Recent Progress in Explaining GNNs

Shichang Zhang

Nov 2022

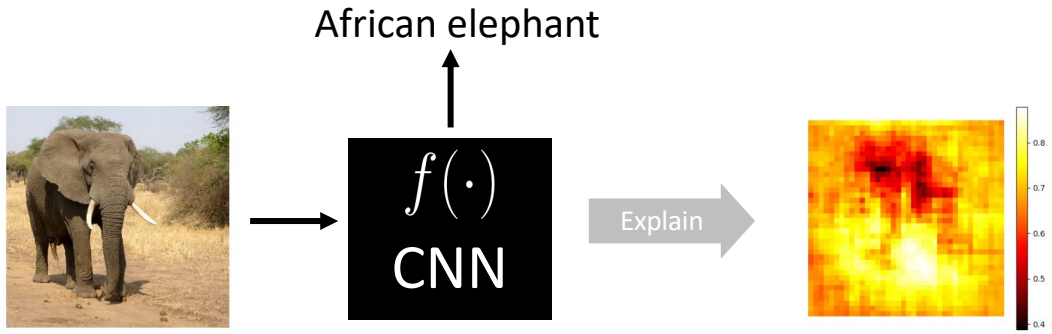
Roadmap

- Introduction
  - Model Explanation
  - GNN Explanation
- Recent progress
  - Towards Multi-Grained Explainability for Graph Neural Networks (NeurIPS 2021)
  - Task-Agnostic Graph Explanations (NeurIPS 2022)

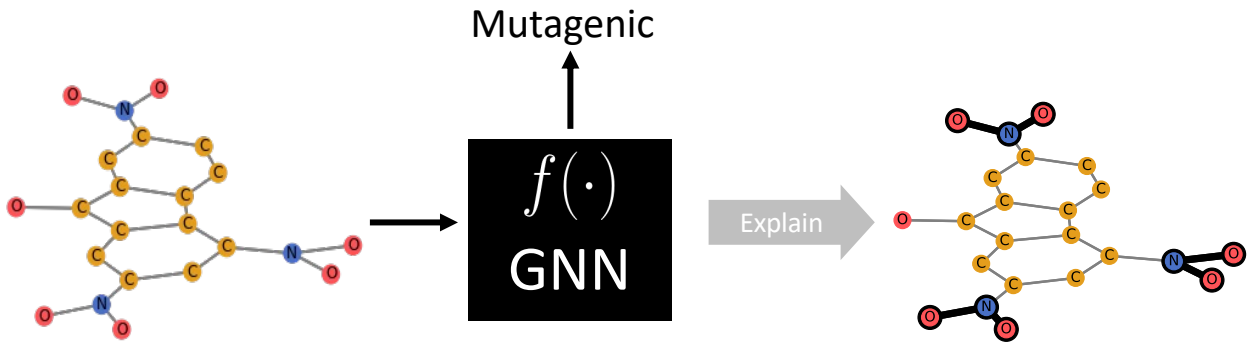
See Shichang Zhang's slides for a paper reading group in Fall 2021 for a more detailed review
https://drive.google.com/file/d/1gQMmQJQdplT5p0kBEwlYc6lSVdiEKfjT/view?usp=sharing

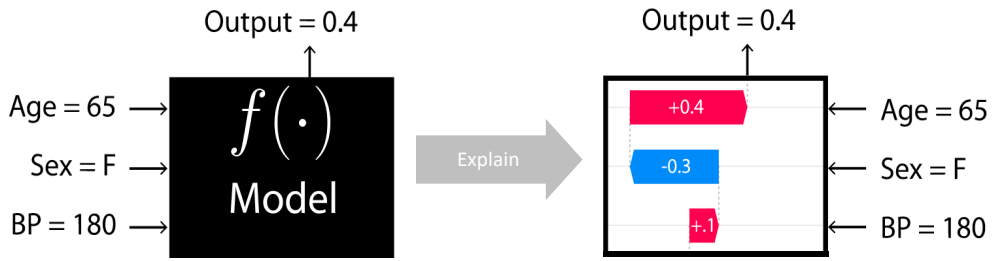
Model Explanation

- Identify importance features for a prediction made by a black-box model
- Improve model transparency and increase user trust

