Explanations of GNN on Evolving Graphs via Axiomatic Layer Edges

Yazheng Liu; Sihong Xie*
The Hong Kong University of Science and Technology (Guangzhou)
Our code is available at https://github.com/yazhengliu/Axiomatic-Layer-Edges

ICLR

Background: GNN on Evolving graphs

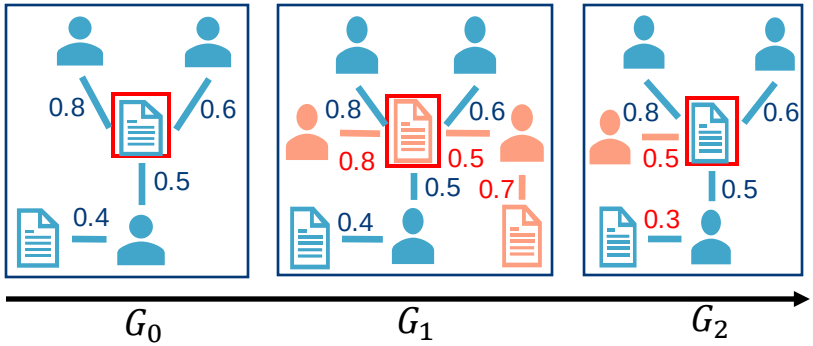
Rumor detection

Good user

Bad user

Non-rumor

Rumor



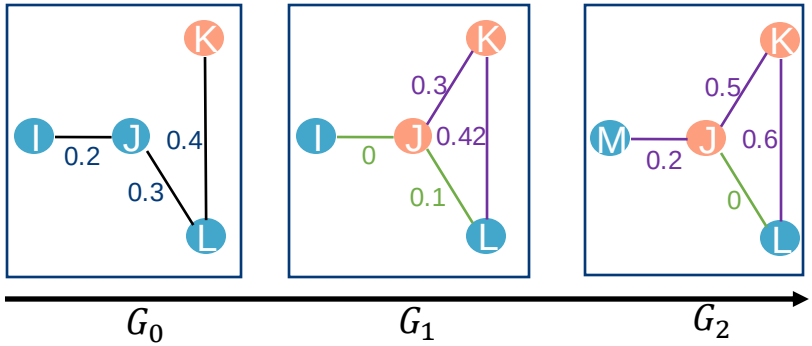
Social Networks

Edge weight increased

Edge weight decreased

Class 1

Class 2



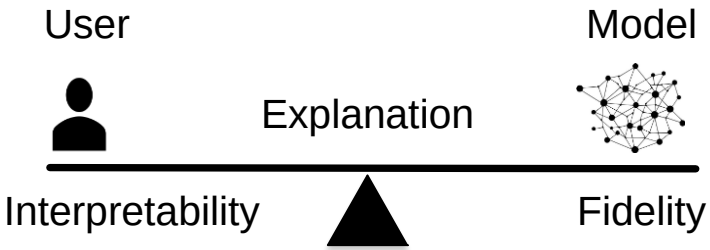
Graphs evolve as edge weights change continuously, leading to model prediction shifts

We aim to select important changed elements to explain model prediction shifts

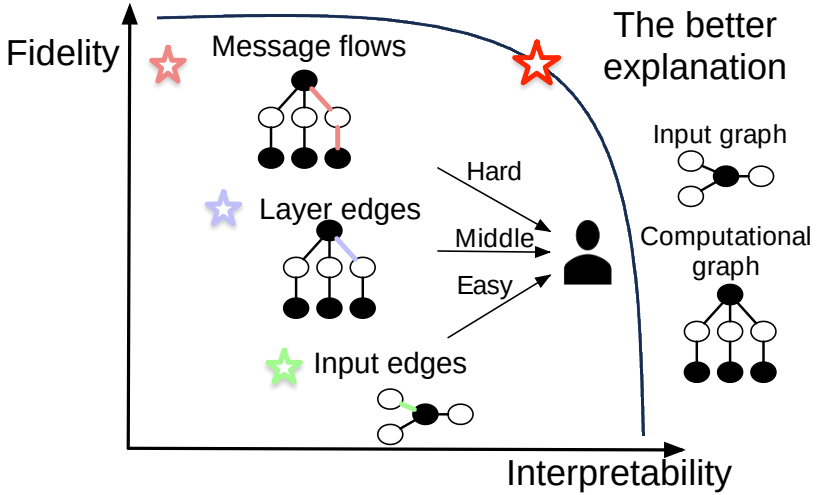
Background: Explainability of GNN

Explanation trade off

- Interpretability: the explanations are easy for users to understand
- Fidelity: the explanations faithfully reflect the model predictions

