR 2022.

Latent Adversarial Training

Obstacles

- A is discrete
- X may also be discrete

Latent Adversarial Training

- Apply it on the hidden layer !
- $\min_{\theta} \max_{\Delta \in \mathcal{P}} f_{\theta}(H^{(1)} + \Delta)$

Jin et al. Latent Adversarial Training of Graph Convolution Networks. ICML 2019 workshop.

Graph Rationalization

	Task Level			Augmented Data		
Attack Methods	**Node**	**Graph**	**Edge**	**Structure**	**Feature**	**Label**
GREA		✓		✓	✓	
DIR		✓		✓	✓	

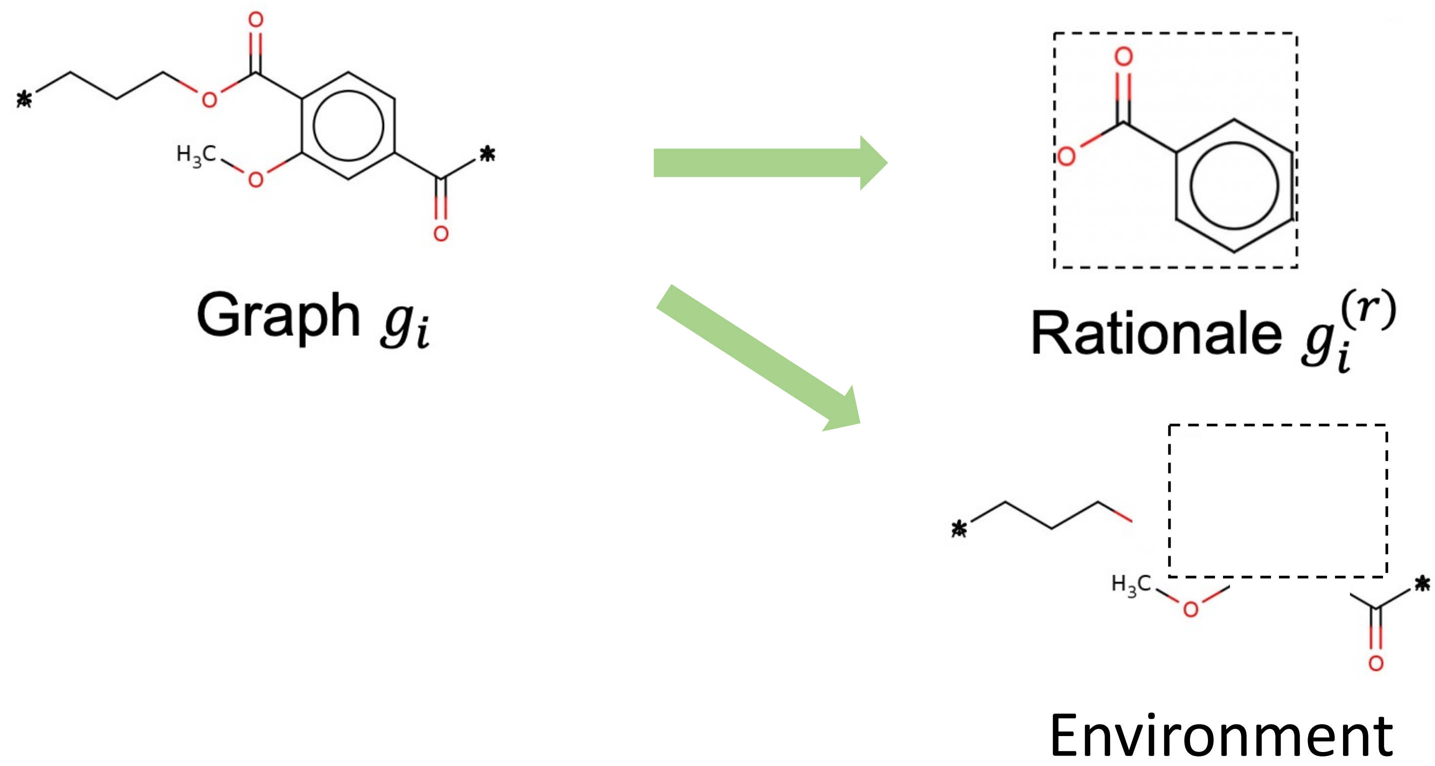
What's Graph Rationalization

- A *rationale* is defined as a subset of input features that best represent, guide and support model prediction

- In the graph domain, rationales are usually intrinsically learned subgraphs that are *representative, provide information or explanation* to the graph models.