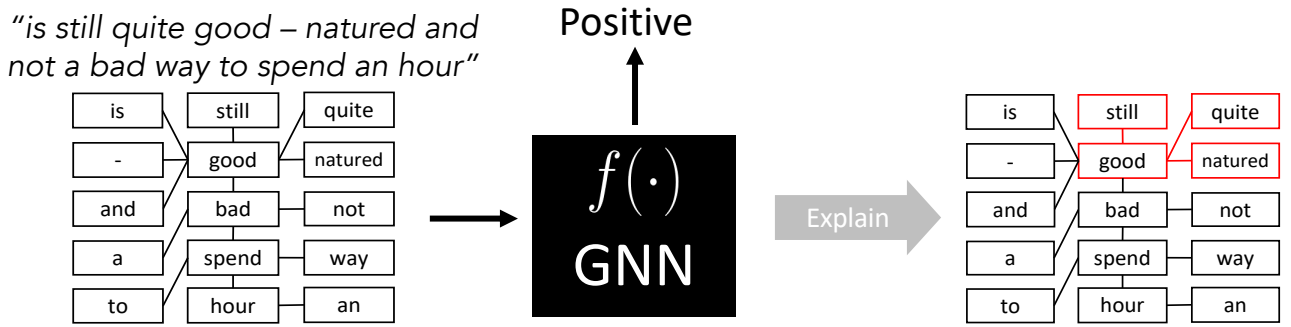


(Zeiler, M. D., & Fergus, R. ECCV 2014)





(Lundberg, S. M., & Lee, S. I. NeurIPS 2017)



GNN Explanation

- Identify important (edge/node-induced) subgraphs for the GNN prediction
- Challenge
  - Discrete graph structure prevents gradient-based methods for CNNs to generalize to GNNs
- Explain via mask learning (GNNExplainer (Ying, et al. NeurIPS 2019))
  - Parameterize a mask over all edges and get a masked subgraph or an edge-induced subgraph

$$G = (\mathcal{V}, \mathcal{E}) \qquad M \in \mathbb{R}^{|\mathcal{E}|} \longrightarrow \mathcal{E}_S = \mathcal{E} \odot \sigma(M) \longrightarrow G_S = (\mathcal{V}_S, \mathcal{E}_S)$$

  - Learn the mask by maximizing the mutual information, which equivalently minimizes the negative probability for the masked graph to make the same prediction as the whole graph

$$\max_{G_S \subseteq G} MI(Y, G_S) = \min_{M} \mathcal{L}_{pred}(M)$$

$$\mathcal{L}_{pred}(M) = -\log P_{\Phi}(Y | \mathcal{E}_S = \mathcal{E} \odot \sigma(M))$$

Towards Multi-Grained Explainability for Graph Neural Networks

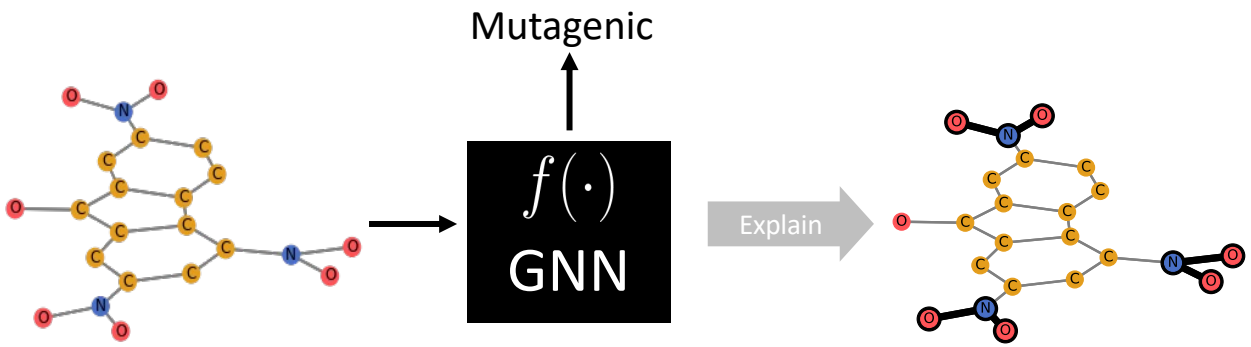
Xiang Wang[§†‡], **Ying-Xin Wu**[§], **An Zhang**[†], **Xiangnan He**[§,*] **Tat-Seng Chua**[†]
[‡]Sea-NExT Joint Lab
[†]National University of Singapore
[§]University of Science and Technology of China
xiangwang@u.nus.edu, wuyxin@mail.ustc.edu.cn, an_zhang@nus.edu.sg
xiangnanhe@gmail.com, dcscts@nus.edu.sg

