A review of GNN explanation methods

Presenter: Zhanke Zhou

2022. 10. 28

Outline

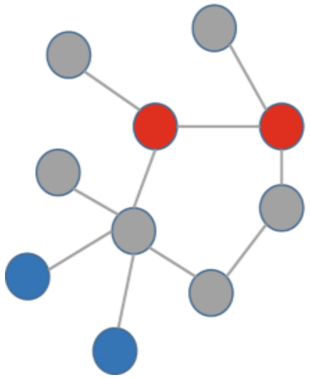
- Background
- A review of existing methods
- Recent advances that go beyond the post-hoc manner
- Summary

Outline

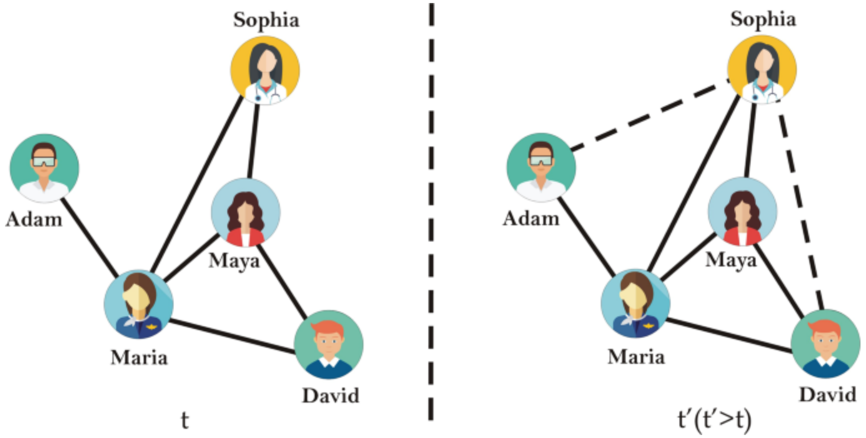
- Background
  - from graph learning to explainable graph learning
- A review of existing methods
- Recent advances that go beyond the post-hoc manner
- Summary

Background | graph learning

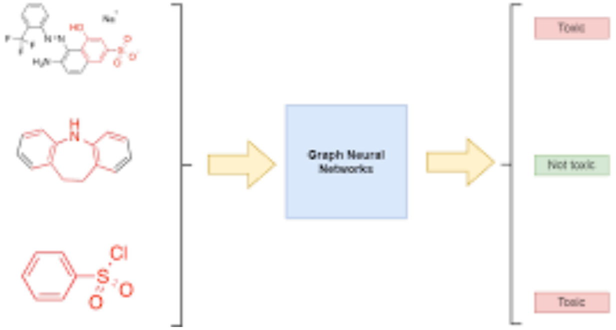
node-level



link-level

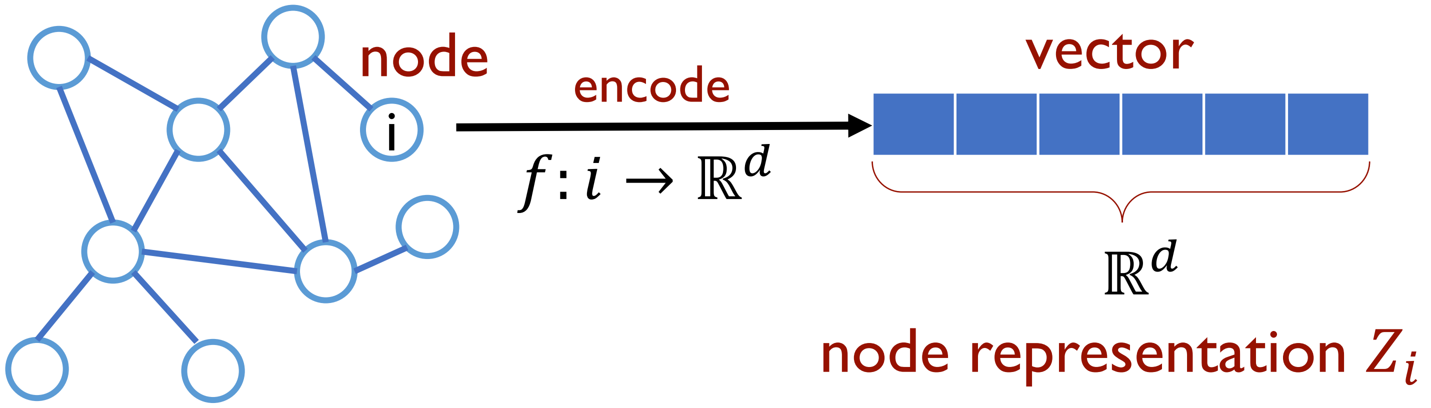


graph-level



Background | graph learning

Graph data $D = (A, X) \rightarrow$ GNN $f \rightarrow$ representation $Z \rightarrow \tilde{Y} \leftrightarrow Y$



However, only powerful is not enough
explainability is also important

Background | explainable graph learning

However, the mere predictive power of the graph classifier is of limited interest to the neuroscientists, which have plenty of tools for the diagnosis of specific mental disorders. What matters is the interpretation of the model, as it can provide novel insights and new hypotheses. [1]

[1]: Counterfactual Graphs for Explainable Classification of Brain Networks. KDD 2021

Background | explainable graph learning

Graph data $D = (A, X) \rightarrow$ GNN $f \rightarrow$ prediction $\tilde{Y} \leftrightarrow$ ground truth Y

Powerful

i.e., to approximate Y by $\tilde{Y}$

the learned representation and graph data are usually highly entangled

explainable

i.e., to determine which parts in D contribute to $\tilde{Y}$

an important property to trustworthy ML

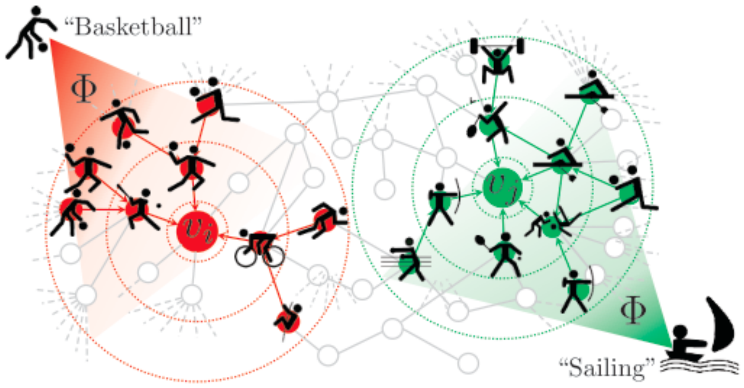
e.g. identifying the functional groups in a molecule

Core problem: how to provide ***better*** explanations?

Background | explainable graph learning

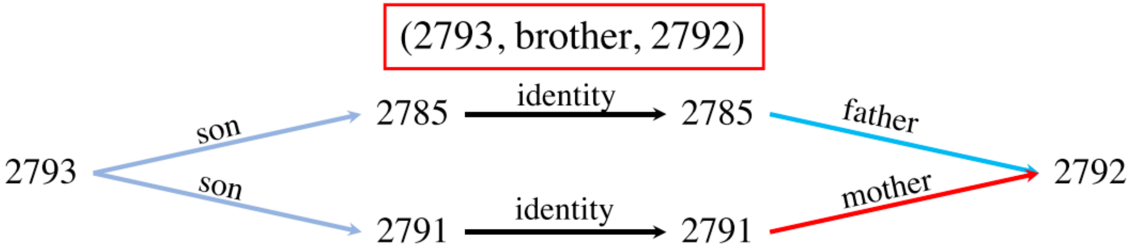
node-level task: requires relevant nodes

- e.g., node classification



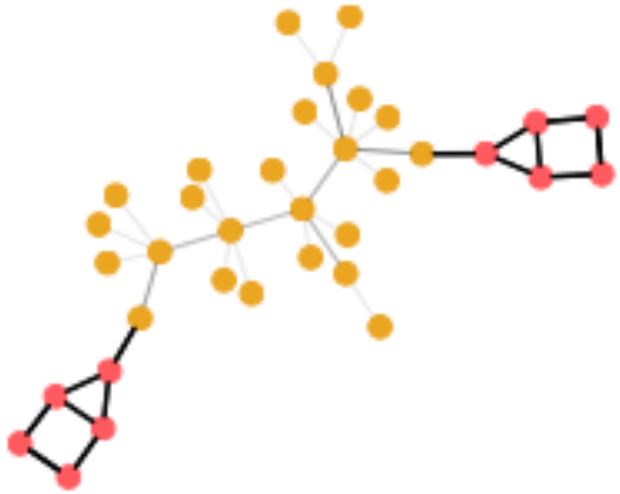
link-level task: requires relevant paths

- e.g., link prediction



graph-level task: requires relevant subgraphs

- e.g., graph classification



To interpret the prediction of a GNN, i.e., to identify a subgraph that contributes most to the prediction.

Background | challenges

- Discrete nature of graph structure
  - hard to optimize in a differentiable way
  - nodes and edges in a graph cannot be resized to the same shape
- Lack of fine-grind annotations
  - e.g., node-level / motif-level annotations are blank for graph-level tasks
  - we need a suitable objective to train and metric to evaluate the explanation method
- Lack of domain knowledge
  - e.g., molecules, social networks, and citation networks
  - graph data are less intuitive than images or texts

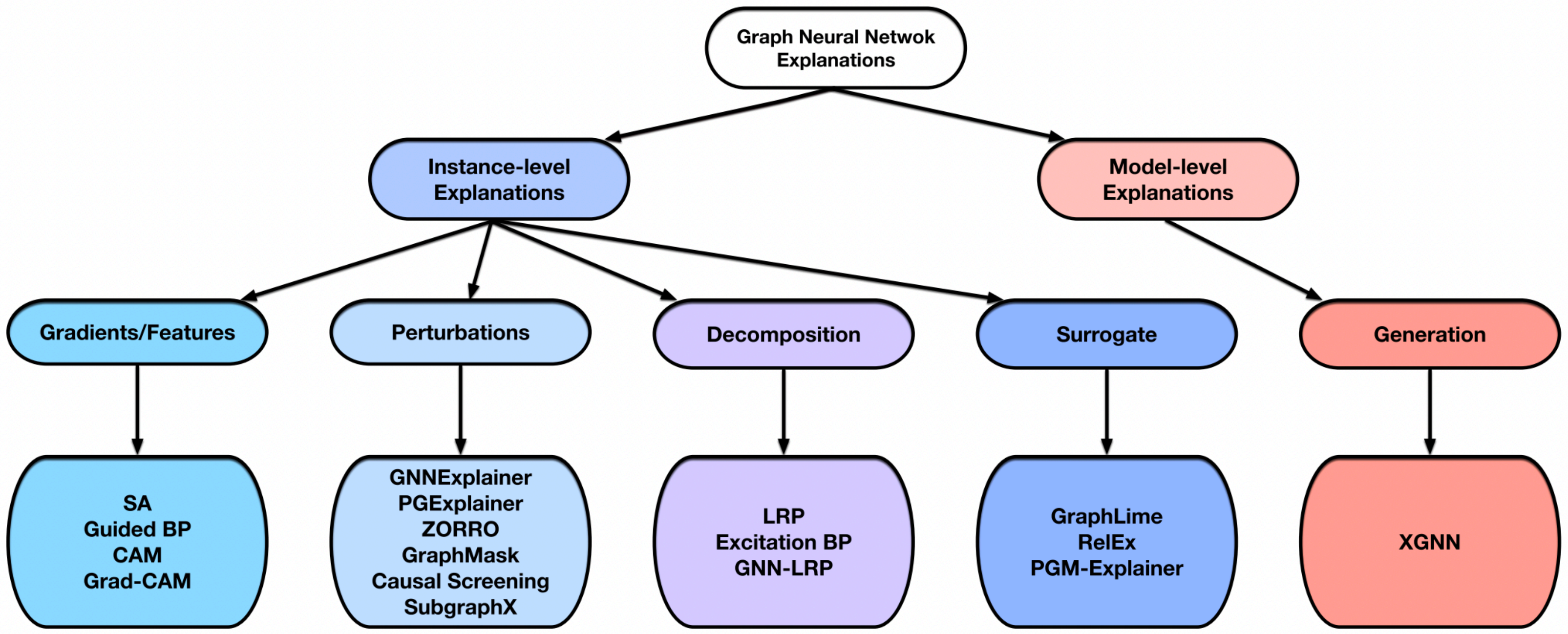
Outline

- Background
- A review of existing methods
  - Taxonomy
  - Metrics and evaluation
- Recent advances that go beyond the post-hoc manner
- Summary

Taxonomy

Instance-level Explanations

- Gradients/Features
- Perturbations *(we will focus on)*
  - GNNExplainer
  - PGExplainer
  - SubgraphX
- Decomposition
- Surrogate



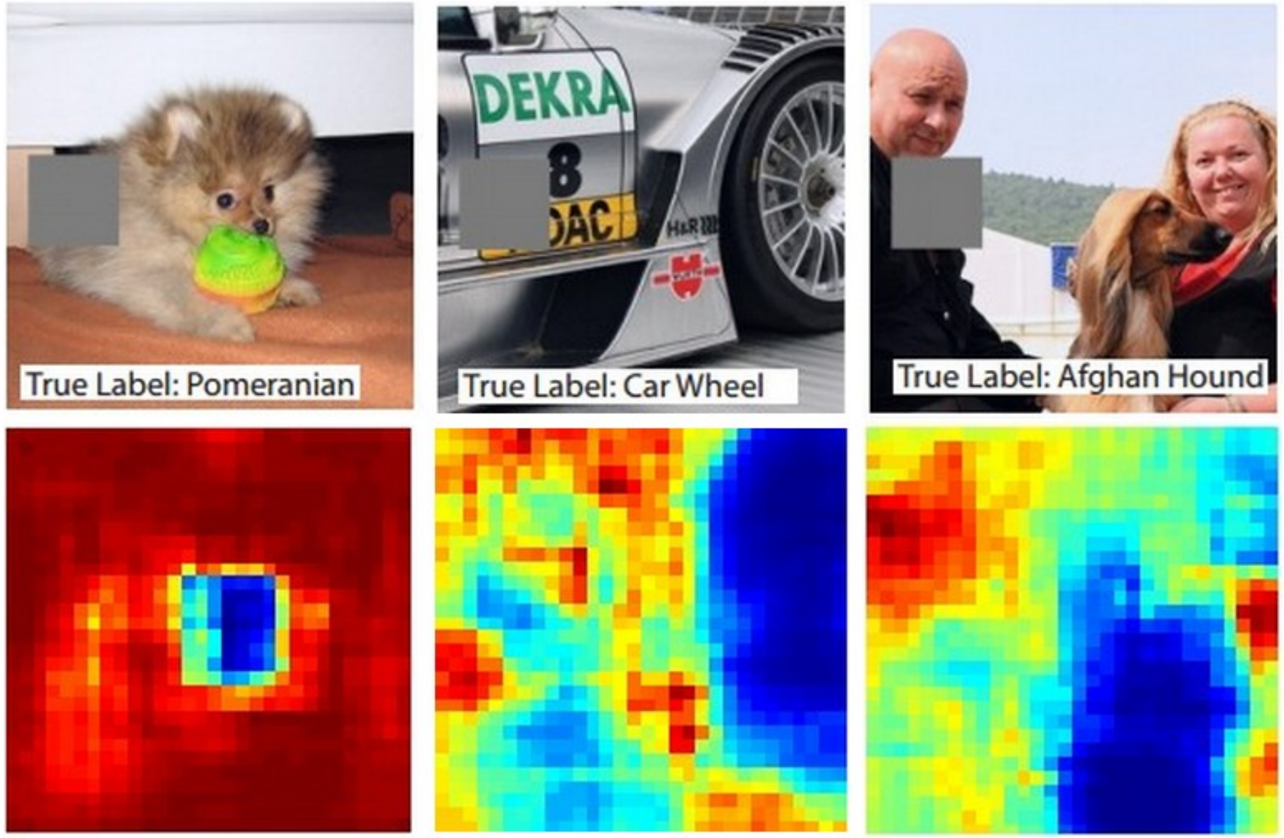
Model-level Explanations

XGNN (the only one)

Explainability in Graph Neural Networks: A Taxonomic Survey. TPAMI 2022.