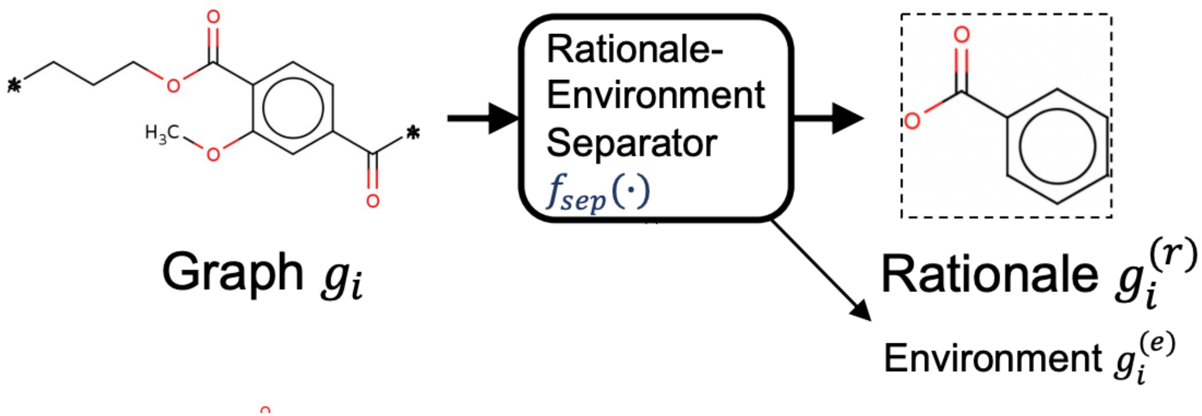
What's Graph Rationalization

- A *rationale* is defined as a subset of input features that best represent, guide and support model prediction



GREA: Framework (I)

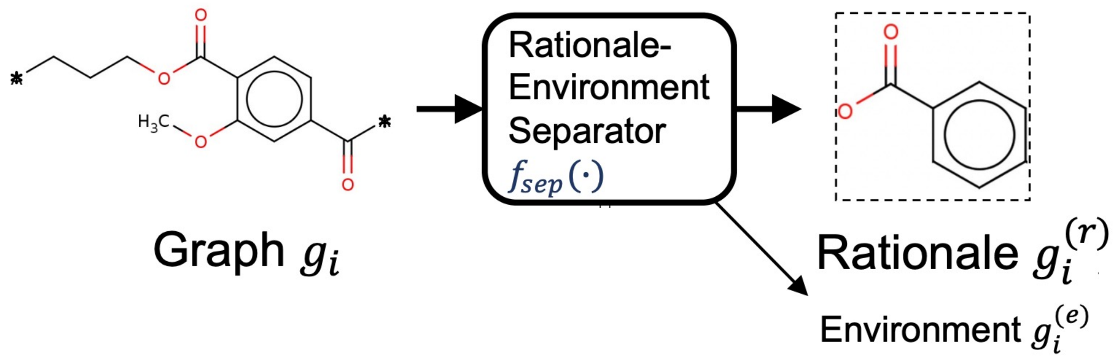
- Environment removal



Liu et al. Graph Rationalization with Environment-based Augmentations. KDD 2022.

GREA: Framework (II)