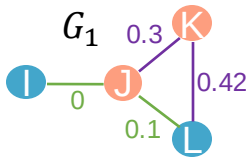


Explanation form



To balance Interpretability and Fidelity, we select important layer edges as explanation

Challenge



Existing method

Calculate contribution

Message flow	Score
K → L → J	[2, 0, 2]
J → L → J	[8, 0, 2]

Average function

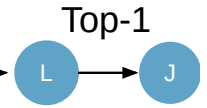
Average function

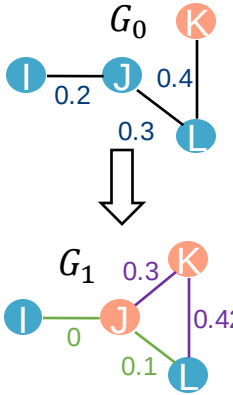
Map contribution

Layer edge	Score
K → L	[1,0,1]
L → J	[1,0,1]
L → J	[4,0,1]
J → L	[4,0,1]

Sum

Select





Challenge

Contribution of $\mathrm{Pr}_J(G_1)$

Not contribution of

$\mathrm{Pr}_J(G_0) \Rightarrow \mathrm{Pr}_J(G_1)$

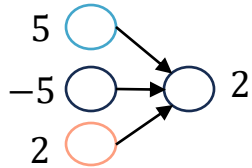
Contribution differently

- Different weights

- Different feature

Unfaithful selection

Top-k

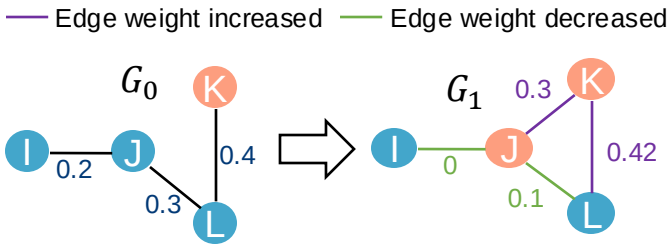


Important node

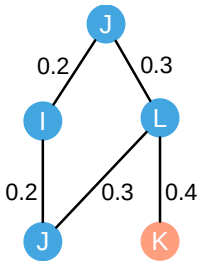
ICLR

Method

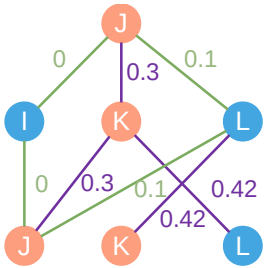
Evolving graphs



Computational graphs



$z_J(G_0) = [1,2,3]$

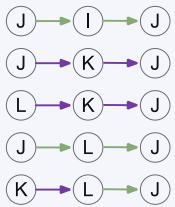


$z_J(G_1) = [3,2,1]$

$Pr_J(G_0) = [0.1,0.24,0.66]$ $Pr_J(G_1) = [0.66,0.24,0.1]$

$\Delta z_J = z_J(G_1) - z_J(G_0) = [2,0,-2]$

(1) Calculate the contribution of message flows $\mathbf{C}$

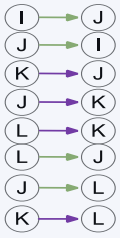


$$\begin{bmatrix} 2 & 0 & 2 \\ 1 & 0 & 0.5 \\ -1 & 0 & -1.5 \\ -0.1 & 0.5 & -2 \\ 0.1 & -0.5 & -1 \end{bmatrix}$$