General Comments

- The question that the paper asks is enlightening
- Whether the proposed approach is the best is questionable
- Some fancy terms that look complicated, but they are not deep

Local Explainability vs. Global Explainability

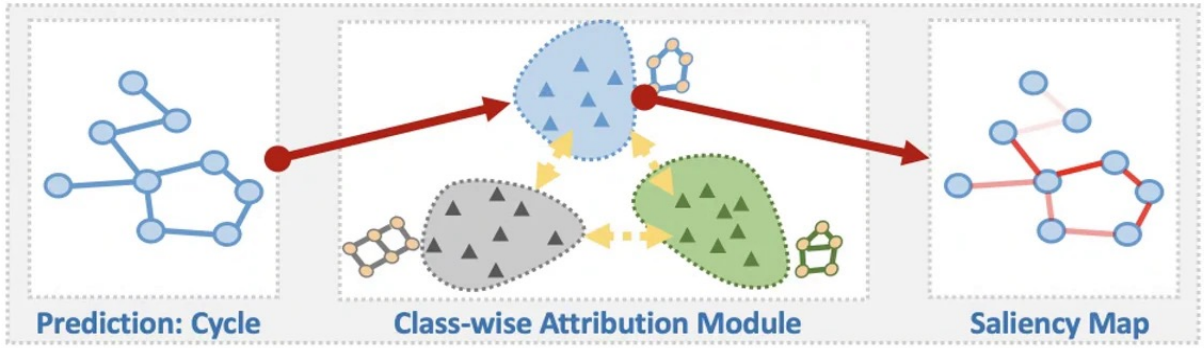
- Local explainability
  - Why the GNN model made the certain prediction for the instance at hand?
- Global explainability
  - What class-wise knowledge does the GNN leverage to make predictions in general?