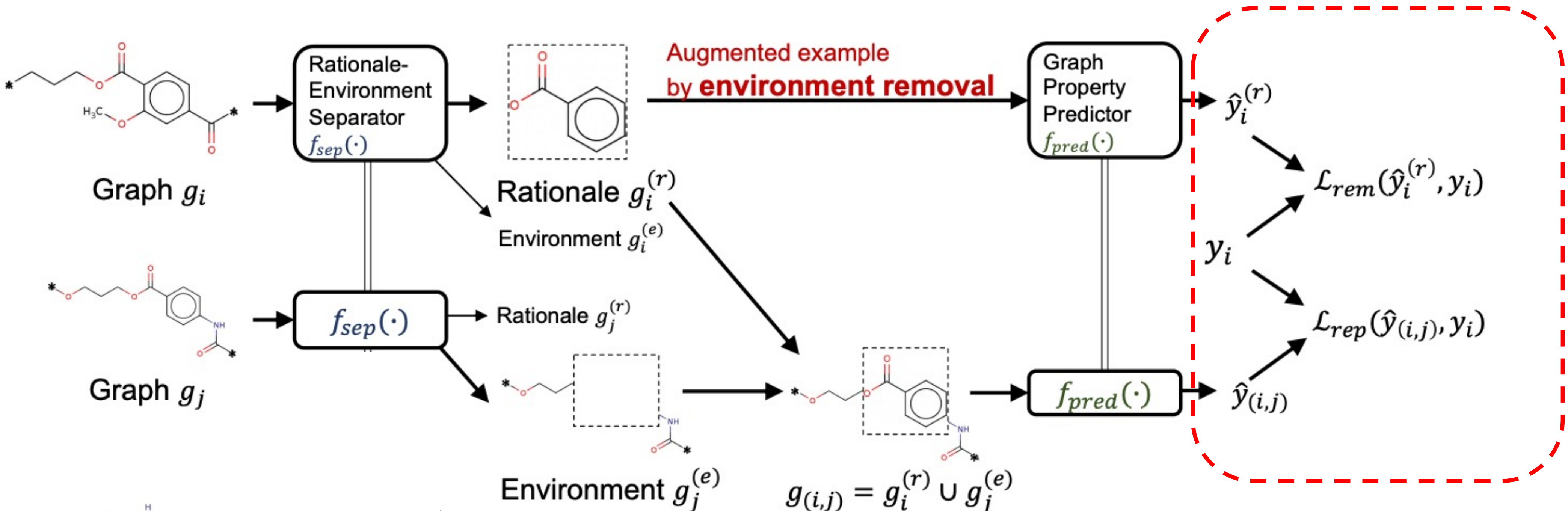
- Environment replacement



Liu et al. Graph Rationalization with Environment-based Augmentations. KDD 2022.

GREA: Framework (II)

- Environment replacement



Liu et al. Graph Rationalization with Environment-based Augmentations. KDD 2022.

GREA: Framework Details

How to model the environment separator?

- Learning a mask *m* that indicates the probability of nodes being classified into the rationale subgraph

$$\mathbf{m} = \sigma(\mathrm{MLP}_1(\mathrm{GNN}_1(g)))$$

- Then with node representation matrix $\mathbf{H} = \mathrm{GNN}_2(g)$

$$\mathbf{h}^{(r)} = \mathbf{1}_N^\top \cdot (\mathbf{m} \times \mathbf{H}) \quad \mathbf{h}^{(e)} = \mathbf{1}_N^\top \cdot ((\mathbf{1}_N - \mathbf{m}) \times \mathbf{H})$$

Liu et al. Graph Rationalization with Environment-based Augmentations. KDD 2022.

GREA: Framework (IV)

- Environment Removal Augmentation

$$\hat{y}_i^{(r)} = \mathrm{MLP}_2\left(\mathbf{h}_i^{(r)}\right)$$

- Environment Replacement Augmentation

$$\mathbf{h}_{(i,j)} = \mathrm{AGG}\left(\mathbf{h}_i^{(r)}, \mathbf{h}_j^{(e)}\right) = \mathbf{h}_i^{(r)} + \mathbf{h}_j^{(e)}$$

$$\hat{y}_{(i,j)} = \mathrm{MLP}_2\left(\mathbf{h}_{(i,j)}\right)$$

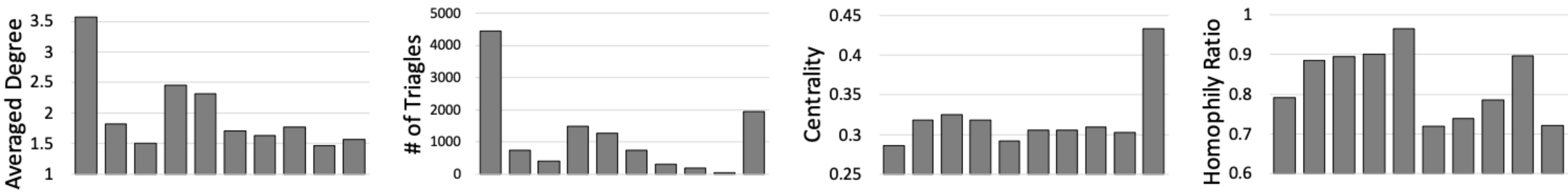
Liu et al. Graph Rationalization with Environment-based Augmentations. KDD 2022.