Class1: Gradients/Features-Based Methods

use the gradients or feature map values as the approximations of input importance



Visualizing and Understanding Convolutional Networks. 2013
https://cs231n.github.io/understanding-cnn/

Class1: Gradients/Features-Based Methods

- [key idea] use the gradients/features to approximate input importance
  - [option1] get gradients of target prediction w.r.t. input by back-propagation
  - [option2] map the hidden features to the input space via interpolation

- generally, larger gradients or feature values indicate higher importance

4 representative methods

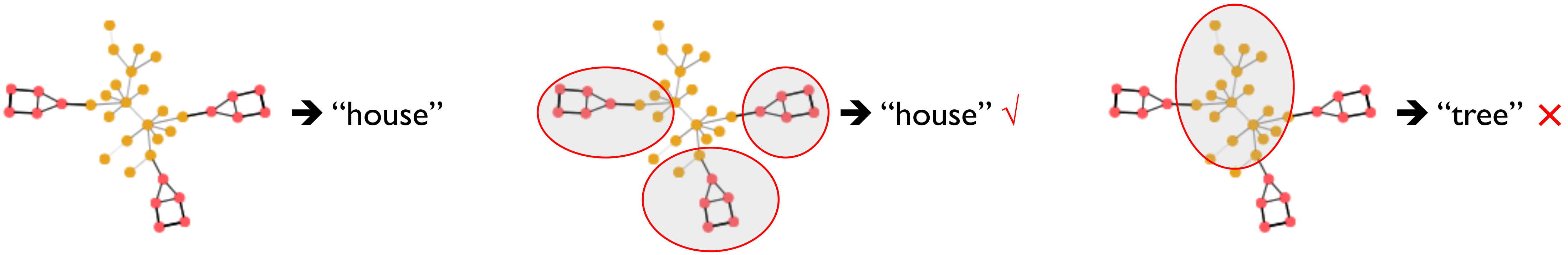
Method	TYPE	LEARNING	TASK	TARGET	BLACK-BOX	FLOW	DESIGN
SA [54], [55]	Instance-level	✗	GC/NC	N/E/NF	✗	Backward	✗
Guided BP [54]	Instance-level	✗	GC/NC	N/E/NF	✗	Backward	✗
CAM [55]	Instance-level	✗	GC	N	✗	Backward	✗
Grad-CAM [55]	Instance-level	✗	GC	N	✗	Backward	✗

no extra learning procedures

not originally designed for graphs

Class2: Perturbation-Based Methods

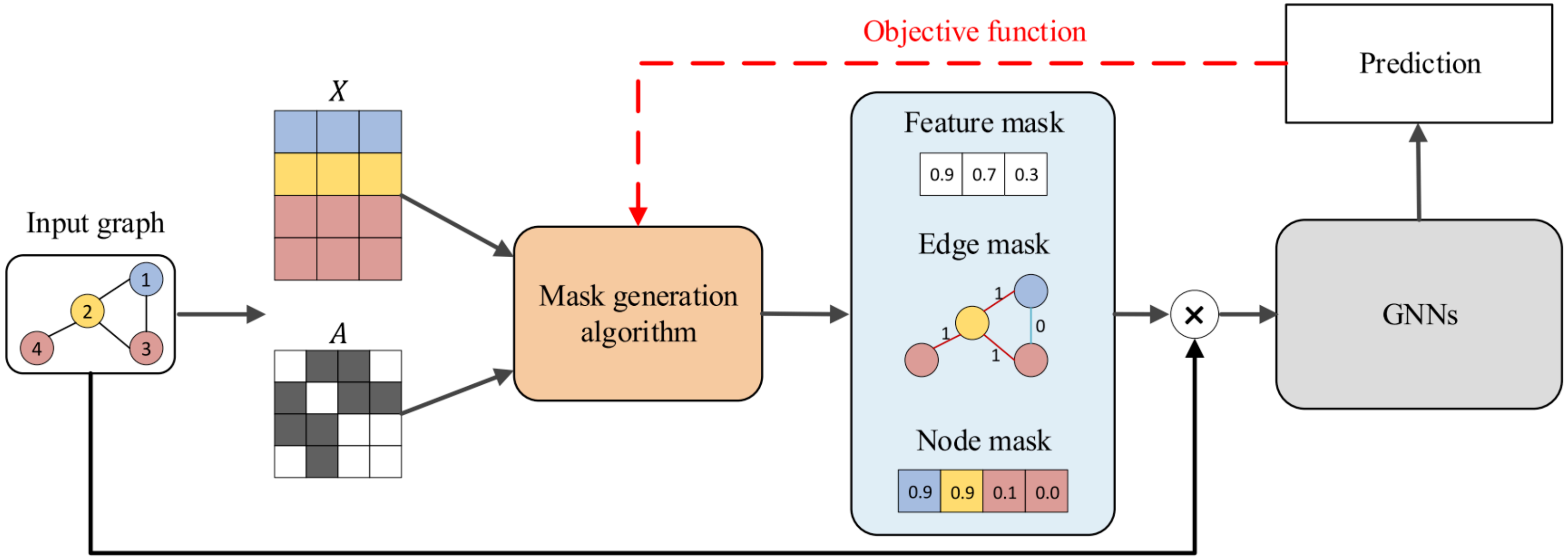
- [key idea] to study the output variations w.r.t input perturbations
  - intuitively,
  - when important input information is retained, the outputs should be similar to the original ones
  - when important input information is removed, the outputs should change greatly



with mask M_A with mask $(1 - M_A)$

Class2: Perturbation-Based Methods

A general framework:



The $A \cdot M_A$ here is a subgraph

Three key aspects here

- the mask generation algorithm
- the type of masks
- the objective function

$$M_A^* = \min_{M_A} Distance\left(f^*(A), f^*(A \cdot M_A)\right)$$
$$\text{s.t. } f^* = \min_{f} L_{cls}(f(A), Y)$$

in a bi-level form

Class2: Perturbation-Based Methods

Method	TYPE	LEARNING	TASK	TARGET	BLACK-BOX	FLOW	DESIGN
SA [54], [55]	Instance-level	✗	GC/NC	N/E/NF	✗	Backward	✗
Guided BP [54]	Instance-level	✗	GC/NC	N/E/NF	✗	Backward	✗
CAM [55]	Instance-level	✗	GC	N	✗	Backward	✗
Grad-CAM [55]	Instance-level	✗	GC	N	✗	Backward	✗
※ **GNNExplainer** [46]	Instance-level	✓	GC/NC	E/NF	✓	Forward	✓
※ **PGExplainer** [47]	Instance-level	✓	GC/NC	E	✗	Forward	✓
GraphMask [57]	Instance-level	✓	GC/NC	E	✗	Forward	✓
ZORRO [56]	Instance-level	✗	GC/NC	N/NF	✓	Forward	✓
Causal Screening [58]	Instance-level	✗	GC/NC	E	✓	Forward	✓
※ **SubgraphX** [48]	Instance-level	✓	GC/NC	Subgraph	✓	Forward	✓

representative methods

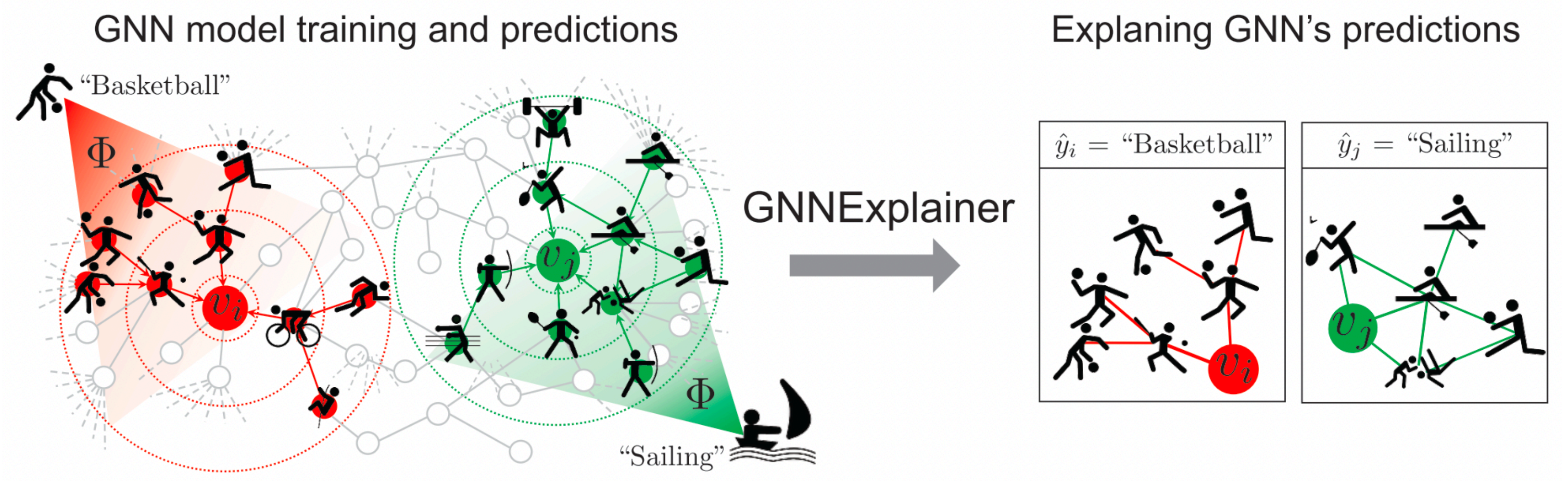
need extra learning procedures

originally designed for graphs

Class2 | GNNExplainer

Class2: Perturbation-Based Methods
- GNNExplainer
- PGExplainer
- SubgraphX

Given a trained GNN and its prediction yi=Basketball for node vi,
GNNExplainer identifies a small subgraph of the input graph that are most influential for yi



$$M_A^* = \min_{M_A} Distance\left(f^*(A), f^*(A \cdot M_A)\right)$$
$$\text{s.t. } f^* = \min_{f} L_{cls}(f(A), Y)$$

learn a mask for each node vi,vj…

GNNExplainer: Generating Explanations for Graph Neural Networks. NeurIPS 2019.

Class2 | GNNExplainer

- the first general, model-agnostic approach for providing explanations
- to learn soft masks for edges and features, treats masks as trainable variables

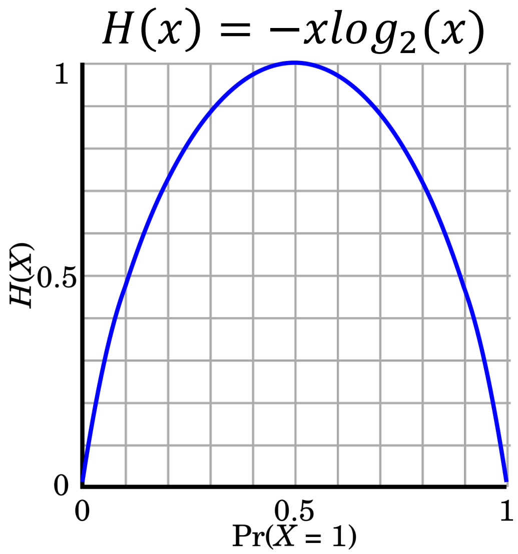
$$\max_{G_S} MI\left(Y, (G_S, X_S)\right) = H(Y) - H(Y|G = G_S, X = X_S)$$

$$\rightarrow \quad H(Y|G{=}G_S, X{=}X_S) = -\mathbb{E}_{Y|G_S,X_S}\left[\log P_\Phi(Y|G{=}G_S, X{=}X_S)\right].$$

$$\rightarrow \quad \min_M - \sum_{c=1}^{C} \boxed{\mathbb{1}[y = c]} \log P_\Phi(Y = y|\boxed{G = A_c \odot \sigma(M)}, X = X_c)$$

- the masks are optimized by maximizing the mutual information between the predictions of the original graph and the predictions of the sampled graph

Class2 | GNNExplainer



To avoid trivial solutions, constraints on the mask are necessary.