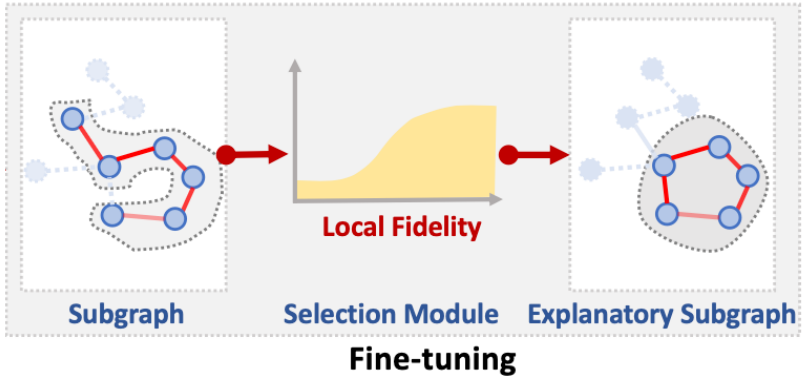


A map of patterns to predictions will be a good global explanation

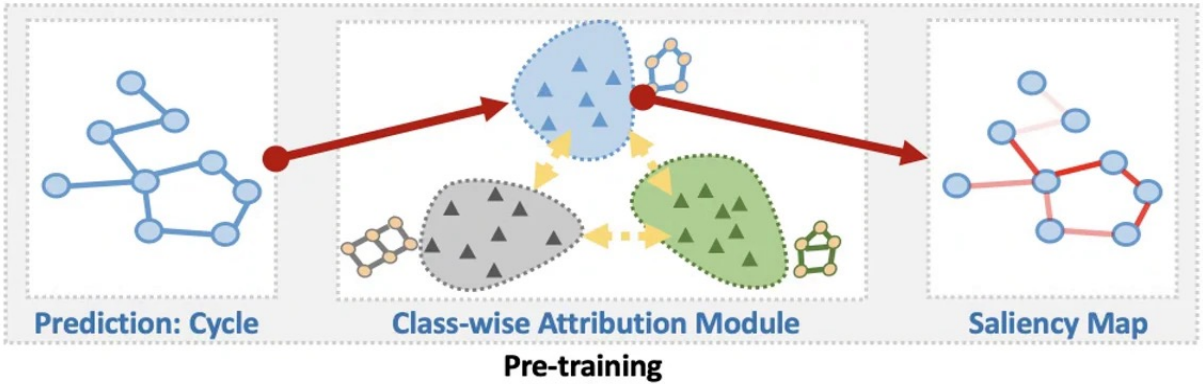
ReFine: Pre-training + Fine-tuning

- Graph $\mathcal{G}$, GNN f, predicted class c, budget ρ
- Saliency map (mask) $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ $\mathbf{M} = \mathcal{T}(\mathcal{G}, f, c)$
- Attentive graph (masked graph) $\mathcal{G}_{att} = \mathbf{A} \odot \mathbf{M}$
- Explanatory subgraph $\mathcal{G}_{exp} = \mathbf{A} \odot \mathbf{S}$ $\mathbf{S} = \mathcal{H}(\mathcal{G}_{att}, f, c, \rho)$





ReFine: Pre-training



$\mathcal{L}_1 = MI(Y, \boldsymbol{M} \odot G_{att})$
Negative mutual information

$\min_\theta \mathcal{L}_1 + \gamma \mathcal{L}_{\text{cts}}$

Contrastive Loss

$$\mathbf{M} = \mathcal{T}(\mathcal{G}, f, c)$$

$$\mathcal{T}_\theta = \{\mathcal{T}^{(c)} | c = 1, \cdots, C\}$$

An attribution module for each class

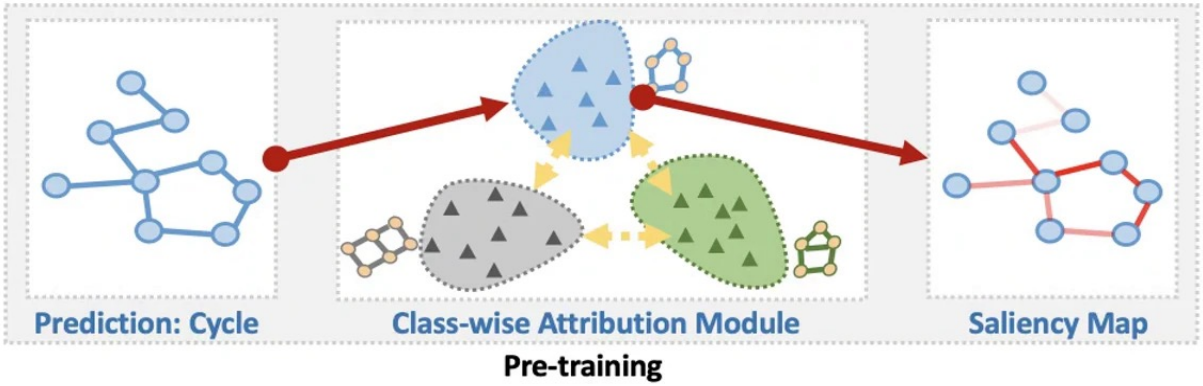
- Node representations + MLP + reparameterization

$$\mathbf{Z} = \text{GNN}(\mathcal{G}, \mathbf{X}) \qquad \alpha_{ij} = \text{MLP}([\mathbf{z}_i, \mathbf{z}_j])$$

$$\epsilon \sim \text{Uniform}(0, 1) \quad P(M_{ij} | \mathbf{z}_i, \mathbf{z}_j) = \sigma((\log \frac{\epsilon}{1-\epsilon} + \alpha_{ij})/\beta)$$

$$\mathcal{L}_1 = -\mathbb{E}_{\mathcal{G}} \mathbb{E}_\epsilon \mathbb{E}_{c'} [P(Y = c' | G = \mathcal{G}) \log P(Y = c' | G = \mathcal{G}_{att}^{(c)})]$$