Summation-to-delta property $\sum_{s=1}^{5} \mathbf{C}_s = \Delta z_J$

Shapley value

(2) Calculate contribution of layer edges $\mathbf{\Phi}$

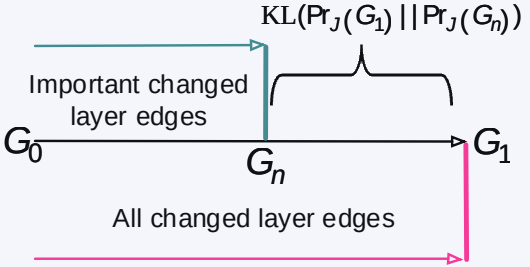


$$\begin{bmatrix} 1.2 & 0 & 1 \\ 0.8 & 0 & 1 \\ -0.5 & 0 & -0.8 \\ 0.7 & 0 & 0.3 \\ -0.2 & 0 & -0.5 \\ -0.02 & 0 & -1.2 \\ 0 & 0.3 & -1.5 \\ 0.02 & -0.3 & -0.3 \end{bmatrix}$$

Summation-to-delta property $\sum_{l=1}^{8} \mathbf{\Phi}_l = \Delta z_J$

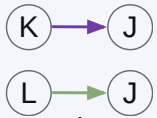
Optimization

(3) Solve objective functions



Select

(4) Final explanations



Explain

$Pr_J(G_0)$ ⇨ $Pr_J(G_1)$

Method: Calculating contributions

Message flow

Let color denote hidden vector of nodes

Contribution decomposition

We want to quantify the contribution of message flow to the change in logits for node J between G_0 and G_1.

The contribution of node L to J - J is 0.1 L - 0.3 L = 0.3 (L - L) + (0.1-0.3) L

The contribution of node K to L - L is 0.42 K - 0.4 K

The contribution of node K to L is 0.42 K

The contribution of message flow is (0.42-0.4)*0.3 K (L - L) + 0.42 *(0.1-0.3) K L

We use DeepLIFT[1] to calculate the contribution $(0.42-0.4)*0.3*\mathbf{h}_K^0(G_0)\frac{\Delta\mathbf{h}_L^1}{\Delta\mathbf{z}_L^1}\theta^1\theta^2 + 0.42*(0.1-0.3)\mathbf{h}_K^0(G_0)\frac{\mathbf{h}_L^1(G_0)}{\mathbf{z}_L^1(G_0)}\theta^1\theta^2$

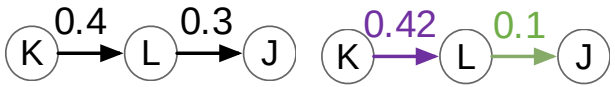
[1] Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences[C]//International conference on machine learning. PMLR, 2017: 3145-3153.

Method: Mapping with Shapley values

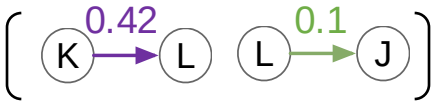
Shapley value

$$\sum_{S \subseteq N \setminus \{i\}} \frac{(|S|!\,(|N|-|S|-1)!)}{|N|!}\left(v(S \cup \{i\}) - v(S)\right)$$

Message flows



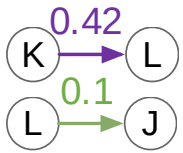
Player sets N



Coalition S	Corresponding message flows	Contributions $v(S)$
{K→0.42→L, L→0.1→J}	K→0.42→L→0.1→J	$[0.1, -0.5, -1]$
{K→0.42→L}	K→0.42→L→0.3→J	$[0.05, -0.2, -0.5]$
{L→0.1→J}	K→0.4→L→0.1→J	$[0.2, -0.1, -0.2]$
$\emptyset$	K→0.4→L→0.3→J	$[0,0,0]$

Efficiency $=$

Shapley value



$$0.5v(\text{K}\xrightarrow{0.42}\text{L}) - 0.5v(\emptyset) + 0.5v(\text{K}\xrightarrow{0.42}\text{L}\quad \text{L}\xrightarrow{0.1}\text{J}) - 0.5v(\text{L}\xrightarrow{0.1}\text{J}) = [-0.025, -0.3, -0.65]$$