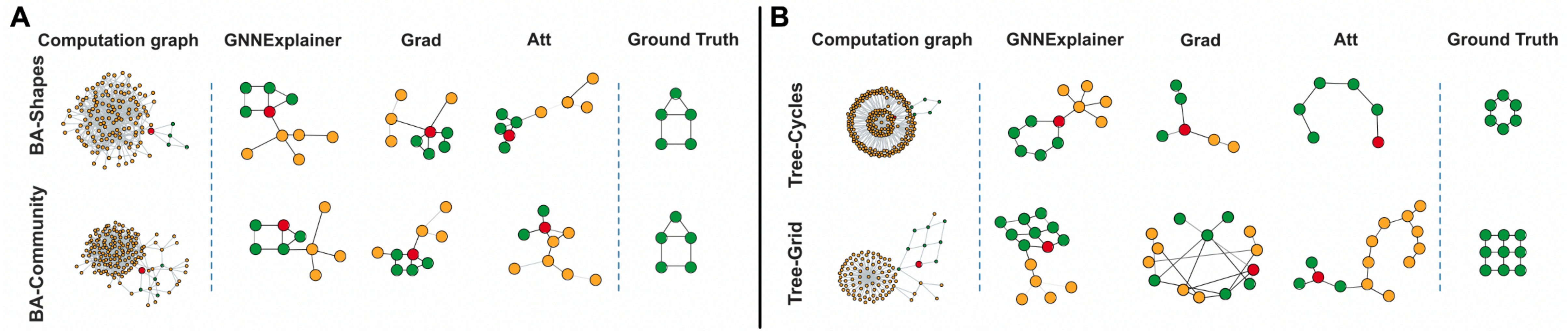
$$\max_{G_S} MI\left(Y, (G_S, X_S)\right) = H(Y) - H(Y|G = G_S, X = X_S)$$

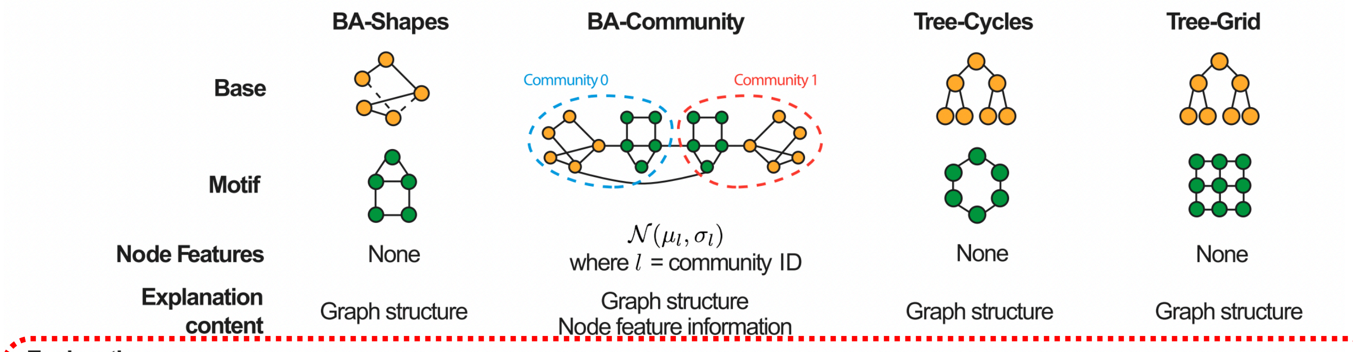
1. (entropy constraint) element-wise entropy to encourage the mask to be discrete.

2. (size constraint) size penalty as the sum of all elements in a mask.

3. (implicit connectivity constraint) get the largest connected subgraph as the explanation.

Class2 | GNNExplainer





Explanation accuracy	BA-Shapes	BA-Community	Tree-Cycles	Tree-Grid
Att	0.815	0.739	0.824	0.612
Grad	0.882	0.750	0.905	0.667
GNNExplainer	**0.925**	**0.836**	**0.948**	**0.875**

GNNExplainer is better than attention/gradient-based methods.

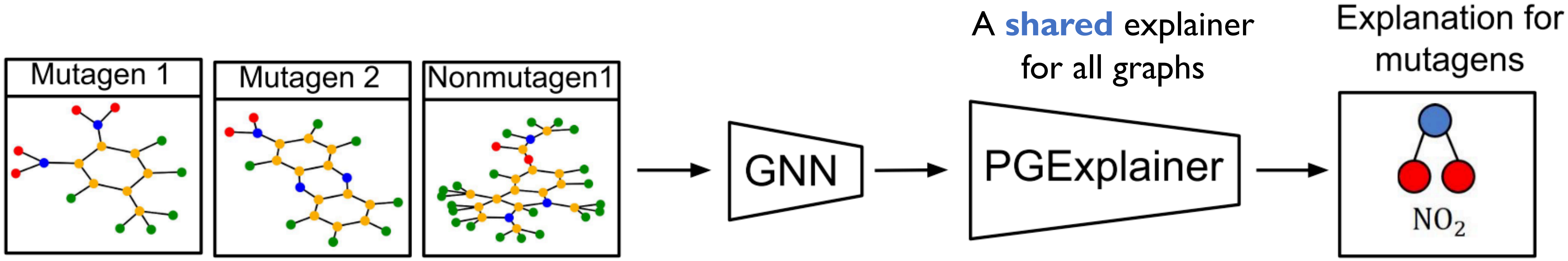
Take a break ☕

However, do we really need to

explain each prediction by learning its mask individually? 🤔

Class2 | PGExplainer

Class2: Perturbation-Based Methods
- GNNExplainer
- PGExplainer
- SubgraphX

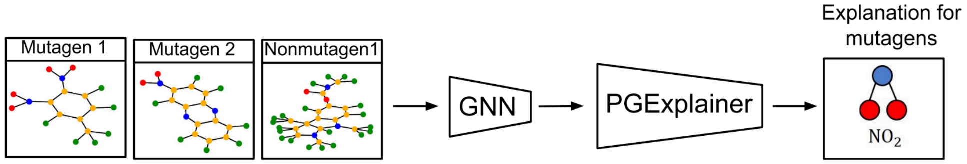


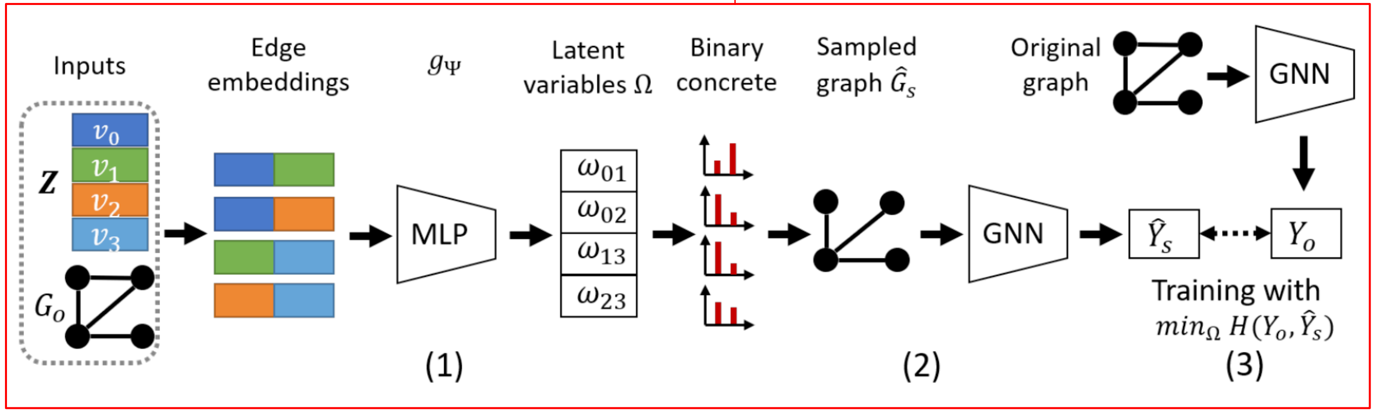
PGExplainer emphasize the collective and inductive nature of this problem

- the explanations can provide a global understanding of the trained GNNs

Parameterized explainer for graph neural network. NeurIPS 2020.

Class2 | PGExplainer





$$\max_{G_s} \mathrm{MI}(Y_o, G_s) = H(Y_o) - H(Y_o|G = G_s)$$

[training details]

Given an input graph,

1. it first obtains the embeddings for each edge by concatenating the corresponding node embeddings

2. uses the edge embeddings to predict the importance of each edge

3. the discrete masks are sampled via the reparameterization trick

4. Finally, the mask predictor is trained by maximizing the mutual information between the original predictions and new predictions

Class2 | PGExplainer

Constraints

In addition to the aforementioned entropy/size constraint,
PGExplainer also adopts a explicit connectivity constraint.

Reason: in many real-life scenarios, determinant motifs are expected to be connected.

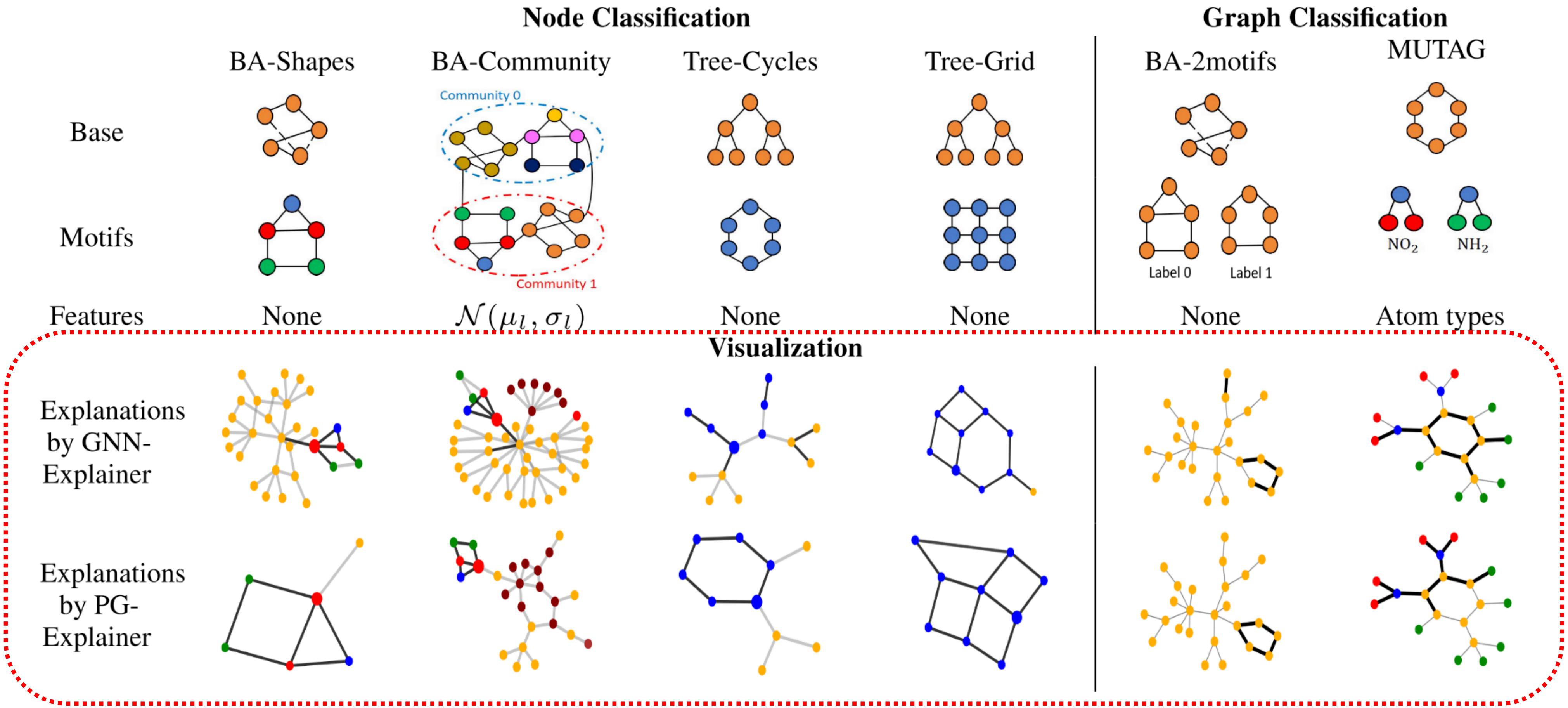
This constraint is implemented with the cross-entropy of adjacent edges connecting to the same node.

$$H(\hat{e}_{ij}, \hat{e}_{ik}) = -[(1 - \hat{e}_{ij}) \log(1 - \hat{e}_{ik}) + \hat{e}_{ij} \log \hat{e}_{ik}].$$

e.g., node j and node k both connected to the node i.
If edge (i, j) is selected in the the explanatory graph, then adjacent edge (i, k) should also be included

Class2 | PGExplainer

PGExplainer is better than GNNExplainer.



Take a break ☕

Connected subgraphs are more intuitive and human-intelligible.

However, the learned subgraphs by GNNExplainer/PGExplainer

are not always connected.