ReFine: Pre-training



$\mathcal{L}_1 = MI(Y, \boldsymbol{M} \odot G_{att})$
Negative mutual information

$\min_\theta \mathcal{L}_1 + \gamma \mathcal{L}_{\text{cts}}$

Contrastive Loss

$$\mathbf{M} = \mathcal{T}(\mathcal{G}, f, c)$$

$$\mathcal{T}_\theta = \{\mathcal{T}^{(c)} | c = 1, \cdots, C\}$$

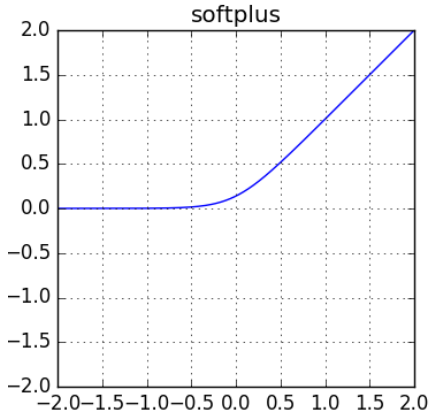
An attribution module for each class

- Similarity maximization/minimization

$$\ell(\mathcal{G}_{att1}^{(c_1)}, \mathcal{G}_{att2}^{(c_2)}) = \mathbf{h}_1^\top \mathbf{h}_2$$

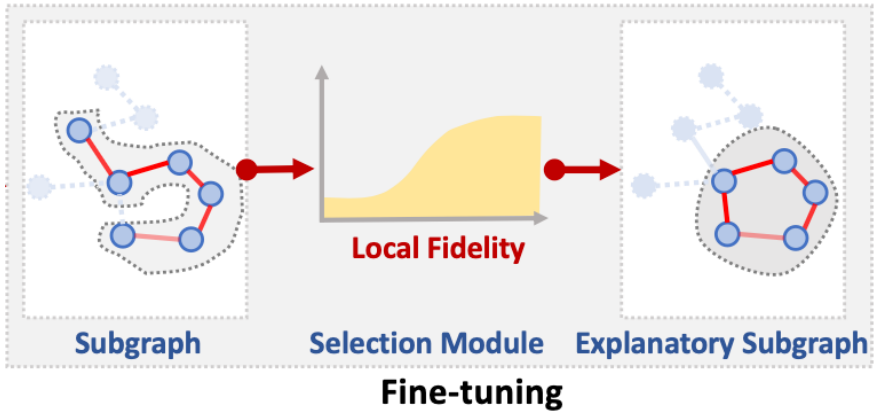
$\mu(x) = \log(1 + \exp(x))$: the softplus function

$$\mathcal{L}_{cts} = \mathbb{E}_{\mathcal{G},\mathcal{G}'} \mathbb{E}_{\epsilon,\epsilon'} [(-1)^{\mathbb{I}(c_1=c_2)} \times \mu(\ell(\mathcal{G}_{att1}^{(c_1)}, \mathcal{G}_{att2}^{(c_2)}))]$$



A continuous approximation of ReLU

ReFine: Fine-tuning



$$\theta_0' = \theta$$

$$\min_{\theta'} \mathcal{L}_2 = MI(Y, G_{exp})$$
$$\text{where } G_{exp} = \mathbf{Top}_{\rho}(G_{att})$$

User-defined ratio

$$\mathbf{S} = \mathcal{H}(\mathcal{G}_{att}, f, c, \rho)$$

A selection module

- The same loss as pre-training, but on the explanatory graph

$$\mathcal{L}_2 = -\mathbb{E}_{\mathcal{G}}\mathbb{E}_{\epsilon}\mathbb{E}_{c'}[P(Y = c'|G = \mathcal{G}) \log P(Y = c'|G = \mathcal{G}_{exp}^{(c)})]$$

The pre-training loss

$$\mathcal{L}_1 = -\mathbb{E}_{\mathcal{G}}\mathbb{E}_{\epsilon}\mathbb{E}_{c'}[P(Y = c'|G = \mathcal{G}) \log P(Y = c'|G = \mathcal{G}_{att}^{(c)})]$$