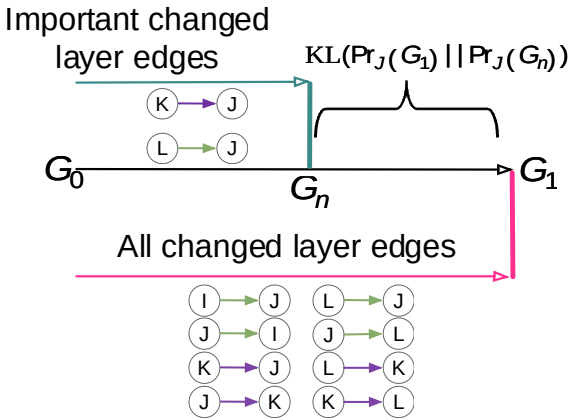
$$+$$

$$0.5v(\text{L}\xrightarrow{0.1}\text{J}) - 0.5v(\emptyset) + 0.5v(\text{K}\xrightarrow{0.42}\text{L}\quad \text{L}\xrightarrow{0.1}\text{J}) - 0.5v(\text{K}\xrightarrow{0.42}\text{L}) = [0.125, -0.2, -0.35]$$

ICLR

Method: Optimization

Optimization principles



KL divergence

$$\mathrm{KL}(\mathrm{Pr}_J(\mathrm{G}_1)||\,\mathrm{Pr}_J(\mathrm{G}_\mathrm{n}))) = \sum_{\mathrm{k}=1}^{\mathrm{c}} \mathrm{Pr}_J(\mathrm{G}_1)[k]\big(\mathrm{z}_J(\mathrm{G}_1)[k] - \mathrm{z}_J(\mathrm{G}_\mathrm{n})[k]\big) - logZ_J(\mathrm{G}_1) + \log Z_J\,(\mathrm{G}_\mathrm{n})$$

where $z_J(G_n) = \sum_{l=1}^{|\Delta\mathcal{A}|} x_l\Phi_l + z_J(G_0)$

Objective function

$$\mathbf{x}^* = \underset{\substack{x\in\{0,1\}^{|\Delta\mathcal{A}|} \\ ||x||_1=m}}{\arg\min} \sum_{k=1}^{c}\left(-\mathrm{Pr}_J(G_1)[k]\sum_{l=1}^{|\Delta\mathcal{A}|} x_l\Phi_{l,k}\right) + \log\sum_{k'=1}^{c}\exp\left(z_J(G_0)[k'] + \sum_{l=1}^{|\Delta\mathcal{A}|} x_l\Phi_{l,k'}\right)$$

- Φ_l: the l-th layer edge contribution
- $\Phi_{l,k}$: the k-th element of the l-th layer edge contribution
- $\Delta\mathcal{A}$: the changed layer edge sets
- x_l: the l-th select variable

- $z_J(G_n)$, $z_J(G_0)$, and $z_J(G_1)$: the logits of node J in G_0, G_1 and G_n
- $\mathrm{z}_J(\mathrm{G}_1)[k]$, $\mathrm{z}_J(\mathrm{G}_\mathrm{n})[k]$, and $\mathrm{Pr}_J(\mathrm{G}_1)[k]$: the k-th element of $\mathrm{z}_J(\mathrm{G}_1)$, $\mathrm{z}_J(\mathrm{G}_\mathrm{n})$ and $\mathrm{Pr}_J(\mathrm{G}_1)$
- $Z_J(\mathrm{G}_1)$: $Z_J(\mathrm{G}_1) = \sum_{k=1}^{c}\exp(Z_J(\mathrm{G}_1)[k])$

Experimental setting

Datasets

- Node classification: Pheme, Weibo, YelpChi, and YelpNYC
- Link prediction: BC-OTC, BC-Alpha, and UCI
- Graph classification: MUTAG, ClinTox, IMDB-BINARY, and REDDIT-BINARY