Is there a better way to extract connected subgraphs? 🤔

Class2 | SubgraphX

Class2: Perturbation-Based Methods
- GNNExplainer
- PGExplainer
- SubgraphX

- to explore subgraph-level explanations for GNNs
  - use Monte Carlo Tree Search to explore different subgraphs via node pruning
  - select the most important subgraph w.r.t. the Shapley value as the explanation

Monte Carlo Tree Search ➔
(the search algorithm)

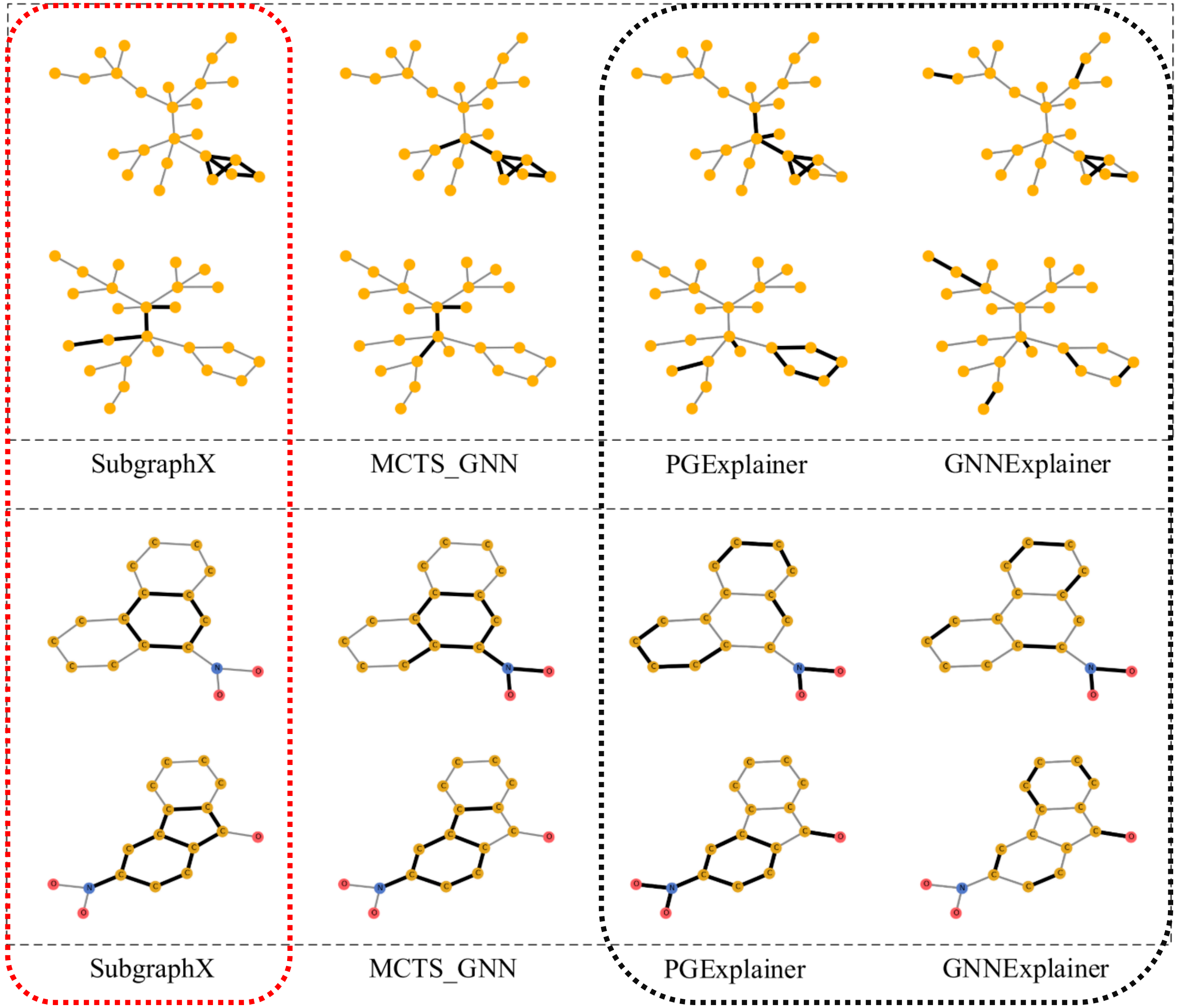
$$\mathcal{G} = \{\mathcal{G}_1, \cdots, \mathcal{G}_i, \cdots, \mathcal{G}_n\}$$

Shapley value ➔
(the score function)

$$\mathcal{G}^* = \underset{|\mathcal{G}_i| \leq N_{\min}}{\operatorname{argmax}} \operatorname{Score}(f(\cdot), \mathcal{G}, \mathcal{G}_i)$$

On Explainability of Graph Neural Networks via Subgraph Explorations. ICML 2021.

Class2 | SubgraphX



Method	SubgraphX	GNNExplainer	PGExplainer
TIME	77.8 ± 3.8s	16.2 ± 0.2s	0.02s (Training 362s)
FIDELITY	0.55	0.19	0.18

SubgraphX is better than GNNExplainer and PGExplainer.

On Explainability of Graph Neural Networks via Subgraph Explorations. ICML 2021.

Outline

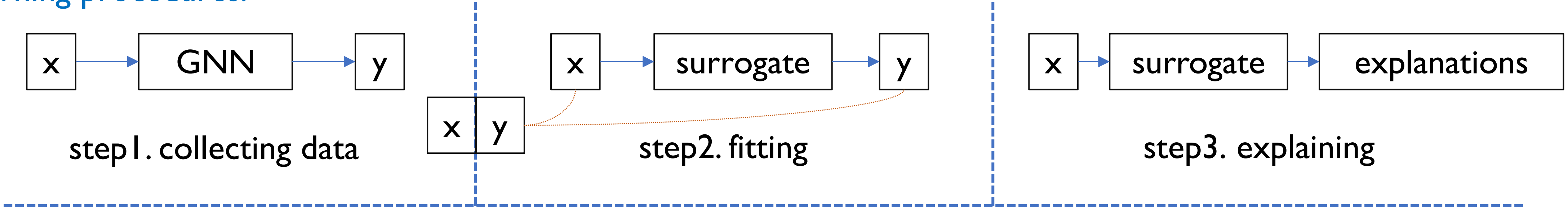
- Background
- A review of existing methods
  - Taxonomy
    - Class1: Gradients/Features-Based Methods
    - Class2: Perturbation-Based Methods
    - Class3: Surrogate-based methods
    - Class4: Decomposition-based methods
  - Metrics and evaluation
  - A brief summary
- Recent advances that go beyond the post-hoc manner
- Summary

Class3: Surrogate-based methods

[key idea] employ a simple and interpretable surrogate model

- to approximate the predictions of the complex deep model
- the explanations from the interpretable model are regarded as the explanations of the GNN

Learning procedures:



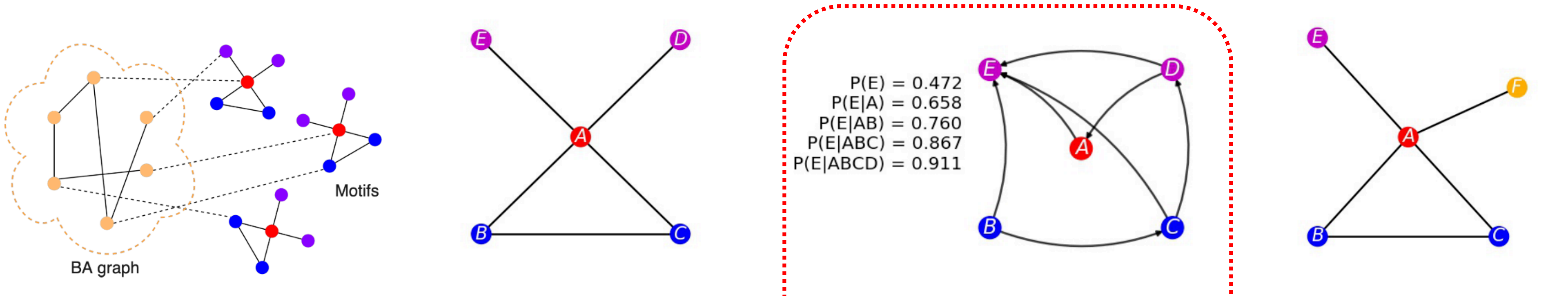
representative methods

Method	TYPE	LEARNING	TASK	TARGET	BLACK-BOX	FLOW	DESIGN
GraphLime [61]	Instance-level	✓	NC	NF	✓	Forward	✗
RelEx [62]	Instance-level	✓	NC	N/E	✓	Forward	✓
PGM-Explainer [63]	Instance-level	✓	GC/NC	N	✓	Forward	✓

key differences

- how to obtain the local dataset (x,y pairs)
- what interpretable surrogate model to use

Class3 | PGM-Explainer



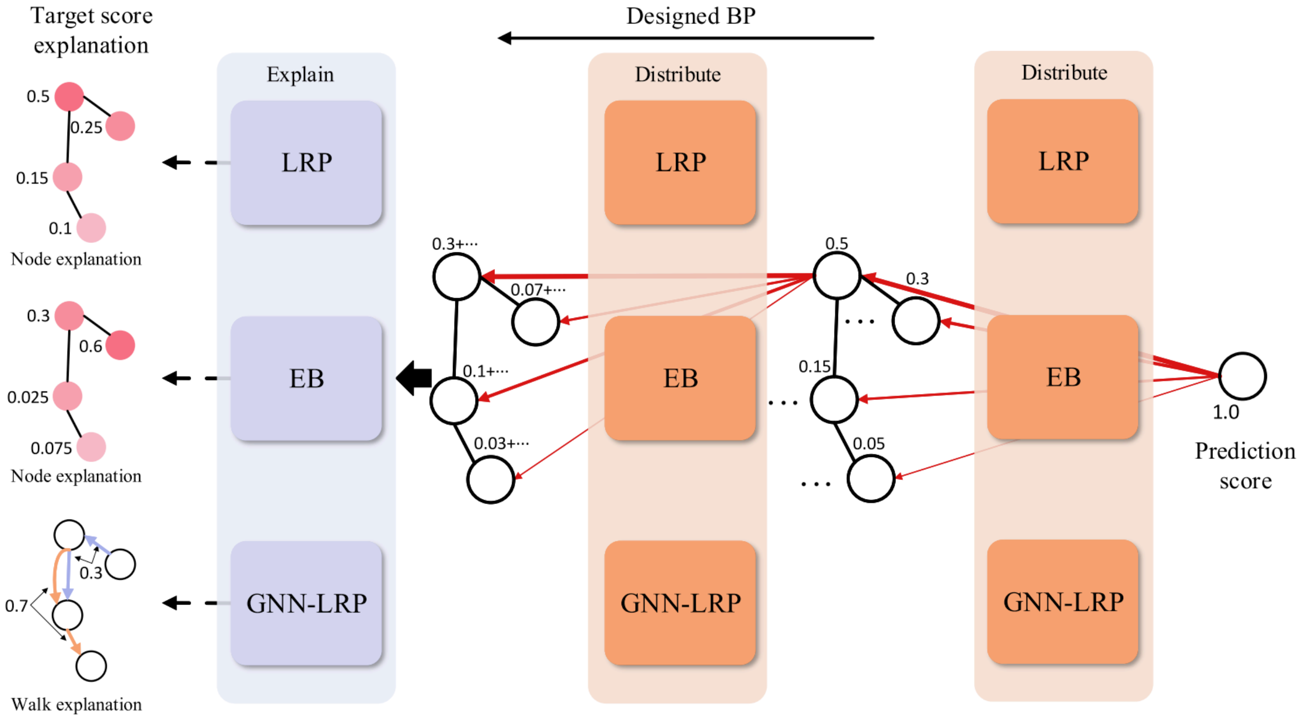
(a) Input graph. (b) Motif containing E. (c) PGM-Explainer. (d) GNNExplainer.

- to build a probabilistic graphical model to provide instance-level explanations
- an interpretable Bayesian network is employed to fit the local dataset
- then to explain the predictions of the original GNN model
  - e.g., estimate the probability that node E has the predicted role given other nodes

PGM-Explainer: Probabilistic Graphical Model Explanations for Graph Neural Networks. NeurIPS 2020.

Class4: Decomposition-based methods

[key idea]
to measure the importance of input features by decomposing the original model predictions into several terms.



Method	Type	Learning	Task	Target	Black-box	Flow	Design
LRP [54], [59]	Instance-level	✗	GC/NC	N	✗	Backward	✗
Excitation BP [55]	Instance-level	✗	GC/NC	N	✗	Backward	✗
GNN-LRP [60]	Instance-level	✗	GC/NC	Walk	✗	Backward	✓

It can only study the importance of different nodes but not the graph structures.

Take a break ☕

So far, we have reviewed the explanation methods of 4 classes.

How can we evaluate these methods? 🤔

Metrics and evaluation | Fidelity

Good explanations should be faithful to the model.

if the identified mask are discriminative to the model

when the mask is removed, the prediction should change significantly ➔ higher Fidelity+

when the mask is retained, the prediction should be similar ➔ lower Fidelity-

$\neg A \rightarrow \neg B$

(necessity)

$$Fidelity+^{prob} = \frac{1}{N}\sum_{i=1}^{N}(f(\mathcal{G}_i)_{y_i} - f(\mathcal{G}_i^{1-m_i})_{y_i}),$$

$A \rightarrow B$

(sufficiency)

$$Fidelity-^{prob} = \frac{1}{N}\sum_{i=1}^{N}(f(\mathcal{G}_i)_{y_i} - f(\mathcal{G}_i^{m_i})_{y_i})$$

Metrics and evaluation | Sparsity

Good explanations should be sparse and compact.

➔ capture the most important input features and ignore the irrelevant ones

$$Sparsity = \frac{1}{N}\sum_{i=1}^{N}(1 - \frac{|m_i|}{|M_i|}),$$

A fair comparison can be conducted under the same level of sparsity

Metrics and evaluation | Stability & Accuracy

Good explanations should be stable.

- when small changes are applied to the input without affecting the predictions
- the explanations should remain similar

Good explanations should be accurate.

- compare the explanations with such ground truths
- the closer to ground truths, the better explanations
- specific metrics here can be accuracy, F1 score, ROC-AUC score.
- *however, the Accuracy metric cannot be applied to real-world datasets due to the lack of ground truths.*

Metrics and evaluation

SubgraphX is the best method here

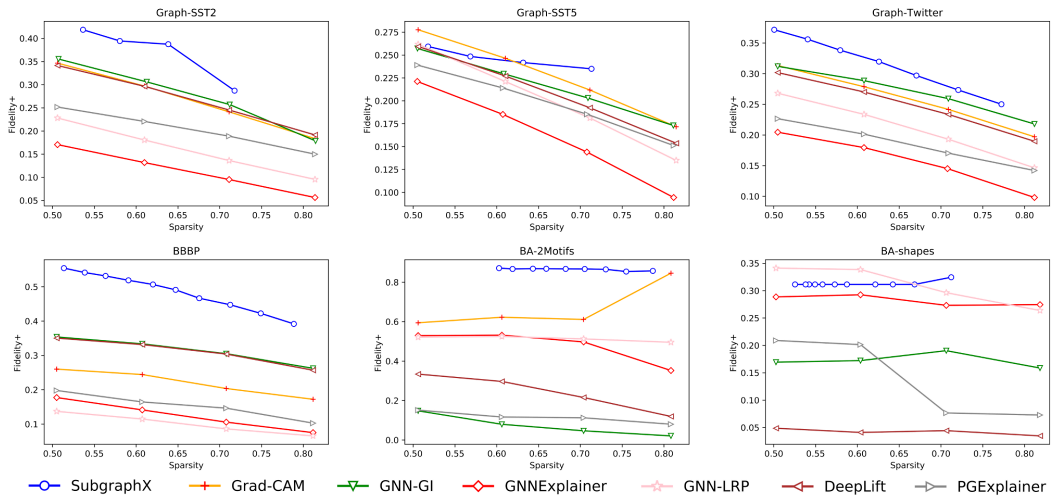


Fig. 6. The Fidelity+ comparisons between different GNN explanation techniques under different Sparsity levels.