Automated Augmentation

	Task Level			Augmented Data		
Attack Methods	**Node**	**Graph**	**Edge**	**Structure**	**Feature**	**Label**
AutoGDA	✓			✓	✓	
GraphAug		✓		✓	✓	
JOAO		✓		✓	✓	

AutoGDA: Background

- Task: node classification
- Motivation: different communities in a graph have different characteristics



Data augmentation should be localized (community-specific)!

(a) Averaged Degree **(b)** # of Triangles **(c)** Centrality **(d)** Homophily Ratio

Figure 1: Graph communities detected by the Louvain method on the PubMed dataset show diverse distribution on different characteristics of graph structure.

Zhao et al. AutoGDA: Automated Graph Data Augmentation for Node Classification. LoG 2022.

AutoGDA: Method

- Goal: find a set of graph data augmentations for each community in the graph to maximize the performance

Outer level: augmentation policy optimization

$$\theta^* = \arg\max_{\theta} ACC\left(f_{\omega_\theta}\left(\bigcup_{k=1}^{N_c} aug\left(g_\theta(G_{Ck})\right)\right), \mathbf{y}_{val}\right),$$

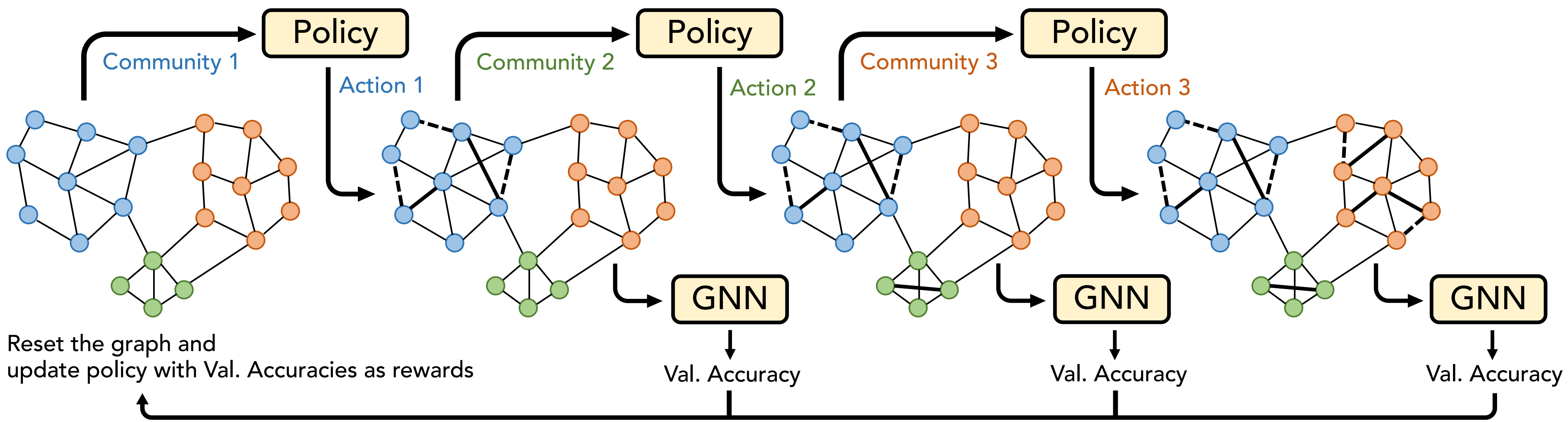
where $\omega_\theta = \arg\min_{\omega} \mathcal{L}\left(f_\omega\left(\bigcup_{k=1}^{N_c} aug\left(g_\theta(G_{Ck})\right)\right), \mathbf{y}_{tr}\right)$

Inner level: model weight optimization

Zhao et al. AutoGDA: Automated Graph Data Augmentation for Node Classification. LoG 2022.

AutoGDA: Framework

- Sequentially pick up the augmentation strategy



Zhao et al. AutoGDA: Automated Graph Data Augmentation for Node Classification. LoG 2022.