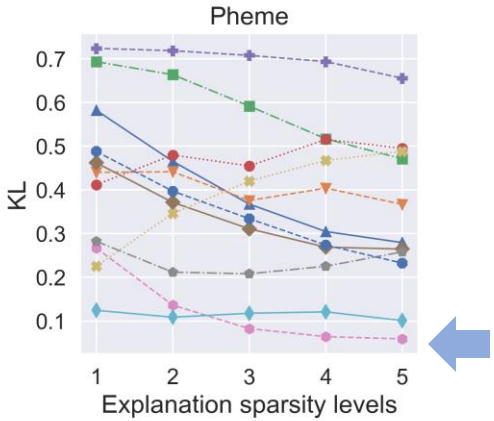
Baselines

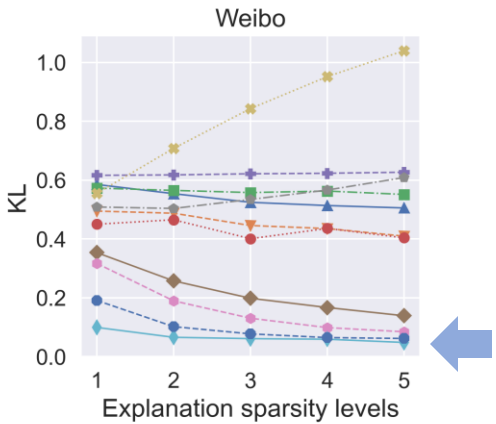
- GNNExplainer
- PGExplainer
- DeepLIFT
- GNN-LRP
- FlowX