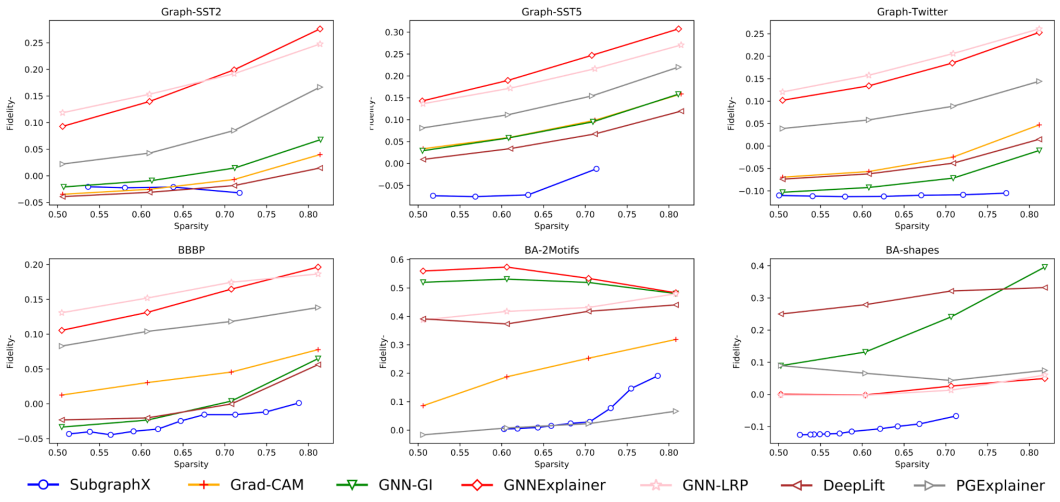


Fig. 7. The Fidelity- comparisons between different GNN explanation techniques under different Sparsity levels.

higher Fidelity+ 👍

TABLE 3
The Fidelity+ comparisons between different GNN explanation techniques and the random designation baseline.

Methods	Graph-Twitter	Graph-SST2	Graph-SST5	BBBP	BA-2Motifs	BA-Shapes
			Sparsity=0.7			Sparsity=0.6
Random	0.1342	0.0915	0.1419	0.1212	0.4903	0.1884
SubgraphX	0.2836	0.3152	0.2351	0.4521	0.8642	0.3171
Grad-CAM	0.2418	0.2414	0.2118	0.2036	0.6112	N/A
GNN-GI	0.2593	0.2571	0.2031	0.3051	0.0466*	0.1723*
GNNExplainer	0.1452	0.0953	0.1441	0.1057*	0.4972	0.2925
PGExplainer	0.1704	0.1889	0.1854	0.1464	0.1126*	0.2015
GNN-LRP	0.1931	0.1363	0.1813	0.0860*	0.5125	0.3386
DeepLift	0.2336	0.2454	0.1924	0.3039	0.2156*	0.0411*

lower Fidelity- 👍

TABLE 4
The Fidelity- comparisons between different GNN explanation techniques and the random designation baseline.

Methods	Graph-Twitter	Graph-SST2	Graph-SST5	BBBP	BA-2Motifs	BA-Shapes
			Sparsity=0.7			Sparsity=0.6
Random	0.2825	0.2745	0.2961	0.2168	0.5394	0.2567
SubgraphX	-0.1085	-0.0288	-0.0298	-0.0169	0.0686	-0.0792
Grad-CAM	-0.0245	-0.0069	0.0987	0.0456	0.2529	N/A
GNN-GI	-0.0715	0.0147	0.0951	0.0039	0.5193	0.1318
GNNExplainer	0.1848	0.1992	0.2471	0.1647	0.5337	-0.0017
PGExplainer	0.0887	0.0852	0.1543	0.1183	0.0227	0.0658
GNN-LRP	0.2060	0.1919	0.2164	0.1746	0.4314	-0.0026
DeepLift	-0.0382	-0.0183	0.0674	-0.0002	0.4179	0.2790*

TABLE 5
The Accuracy and Stability comparisons between different GNN explanation techniques.

Methods	BA-shapes		BA-Community	
Metric	Accuracy	Stability	Accuracy	Stability
GNN-GI	0.8369	0.1361	0.8291	0.1723
GNNExplainer	0.8786	0.1721	0.9194	0.1820
PGExplainer	0.7147	0.0522	0.6843	0.1177
GNN-LRP	0.9243	0.1872	0.8357	0.1239
DeepLift	0.5698	0.0432	0.4190	0.0842

Outline

- Background
- A review of existing methods
  - Taxonomy
  - Metrics and evaluation
  - A brief summary
- Recent advances that go beyond the post-hoc manner
- Summary

Taxonomy | summary

Class1: Gradients/Features-Based Methods
Class2: Perturbation-Based Methods
Class3: Surrogate-based methods
Class4: Decomposition-based methods

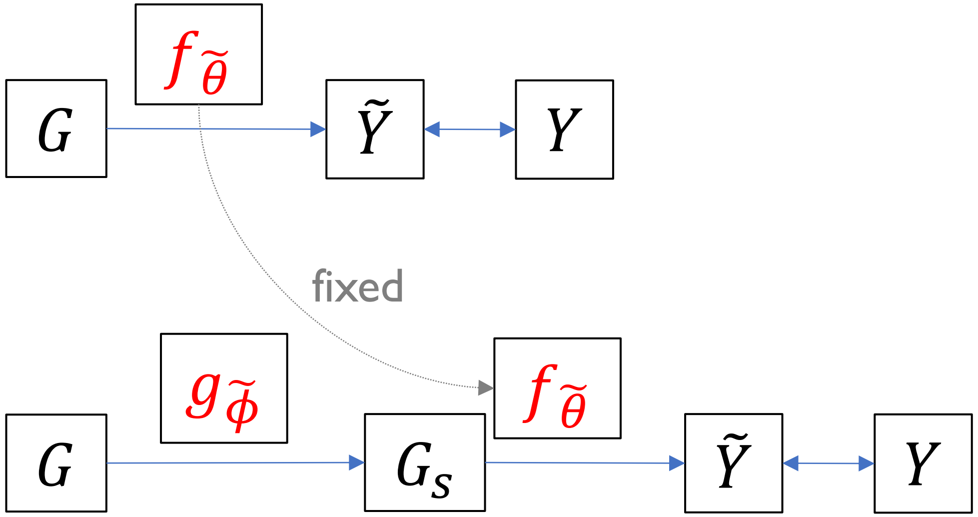
The post-hoc methods are popular

step1: obtain the model parameter $\tilde{\theta}$

- i.e., train the predictor

step2: optimize the subgraph extractor $\tilde{\phi}$

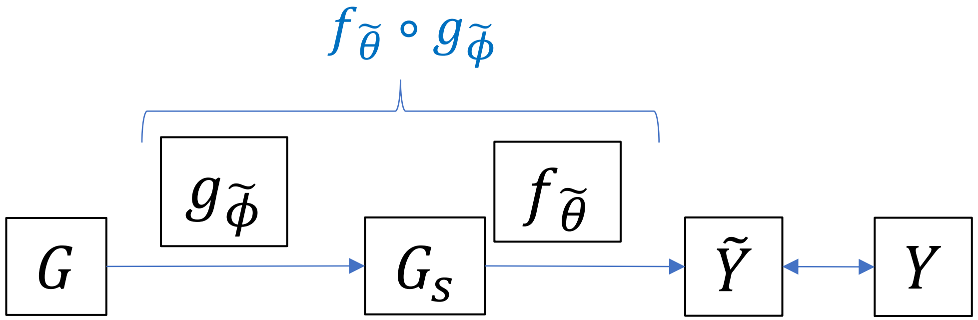
- approximate the MI: $I(G_S; \tilde{Y}) - I(G; \tilde{Y}) \to 0$
- usually with constraints (e.g., size, connectivity)



$f_{\tilde{\theta}} \circ g_{\tilde{\phi}}$ is the interpretating system

As $\text{Accuracy}(f_{\tilde{\theta}}) < \text{Accuracy}(f_{\tilde{\theta}} \circ g_{\tilde{\phi}})$ is usual,
can we provide interpretation without sacrificing the accuracy? 🤔

Take a break ☕



a joint training of $f_{\tilde{\theta}} \circ g_{\tilde{\phi}}$ might be better? 🤔

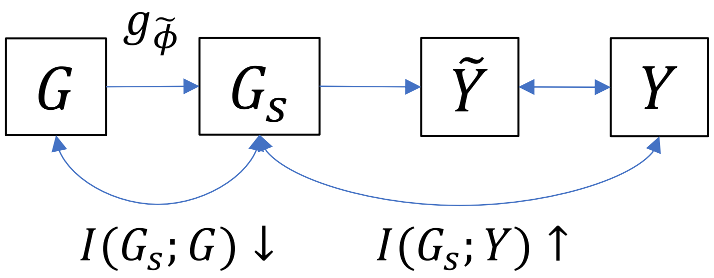
Outline

- Background
- A review of existing methods
- Recent advances that go beyond the post-hoc manner
- Summary

GSAT | method

use information constraint to select label-relevant subgraph

- inspired by the Graph Information Bottleneck (GIB)
- form a joint learning framework of $f_{\tilde{\theta}}$ and $g_{\tilde{\phi}}$



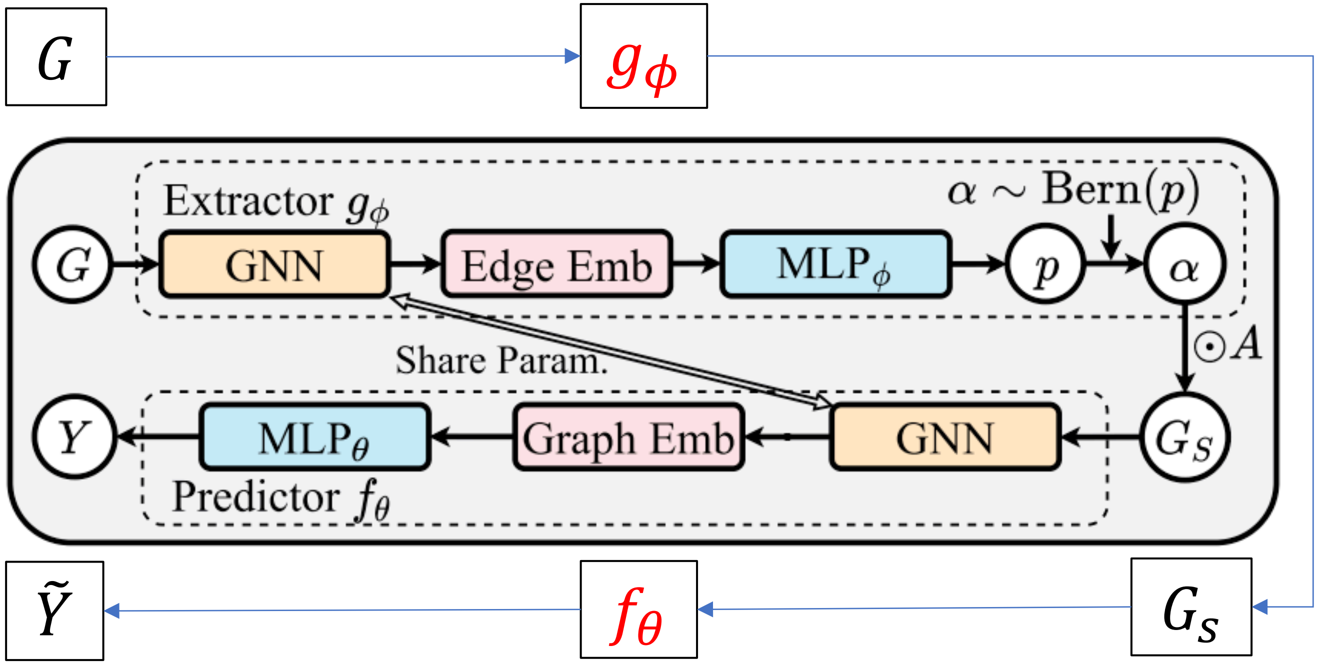
$$\min_{\phi} -I(G_S; Y) + \beta I(G_S; G), \text{ s.t. } G_S \sim g_\phi(G)$$

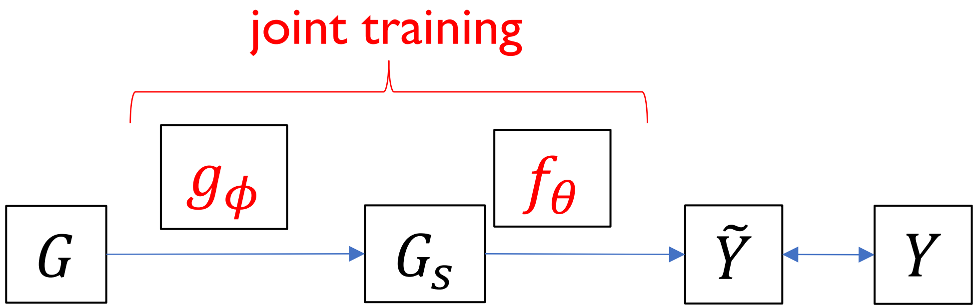
do not impose any potentially biased constraints

- e.g., graph size or connectivity

Interpretable and Generalizable Graph Learning via Stochastic Attention Mechanism. ICML 2022.

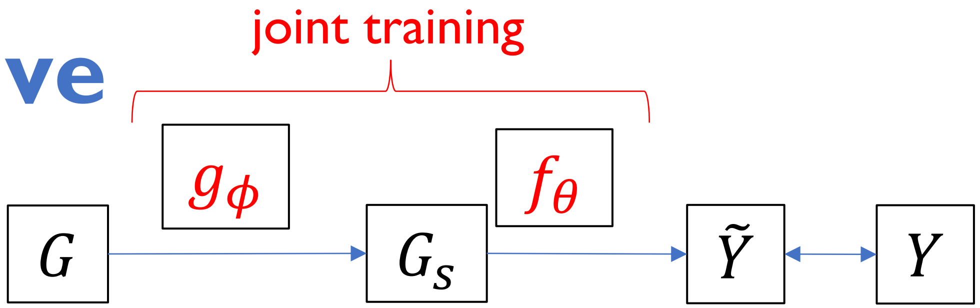
GSAT | method





$$\min_{\phi} -I(G_S; Y) + \beta I(G_S; G), \text{ s.t. } G_S \sim g_\phi(G)$$

GSAT | Full learning objective



$$\min_{\phi} -I(G_S;Y) + \beta I(G_S;G), \text{ s.t. } G_S \sim g_\phi(G)$$

$$I(G_s;G) \leq \mathbb{E}_G\left[\mathbf{KL}(\mathbb{P}_\phi(G_S|G)||\mathbb{Q}(G_S))\right] \quad (g_{\tilde{\phi}})$$

$$I(G_S;Y) \geq \mathbb{E}_{G_S,Y}\left[\log \mathbb{P}_\theta(Y|G_S)\right] + H(Y) \quad (f_{\tilde{\theta}})$$

$$\min_{\theta,\phi} -\mathbb{E}\left[\log \mathbb{P}_\theta(Y|G_S)\right] + \beta \mathbb{E}\left[\mathbf{KL}(\mathbb{P}_\phi(G_S|G)||\mathbb{Q}(G_S))\right]$$

GSAT | Experiment

Interpretation 👍

Table 1. Interpretation Performance (AUC). The underlined results highlight the best baselines. The **bold** font and **bold**† font highlight when GSAT outperform the means of the best baselines based on the mean of GSAT and the mean-2*std of GSAT, respectively.