Experimental Results

- Is global explain (pre-training) + local explain (fine-tuning) effective?
- ACC-AUC: accuracy with different budget ρ, and then compute AUC

	Pre-training		Fine-tuning
	Class-wise Attributors	Contrastive Learning	
PG-Explainer	-	-	-
Refine-CT	✔	-	-
Refine-FT	✔	✔	-
Refine	✔	✔	✔

	Mutagenicity	VG-5	MNIST	BA-3motif	
	ACC-AUC	ACC-AUC	ACC-AUC	ACC-AUC	Recall@5
SA	0.769	0.769	0.559	0.518	0.243
GNNExplainer	0.895±0.010	0.895±0.003	0.535±0.013	0.528±0.005	0.157±0.002
PG-Explainer	0.631±0.008	0.790±0.004	0.504±0.010	0.586±0.004	0.293±0.001
PGM-Explainer	0.714±0.007	0.792±0.001	0.615±0.003	0.575±0.002	0.250±0.000
ReFine-CT	0.888±0.008	0.891±0.002	0.526±0.007	0.610±0.004	0.248±0.001
ReFine-FT	0.945±0.011	0.906±0.002	0.587±0.008	0.616±0.003	0.299±0.002
ReFine	**0.955**±0.005	**0.914**±0.001	**0.636**±0.003	**0.630**±0.006	**0.304**±0.000
Relative Impro.	6.7%	2.1%	3.4%	7.5%	3.8%

It will be more interesting if class patterns can be discovered and visualized.

Task-Agnostic Graph Explanations

Yaochen Xie*
Texas A&M University
College Station, TX
ethanycx@tamu.edu

Sumeet Katariya
Amazon Search
Palo Alto, CA
katsumee@amazon.com

Xianfeng Tang
Amazon Search
Palo Alto, CA
xianft@amazon.com

Edward Huang
Amazon Search
Palo Alto, CA
ewhuang@amazon.com

Nikhil Rao
Amazon Search
Palo Alto, CA
nikhilsr@amazon.com

Karthik Subbian
Amazon Search
Palo Alto, CA
ksubbian@amazon.com

Shuiwang Ji
Texas A&M University
College Station, TX
sji@tamu.edu

General Comments

- Their approach is enlightening
- Whether the task is the most useful is questionable

Motivation

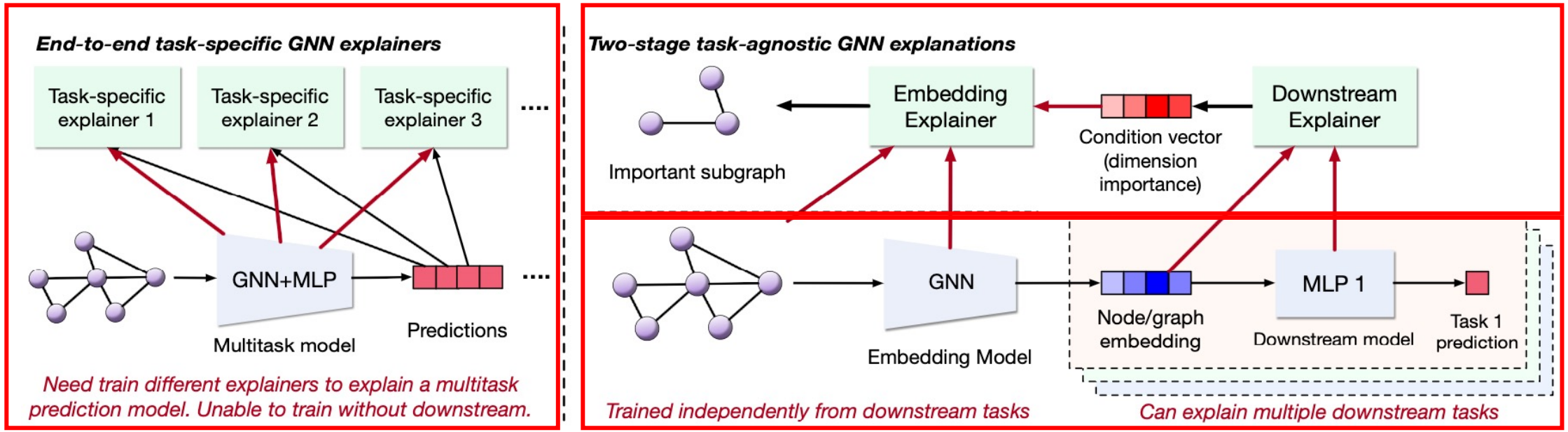
- A single explainer for multiple tasks
- Maximize explanation efficiently in a multitask setting