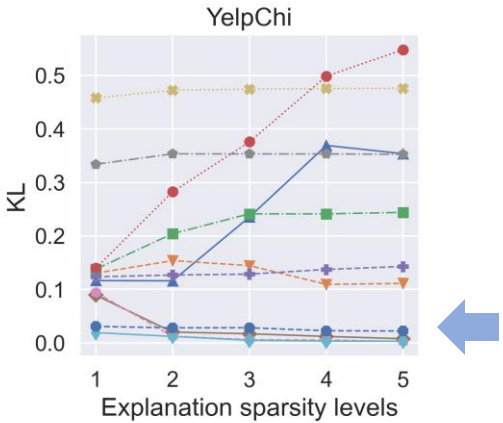
Metrics

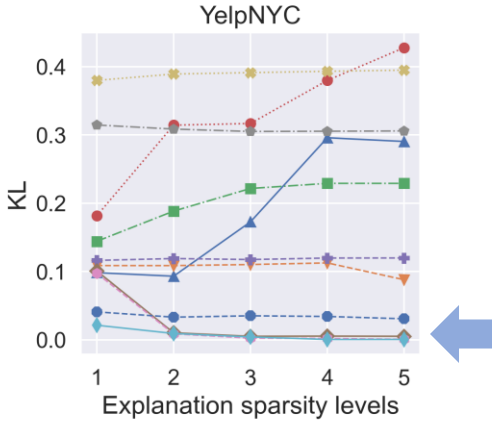
- Fidelity
- Accuracy of explanation

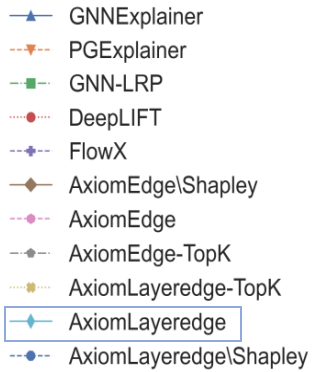
Experiment: Explanation Fidelity







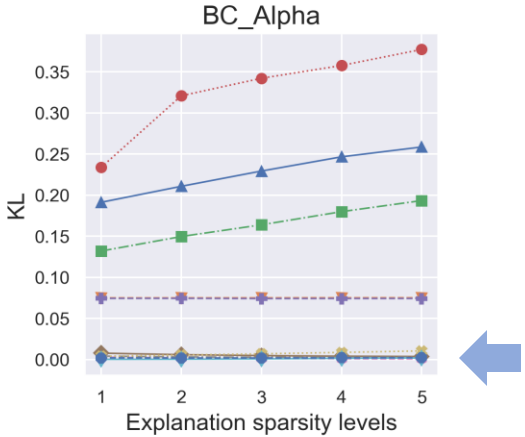


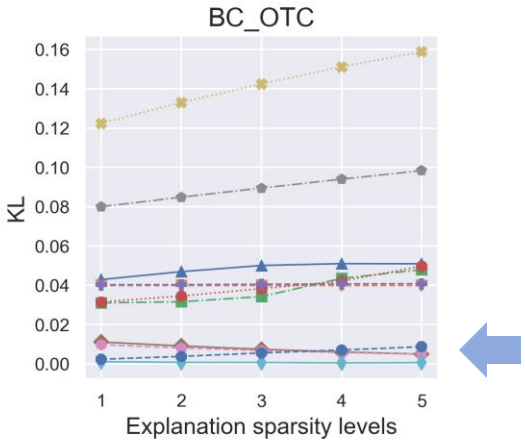


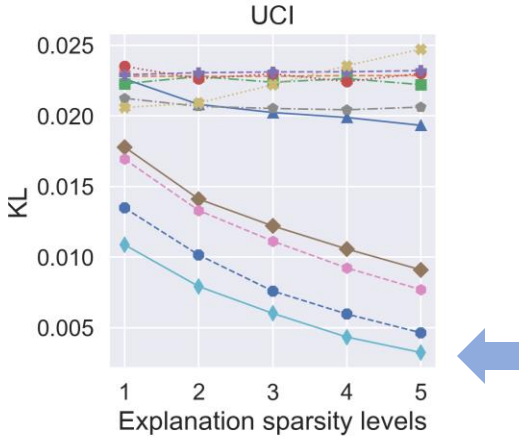
Node classification

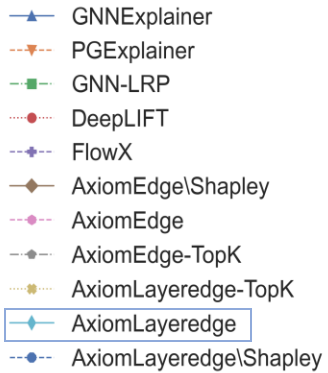
Our method surpasses the baselines in explanation fidelity

Experiment: Explanation Fidelity





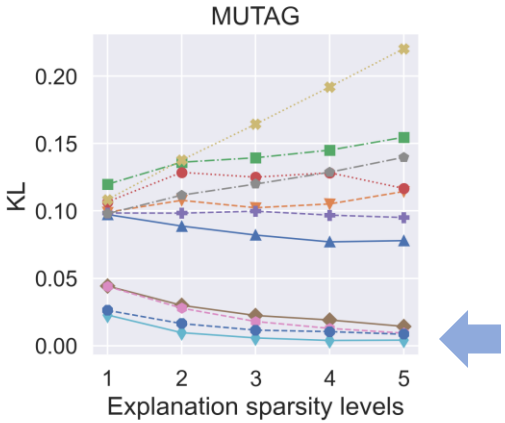


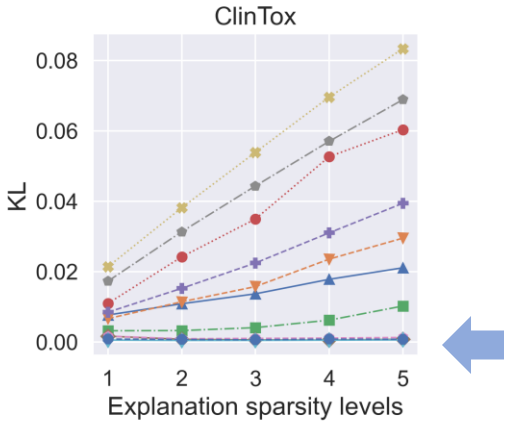


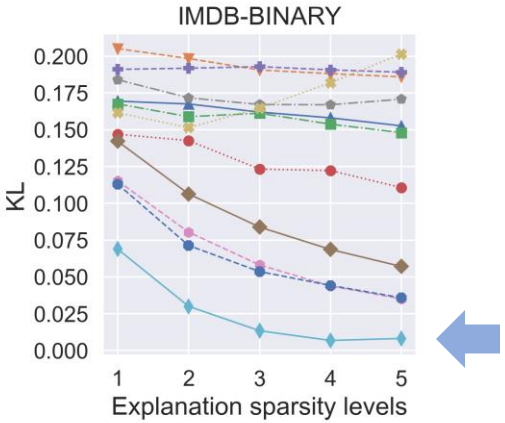
Link prediction

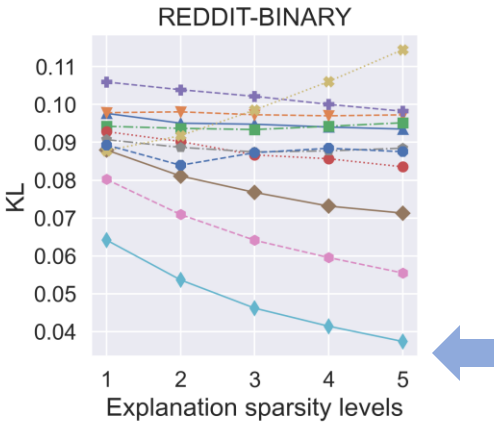
Experiments demonstrate the better faithfulness of the proposed method over the baseline methods