GraphAug: Motivation

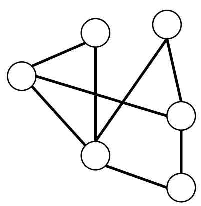
- Task: graph classification
- Motivation: ensuring label-invariance is hard for graphs

  Minor modification of a graph may change its semantics and thus labels

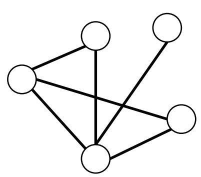
- GraphAug aims to produce augmented graphs that do not change the graph labels

Luo et al. Automated Data Augmentations for Graph Classification. ICLR 2023.

GraphAug: Reward Generation Model



G



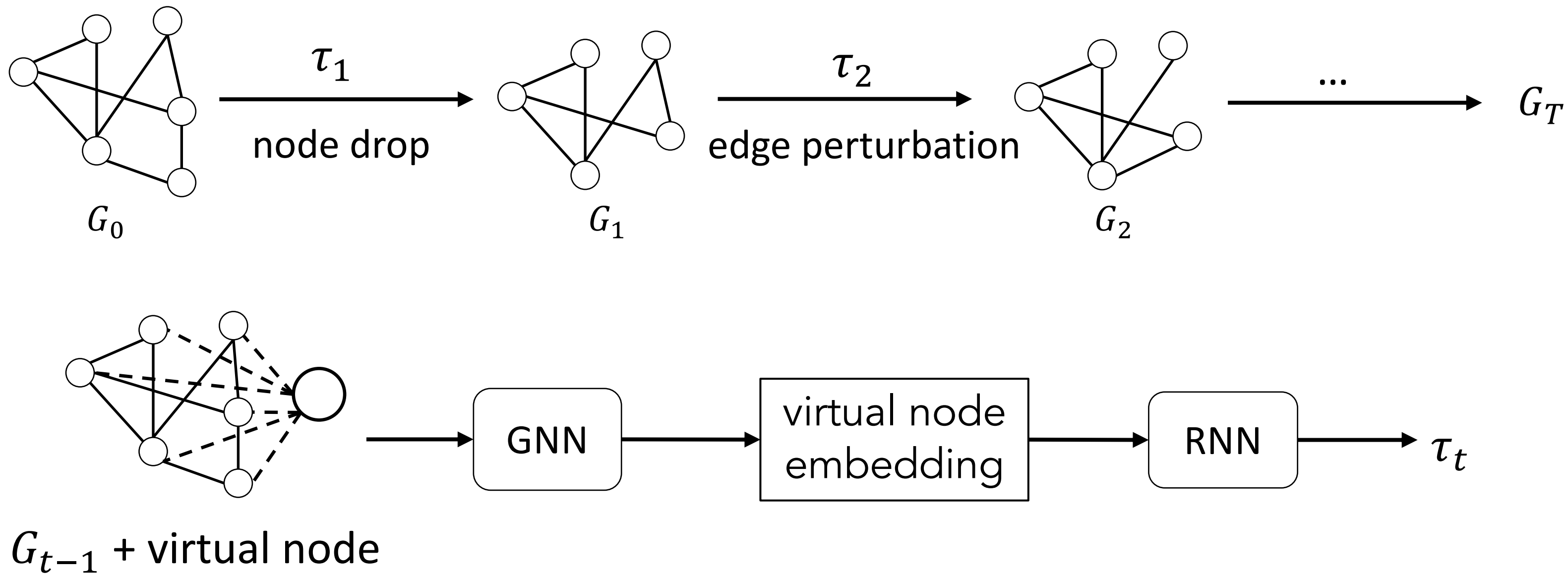
G_{aug}

Reward Generation Model

The probability that G and G_{aug} have the same label

Luo et al. Automated Data Augmentations for Graph Classification. ICLR 2023.

GraphAug: Augmentation Model



- ➢ Add a virtual node to the input graph, use a GNN based encoder to extract graph-level information.
- ➢ Use an RNN model, which takes the virtual node embedding as the input and outputs the augmentation probability distribution τ_t.

Luo et al. Automated Data Augmentations for Graph Classification. ICLR 2023.

Summary: Learned Augmentation Approaches

- Graph Structure Learning
- Adversarial Training
- Rationalization
- Automated Augmentation