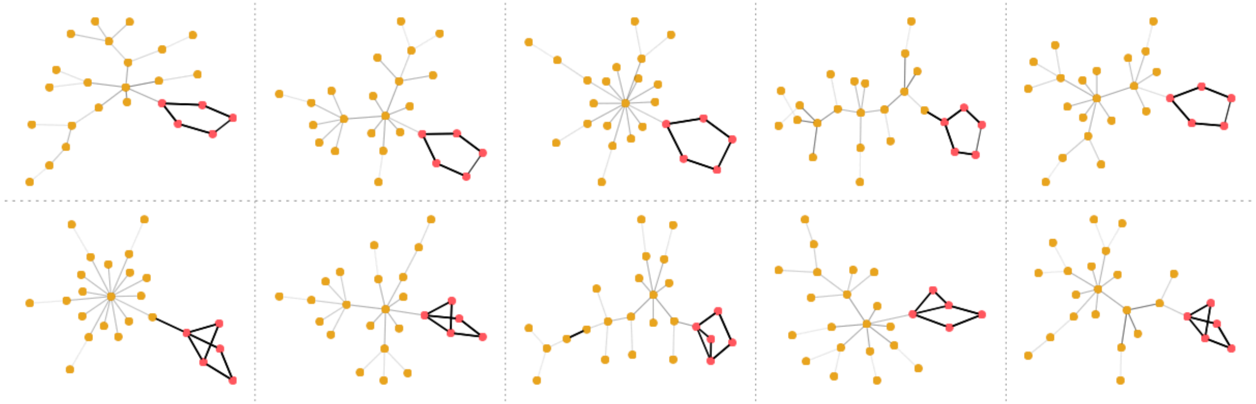
	BA-2MOTIFS	MUTAG	MNIST-75SP	SPURIOUS-MOTIF $b = 0.5$	SPURIOUS-MOTIF $b = 0.7$	SPURIOUS-MOTIF $b = 0.9$
GNNEXPLAINER	67.35 ± 3.29	61.98 ± 5.45	59.01 ± 2.04	62.62 ± 1.35	62.25 ± 3.61	58.86 ± 1.93
PGEXPLAINER	84.59 ± 9.09	60.91 ± 17.10	69.34 ± 4.32	69.54 ± 5.64	72.33 ± 9.18	72.34 ± 2.91
GRAPHMASK	92.54 ± 8.07	62.23 ± 9.01	73.10 ± 6.41	72.06 ± 5.58	73.06 ± 4.91	66.68 ± 6.96
IB-SUBGRAPH	86.06 ± 28.37	91.04 ± 6.59	51.20 ± 5.12	57.29 ± 14.35	62.89 ± 15.59	47.29 ± 13.39
DIR	82.78 ± 10.97	64.44 ± 28.81	32.35 ± 9.39	78.15 ± 1.32	77.68 ± 1.22	49.08 ± 3.66
GIN+GSAT	$\mathbf{98.74}^{\dagger} \pm 0.55$	$\mathbf{99.60}^{\dagger} \pm 0.51$	$\mathbf{83.36}^{\dagger} \pm 1.02$	$\mathbf{78.45} \pm 3.12$	74.07 ± 5.28	71.97 ± 4.41
GIN+GSAT*	$\mathbf{97.43}^{\dagger} \pm 1.77$	$\mathbf{97.75}^{\dagger} \pm 0.92$	$\mathbf{83.70}^{\dagger} \pm 1.46$	$\mathbf{85.55}^{\dagger} \pm 2.57$	$\mathbf{85.56}^{\dagger} \pm 1.93$	$\mathbf{83.59}^{\dagger} \pm 2.56$
PNA+GSAT	$\mathbf{93.77} \pm 3.90$	$\mathbf{99.07}^{\dagger} \pm 0.50$	$\mathbf{84.68}^{\dagger} \pm 1.06$	$\mathbf{83.34}^{\dagger} \pm 2.17$	$\mathbf{86.94}^{\dagger} \pm 4.05$	$\mathbf{88.66}^{\dagger} \pm 2.44$
PNA+GSAT*	89.04 ± 4.92	$\mathbf{96.22}^{\dagger} \pm 2.08$	$\mathbf{88.54}^{\dagger} \pm 0.72$	$\mathbf{90.55}^{\dagger} \pm 1.48$	$\mathbf{89.79}^{\dagger} \pm 1.91$	$\mathbf{89.54}^{\dagger} \pm 1.78$

Prediction 👍

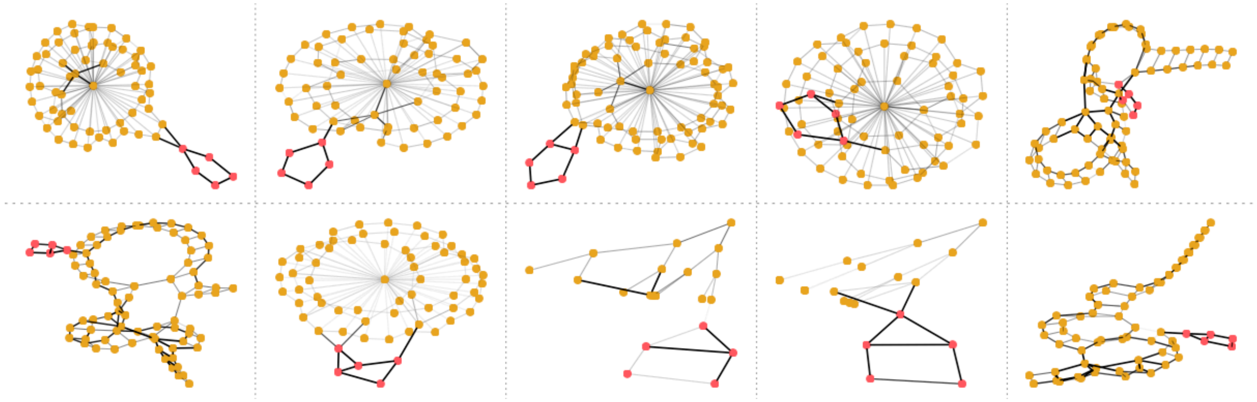
Table 2. Prediction Performance (Acc.). The **bold** font highlights the inherently interpretable methods that significantly outperform the corresponding backbone model, GIN or PNA, when the mean-1*std of a method > the mean of its corresponding backbone model.

	MOLHIV (AUC)	GRAPH-SST2	MNIST-75SP	SPURIOUS-MOTIF $b = 0.5$	SPURIOUS-MOTIF $b = 0.7$	SPURIOUS-MOTIF $b = 0.9$
GIN	76.69 ± 1.25	82.73 ± 0.77	95.74 ± 0.36	39.87 ± 1.30	39.04 ± 1.62	38.57 ± 2.31
IB-SUBGRAPH	76.43 ± 2.65	82.99 ± 0.67	93.10 ± 1.32	$\mathbf{54.36} \pm 7.09$	$\mathbf{48.51} \pm 5.76$	$\mathbf{46.19} \pm 5.63$
DIR	76.34 ± 1.01	82.32 ± 0.85	88.51 ± 2.57	$\mathbf{45.49} \pm 3.81$	41.13 ± 2.62	37.61 ± 2.02
GIN+GSAT	76.47 ± 1.53	82.95 ± 0.58	$\mathbf{96.24} \pm 0.17$	$\mathbf{52.74} \pm 4.08$	$\mathbf{49.12} \pm 3.29$	$\mathbf{44.22} \pm 5.57$
GIN+GSAT*	76.16 ± 1.39	82.57 ± 0.71	$\mathbf{96.21} \pm 0.14$	$\mathbf{46.62} \pm 2.95$	41.26 ± 3.01	39.74 ± 2.20
PNA (NO SCALARS)	78.91 ± 1.04	79.87 ± 1.02	87.20 ± 5.61	68.15 ± 2.39	66.35 ± 3.34	61.40 ± 3.56
PNA+GSAT	$\mathbf{80.24} \pm 0.73$	$\mathbf{80.92} \pm 0.66$	$\mathbf{93.96} \pm 0.92$	68.74 ± 2.24	64.38 ± 3.20	57.01 ± 2.95
PNA+GSAT*	$\mathbf{80.67} \pm 0.95$	$\mathbf{82.81} \pm 0.56$	$\mathbf{92.38} \pm 1.44$	$\mathbf{69.72} \pm 1.93$	$\mathbf{67.31} \pm 1.86$	61.49 ± 3.46

GSAT | Experiment



since the GSAT dose not make any assumptions on the selected subgraphs,
the improvement of GSAT can be even more
if the true subgraph are dis-connected or vary in sizes.



Outline

- Background
- A review of existing methods
- Recent advances that go beyond the post-hoc manner
- Summary

Summary

A review of 4-class explanation methods

- Class1: Gradients/Features-Based Methods
- **Class2: Perturbation-Based Methods**
- Class3: Surrogate-based methods
- Class4: Decomposition-based methods

Future directions

- go beyond the post-hoc manner
- model-level explanation
- explain for KG reasoners and corresponding analysis

Related works | interpretating GNN

Most Influential

1. **Explainability in graph neural networks: A taxonomic survey**. *Yuan Hao, Yu Haiyang, Gui Shurui, Ji Shuiwang*. ARXIV 2020. paper
2. **Gnnexplainer: Generating explanations for graph neural networks**. *Ying Rex, Bourgeois Dylan, You Jiaxuan, Zitnik Marinka, Leskovec Jure*. NeurIPS 2019. paper code
3. **Explainability methods for graph convolutional neural networks**. *Pope Phillip E, Kolouri Soheil, Rostami Mohammad, Martin Charles E, Hoffmann Heiko*. CVPR 2019.paper
4. **Parameterized Explainer for Graph Neural Network**. *Luo Dongsheng, Cheng Wei, Xu Dongkuan, Yu Wenchao, Zong Bo, Chen Haifeng, Zhang Xiang*. NeurIPS 2020. paper code
5. **Xgnn: Towards model-level explanations of graph neural networks**. *Yuan Hao, Tang Jiliang, Hu Xia, Ji Shuiwang*. KDD 2020. paper.
6. **Evaluating Attribution for Graph Neural Networks**. *Sanchez-Lengeling Benjamin, Wei Jennifer, Lee Brian, Reif Emily, Wang Peter, Qian Wesley, McCloskey Kevin, Colwell Lucy, Wiltschko Alexander*. NeurIPS 2020.paper
7. **PGM-Explainer: Probabilistic Graphical Model Explanations for Graph Neural Networks**. *Vu Minh, Thai My T.*. NeurIPS 2020.paper
8. **Explanation-based Weakly-supervised Learning of Visual Relations with Graph Networks**. *Federico Baldassarre and Kevin Smith and Josephine Sullivan and Hossein Azizpour*. ECCV 2020.paper
9. **GCAN: Graph-aware Co-Attention Networks for Explainable Fake News Detection on Social Media**. *Lu, Yi-Ju and Li, Cheng-Te*. ACL 2020.paper
10. **On Explainability of Graph Neural Networks via Subgraph Explorations**. *Yuan Hao, Yu Haiyang, Wang Jie, Li Kang, Ji Shuiwang*. ICML 2021.paper

Recent SOTA

1. **Quantifying Explainers of Graph Neural Networks in Computational Pathology**. *Jaume Guillaume, Pati Pushpak, Bozorgtabar Behzad, Foncubierta Antonio, Anniciello Anna Maria, Feroce Florinda, Rau Tilman, Thiran Jean-Philippe, Gabrani Maria, Goksel Orcun*. Proceedings of the IEEECVF Conference on Computer Vision and Pattern Recognition CVPR 2021.paper
2. **Counterfactual Supporting Facts Extraction for Explainable Medical Record Based Diagnosis with Graph Network**. *Wu Haoran, Chen Wei, Xu Shuang, Xu Bo*. NAACL 2021. paper
3. **When Comparing to Ground Truth is Wrong: On Evaluating GNN Explanation Methods**. *Faber Lukas, K. Moghaddam Amin, Wattenhofer Roger*. KDD 2021. paper
4. **Counterfactual Graphs for Explainable Classification of Brain Networks**. *Abrate Carlo, Bonchi Francesco*. Proceedings of the th ACM SIGKDD Conference on Knowledge Discovery Data Mining KDD 2021. paper
5. **Explainable Subgraph Reasoning for Forecasting on Temporal Knowledge Graphs**. *Zhen Han, Peng Chen, Yunpu Ma, Volker Tresp*. International Conference on Learning Representations ICLR 2021.paper
6. **Generative Causal Explanations for Graph Neural Networks**. *Lin Wanyu, Lan Hao, Li Baochun*. Proceedings of the th International Conference on Machine Learning ICML 2021.paper
7. **Improving Molecular Graph Neural Network Explainability with Orthonormalization and Induced Sparsity**. *Henderson Ryan, Clevert Djork-Arné, Montanari Floriane*. Proceedings of the th International Conference on Machine Learning ICML 2021.paper
8. **Explainable Automated Graph Representation Learning with Hyperparameter Importance**. *Wang Xin, Fan Shuyi, Kuang Kun, Zhu Wenwu*. Proceedings of the th International Conference on Machine Learning ICML 2021.paper
9. **Higher-order explanations of graph neural networks via relevant walks**. *Schnake Thomas, Eberle Oliver, Lederer Jonas, Nakajima Shinichi, Schütt Kristof T, Müller Klaus-Robert, Montavon Grégoire*. arXiv preprint arXiv:2006.03589 2020. paper
10. **HENIN: Learning Heterogeneous Neural Interaction Networks for Explainable Cyberbullying Detection on Social Media**. *Chen, Hsin-Yu and Li, Cheng-Te*. EMNLP 2020. paper

https://github.com/THUDM/cogdl/blob/master/gnn_papers.md#explainability

Q&A

Thanks for your attention!

interpretable v.s. explainable

- We consider a model to be "interpretable" if the model itself can provide humanly understandable interpretations of its predictions
  - Note that such a model is no longer a black box to some extent.
  - For example, a decision tree model is an "interpretable" one.