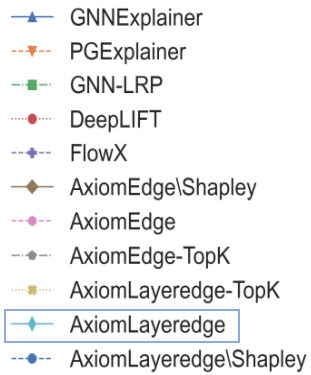
Experiment: Explanation Fidelity







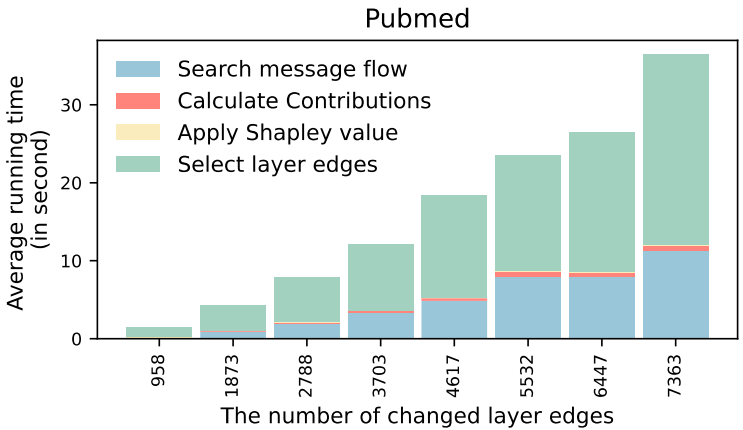


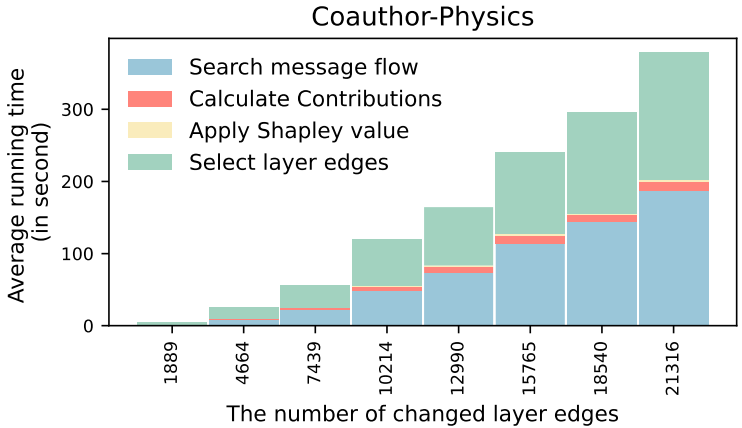


Graph classification

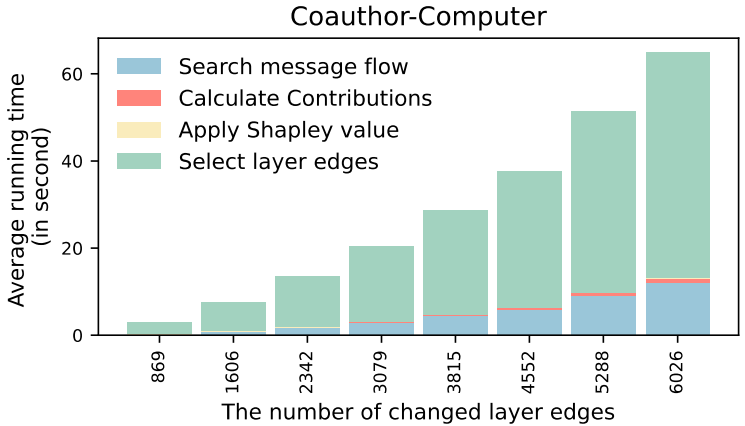
Our approach achieves superior explanation fidelity compared to baselines