- Meanwhile, an "explainable" model implies that model is still a black box
  - its predictions could potentially be understood by post-hoc techniques.

Explainability in Graph Neural Networks: A Taxonomic Survey. TPAMI 2022.

Taxonomy

TABLE 1
A comprehensive analysis of different explanation methods. Here "Type" indicates what type of explanations are provided, "Learning" denotes whether learning procedures are involved, "Task" means what tasks each method can be applied to, "Target" indicates the targets of explanations, "Black-box" means if the trained GNNs are treated as a black-box during the explanation stage, "Flow" denotes the computational flow for explanations, and "Design" indicates whether an explanation method has specific designs for graph data. Note that GC denotes graph classification, NC denotes node classification, N means nodes, E means edges, NF represents node features, and Walk indicates graph walks.

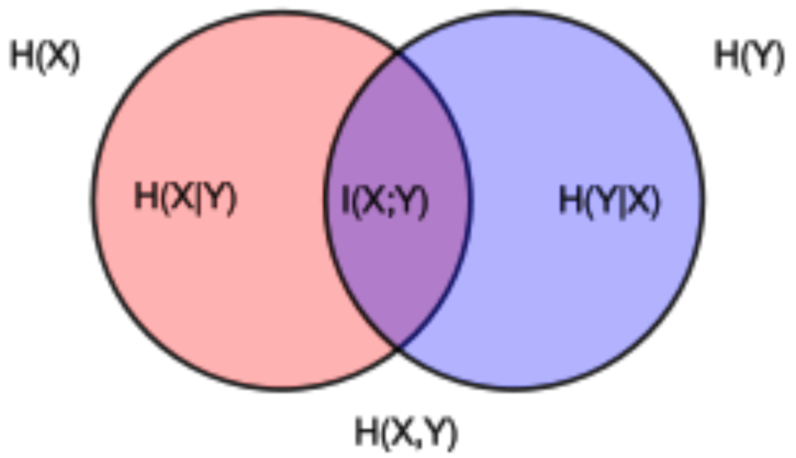
Method	Type	Learning	Task	Target	Black-box	Flow	Design
SA [54], [55]	Instance-level	✗	GC/NC	N/E/NF	✗	Backward	✗
Guided BP [54]	Instance-level	✗	GC/NC	N/E/NF	✗	Backward	✗
CAM [55]	Instance-level	✗	GC	N	✗	Backward	✗
Grad-CAM [55]	Instance-level	✗	GC	N	✗	Backward	✗
GNNExplainer [46]	Instance-level	✓	GC/NC	E/NF	✓	Forward	✓
PGExplainer [47]	Instance-level	✓	GC/NC	E	✗	Forward	✓
GraphMask [57]	Instance-level	✓	GC/NC	E	✗	Forward	✓
ZORRO [56]	Instance-level	✗	GC/NC	N/NF	✓	Forward	✓
Causal Screening [58]	Instance-level	✗	GC/NC	E	✓	Forward	✓
SubgraphX [48]	Instance-level	✓	GC/NC	Subgraph	✓	Forward	✓
LRP [54], [59]	Instance-level	✗	GC/NC	N	✗	Backward	✗
Excitation BP [55]	Instance-level	✗	GC/NC	N	✗	Backward	✗
GNN-LRP [60]	Instance-level	✗	GC/NC	Walk	✗	Backward	✓
GraphLime [61]	Instance-level	✓	NC	NF	✓	Forward	✗
RelEx [62]	Instance-level	✓	NC	N/E	✓	Forward	✓
PGM-Explainer [63]	Instance-level	✓	GC/NC	N	✓	Forward	✓
XGNN [45]	Model-level	✓	GC	Subgraph	✓	Forward	✓

Explainability in Graph Neural Networks: A Taxonomic Survey. TPAMI 2022.

Preliminaries | mutual information

- the mutual information (MI) of two random variables
  is a measure of the mutual dependence between the two variables.

- $I(X;Y) = H(X) - H(X|Y)$
- $I(X;Y) = H(Y) - H(Y|X)$
- $I(X;Y) = H(X) + H(Y) - H(X,Y)$



https://en.wikipedia.org/wiki/Mutual_information

Preliminaries | mutual information

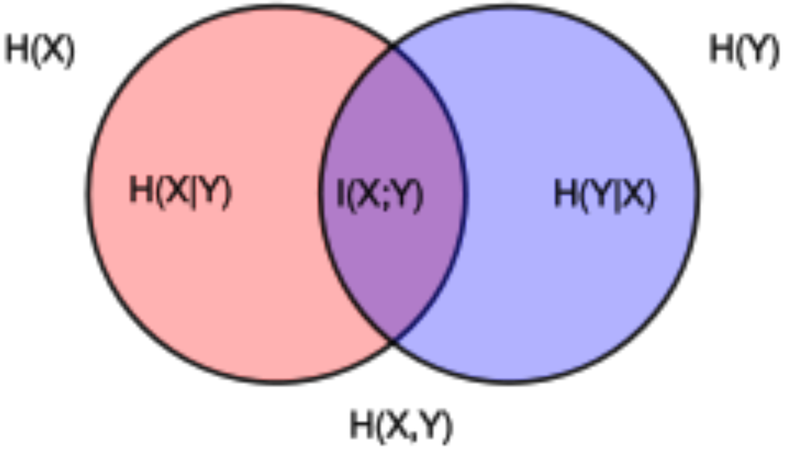
- the mutual information (MI) of two random variables
  is a measure of the mutual dependence between the two variables.

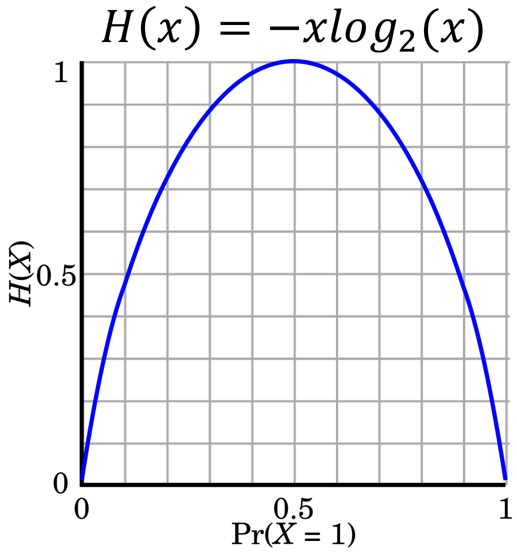
- definition: $I(X;Y) = I(Y;X) = D_{KL}(p(x,y)||p(x)\otimes p(y))$
- discrete variables: $I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log(\frac{p(x,y)}{p(x)p(y)})$
- continuous variables: $I(X;Y) = \int_Y \int_X p(x,y)\log(\frac{p(x,y)}{p(x)p(y)})$

https://en.wikipedia.org/wiki/Mutual_information

Preliminaries | mutual information

- $I(X;Y) = H(X) - H(X|Y)$
- $I(X;Y) = H(Y) - H(Y|X)$
- $I(X;Y) = H(X) + H(Y) - H(X,Y)$



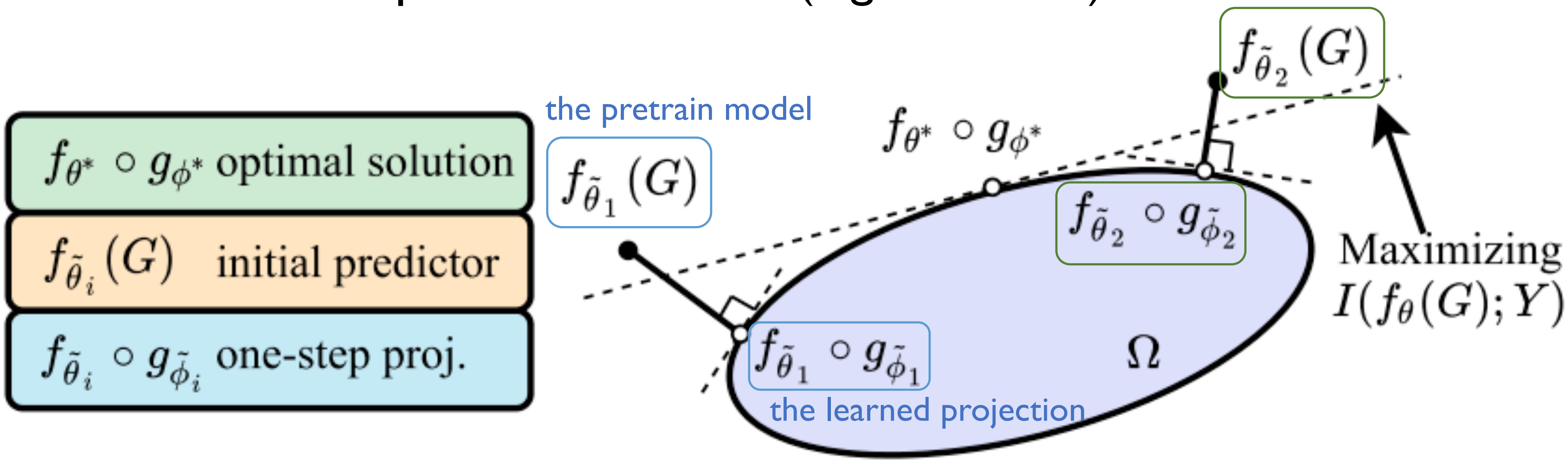


The existing post-hoc methods | problems

Post-hoc methods just perform one-step projection to the information-constrained space (Ω)

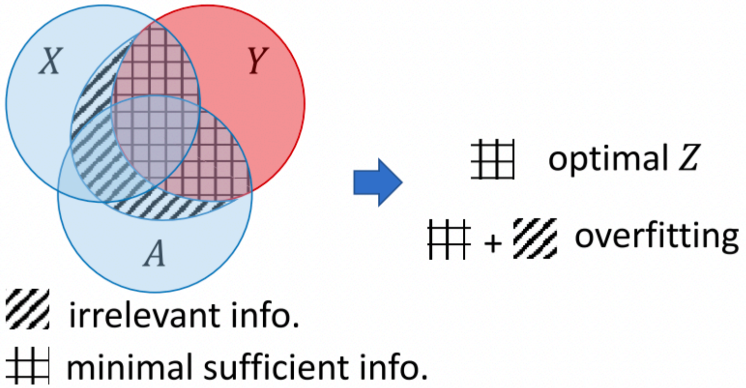
cons

- always suboptimal (low accuracy)
- sensitive to the pre-trained model (high variance)



Graph information bottleneck (GIB)

D=(X,A) → (GNN) → **Z** ↔ **Y**



Y: The target, $\mathcal{D}$: The input data ($= (A, X)$)
A: The graph structure, X: The node features
Z: The representation

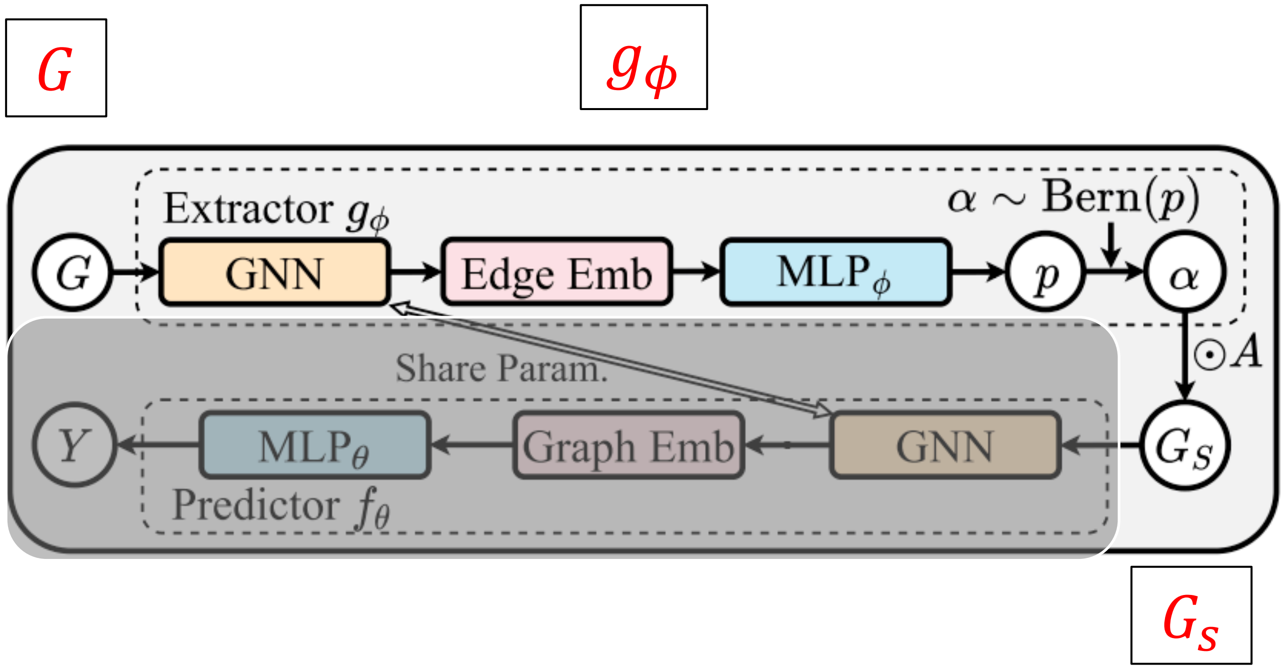
Graph Information Bottleneck:

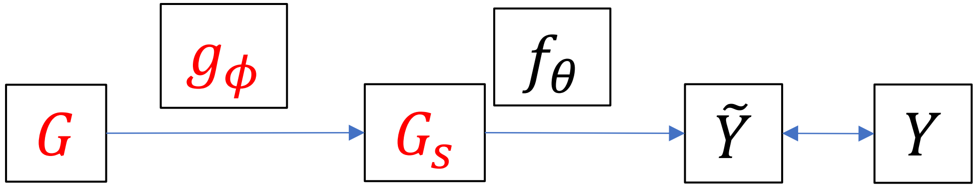
$$\min_{\mathbb{P}(Z|\mathcal{D})\in\Omega} \mathrm{GIB}_\beta(\mathcal{D}, Y; Z) \triangleq [-I(Y; Z) + \beta I(\mathcal{D}; Z)]$$

Figure 1: Graph Information Bottleneck is to optimize the representation Z to capture the minimal sufficient information within the input data $\mathcal{D} = (A, X)$ to predict the target Y. $\mathcal{D}$ includes information from both the graph structure A and node features X. When Z contains irrelevant information from either of these two sides, it overfits the data and is prone to adversarial attacks and model hyperparameter change. Ω defines the search space of the optimal model $\mathbb{P}(Z|\mathcal{D})$. $I(\cdot;\cdot)$ denotes the mutual information [17].

Graph Information Bottleneck. NeurIPS 2020.

The proposed method | extractor





1. obtain the node embeddings (representation)

$$GNN(G) \rightarrow \boldsymbol{H} \in \mathbb{R}^{N \times D}$$

2. obtain the edge embeddings

$$\boldsymbol{H}_{edge} = \{[\boldsymbol{h}_i, \boldsymbol{h}_j] : e_{ij} \in \mathcal{E}\}$$

3. obtain the edge probabilities (importance)

$$\boldsymbol{P}_{edge} = MLP(\boldsymbol{H}_{edge})$$

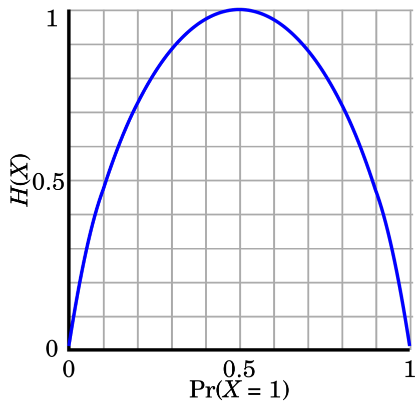
4. obtain the sampled graph G_s with random noise

$$\alpha_{ij} \sim \text{Bernoulli}(\boldsymbol{p}_{ij} + u)$$

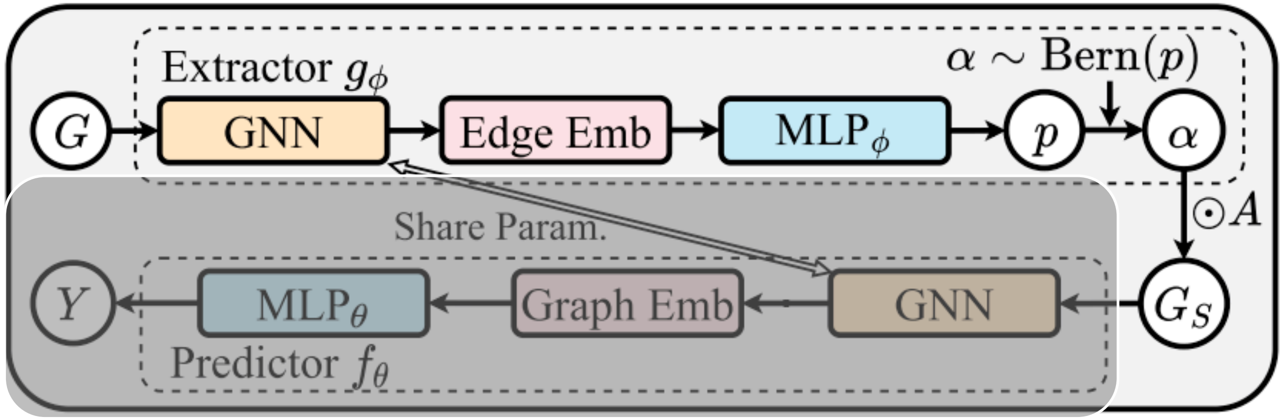
$$A_s = \alpha \odot A \in \mathbb{R}^{N \times N}$$

$$G_s = (A_s, X)$$

The proposed method | extractor



G g_ϕ

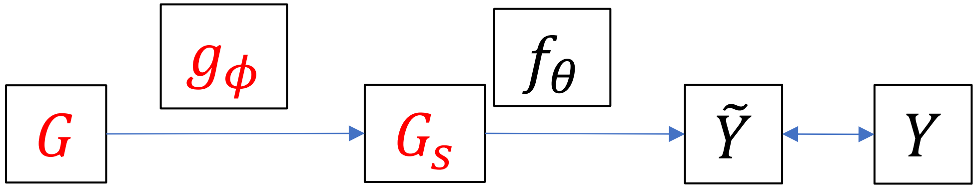


G_s

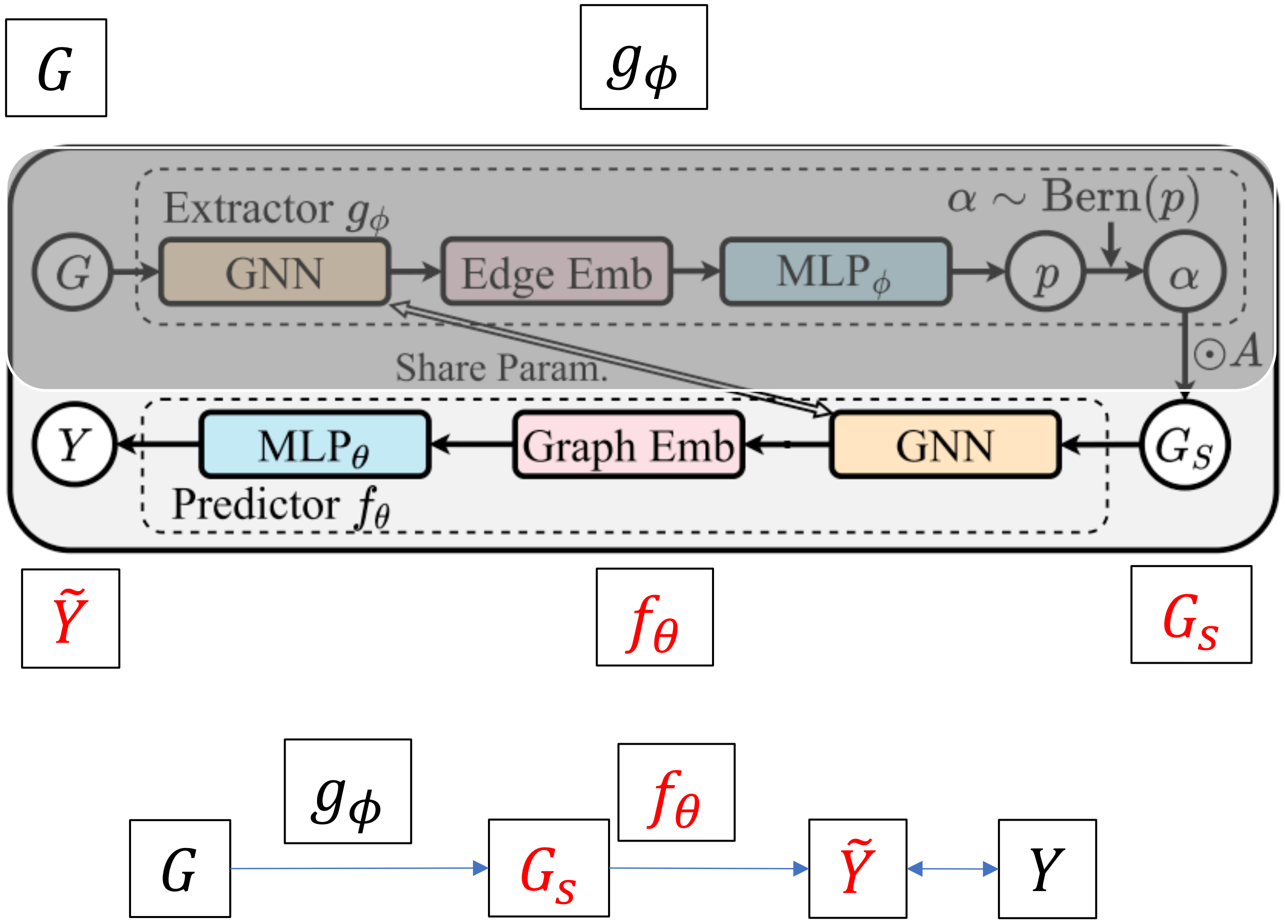
$$\min_{\phi} -I(G_S;Y) + \beta I(G_S;G), \text{ s.t. } G_S \sim g_\phi(G)$$

$$I\left(G_s;G\right) \le \mathbb{E}_G\left[\mathrm{KL}(\mathbb{P}_\phi(G_S|G)||\mathbb{Q}(G_S))\right]$$

$$\mathrm{KL}(\mathbb{P}_\phi(G_S|G)||\mathbb{Q}(G_S)) = \sum_{(u,v)\in E} p_{uv}\log\frac{p_{uv}}{r} + (1-p_{uv})\log\frac{1-p_{uv}}{1-r} + c(n,r). \quad (9)$$



The proposed method | predictor



$$\min_{\phi} -I(G_S;Y) + \beta I(G_S;G), \text{ s.t. } G_S \sim g_\phi(G)$$

$$I\left(G_S;Y\right) \geq \mathbb{E}_{G_S,Y}\left[\log \mathbb{P}_\theta(Y|G_S)\right] + H(Y)$$

classification loss, e.g., cross entropy

Experiment

Table 5. Ablation study on β and stochasticity in GSAT (GIN as the backbone model) on Spurious-Motif. We report both interpretation ROC AUC (top) and prediction accuracy (bottom).

SPURIOUS-MOTIF	$b = 0.5$	$b = 0.7$	$b = 0.9$
GSAT	79.81 ± 3.98	74.07 ± 5.28	71.97 ± 4.41
GSAT-$\beta = 0$	66.00 ± 11.04	65.92 ± 3.28	66.31 ± 6.82
GSAT-NOSTOCH	59.64 ± 5.33	55.78 ± 2.84	55.27 ± 7.49
GSAT-NOSTOCH-$\beta = 0$	63.37 ± 12.33	60.61 ± 10.08	66.19 ± 7.76
GIN	39.87 ± 1.30	39.04 ± 1.62	38.57 ± 2.31
GSAT	51.86 ± 5.51	49.12 ± 3.29	44.22 ± 5.57
GSAT-$\beta = 0$	45.97 ± 8.37	49.67 ± 7.01	49.84 ± 5.45
GSAT-NOSTOCH	40.34 ± 2.77	41.90 ± 3.70	37.98 ± 2.64
GSAT-NOSTOCH-$\beta = 0$	43.41 ± 8.05	45.88 ± 9.54	42.25 ± 9.77

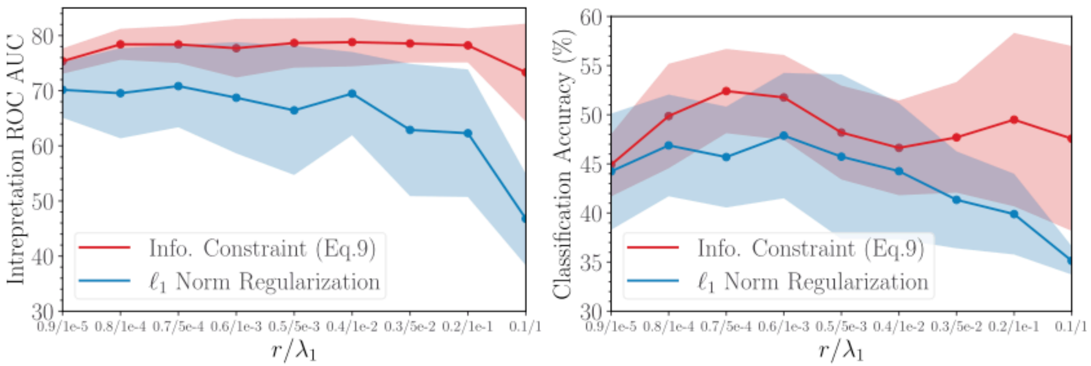


Figure 7. Comparison between (a) using the information constraint in Eq. (9) and (b) replacing it with ℓ_1-norm. Results are shown for Spurious-Motif $b = 0.5$, where r is tuned from 0.9 to 0.1 and the coefficient of the ℓ_1-norm λ_1 is tuned from $1e$-5 to 1.

graph information bottleneck (GIB) 👍🏻
stochasticity (gumbel trick) 👍🏻

Experiment

Table 4. Direct comparison (Acc.) with invariant learning methods on the ability to remove spurious correlations, by applying the backbone model used in (Wu et al., 2022).

SPURIOUS-MOTIF	$b = 0.5$	$b = 0.7$	$b = 0.9$
ERM	39.69 ± 1.73	38.93 ± 1.74	33.61 ± 1.02
V-REx	39.43 ± 2.69	39.08 ± 1.56	34.81 ± 2.04
IRM	41.30 ± 1.28	40.16 ± 1.74	35.12 ± 2.71
DIR	45.50 ± 2.15	43.36 ± 1.64	39.87 ± 0.56
GSAT	$\mathbf{53.27}^{\dagger} \pm 5.12$	$\mathbf{56.50}^{\dagger} \pm 3.96$	$\mathbf{53.11}^{\dagger} \pm 4.64$
GSAT*	43.27 ± 4.58	42.51 ± 5.32	$\mathbf{45.76}^{\dagger} \pm 5.32$

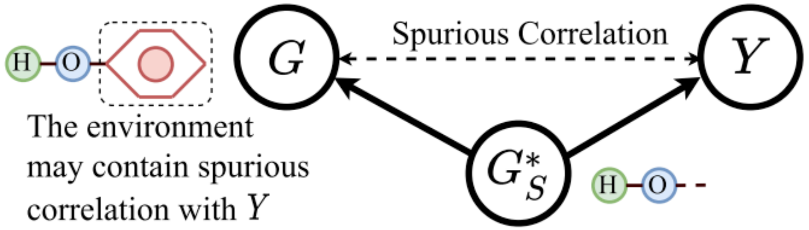


Figure 6. G_S^* determines Y. However, the environment features in $G \backslash G_S^*$ may contain spurious (backdoor) correlation with Y.

Theorem 4.1. Suppose each G contains a subgraph G_S^* such that Y is determined by G_S^* in the sense that $Y = f(G_S^*) + \epsilon$ for some deterministic invertible function f with randomness ϵ that is independent from G. Then, for any $\beta \in [0, 1]$, $G_S = G_S^*$ maximizes the GIB $I(G_S; Y) - \beta I(G_S; G)$, where $G_S \in \mathbb{G}_{\text{sub}}(G)$.

GSAT can remove spurious correlations in the training data 👍🏻

- mainly due to the injecting stochasticity