Experiment: Running time

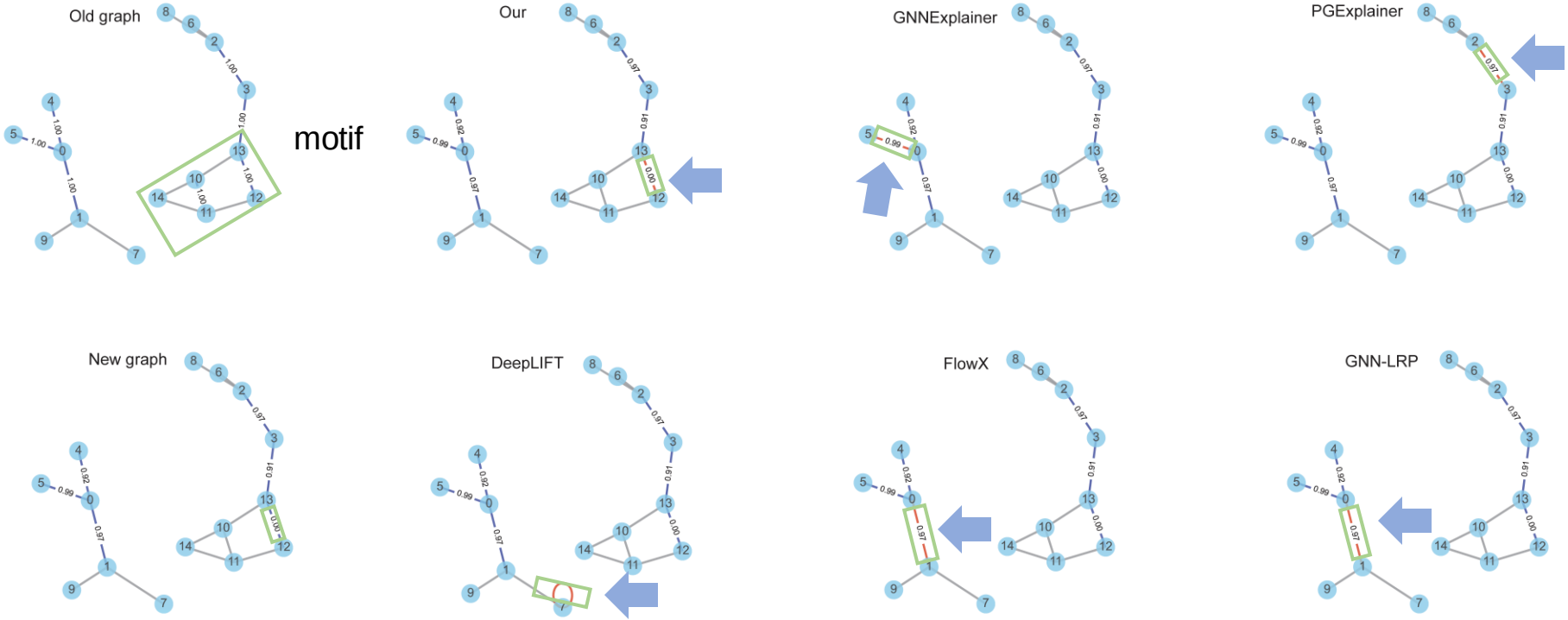




On large graphs, the time for calculating contributions and applying the Shapley value is small. The searching and selecting steps dominate the running time, but the overall time remains manageable.



Experiment: Case study



The edge (12,13) is removed to destroy the motif. The edge (12,13) is the ground truth explanation. Our method correctly identifies (12, 13) as the explanation, while baselines select the wrong edge.

Experiment: Explanation accuracy

On the BA-Shapes dataset

a) we randomly generate graphs with a House motif or a Circle motif.
b) we randomly delete one edge to disrupt the motif and perturbe edges outside the motif area.
c) We train a GNN model to classify the presence of the motif and apply explanation methods to select one edge as explanation.
d) The edge that disrupts the motif is the ground truth explanation.

Datasets	Our	GNNExplainer	PGExplainer	DeepLIFT	GNN-LRP	FlowX
Circle-motif	0.9657	0.9067	0.4765	0.8848	0.1152	0.4156
House-motif	0.9936	0.3618	0.6025	0.9897	0.1755	0.0238

Thanks for listening

ICLR