Task-Agnostic GNN Explainer (TAGE)

- Embedding explainer $\mathcal{T}_{\mathcal{E}} : \mathbb{R}^d \times \mathcal{G} \to \mathcal{G}$
- Downstream explainer $\mathcal{T}_{down} : \mathscr{D} \times \mathbb{R}^d \to \mathbb{R}^d$

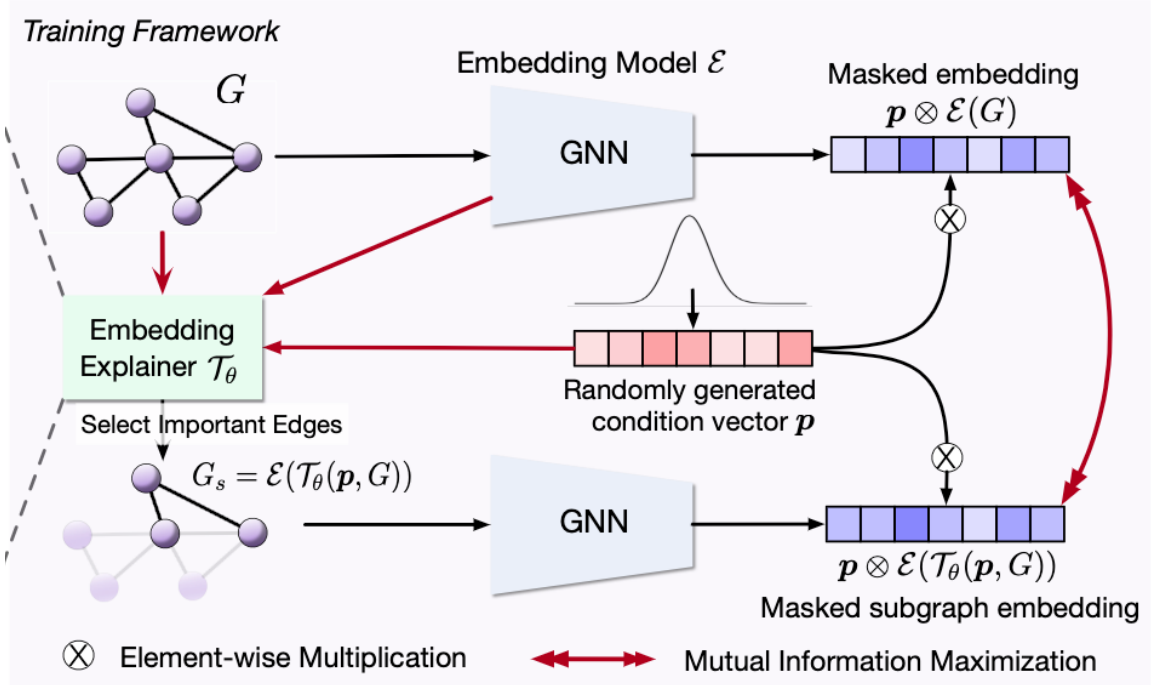
Task-Agnostic GNN Explainer (TAGE)

- Embedding explainer $\mathcal{T}_{\mathcal{E}} : \mathbb{R}^d \times \mathcal{G} \to \mathcal{G}$
- Downstream explainer $\mathcal{T}_{down} : \mathscr{D} \times \mathbb{R}^d \to \mathbb{R}^d$



Training The Embedding Explainer

- The embedding explainer is independent of the downstream task and can be trained in a self-supervised manner



$$\mathcal{T}_\mathcal{E} : \mathbb{R}^d \times \mathcal{G} \to \mathcal{G}$$

Condition vector draw from a Laplace distribution

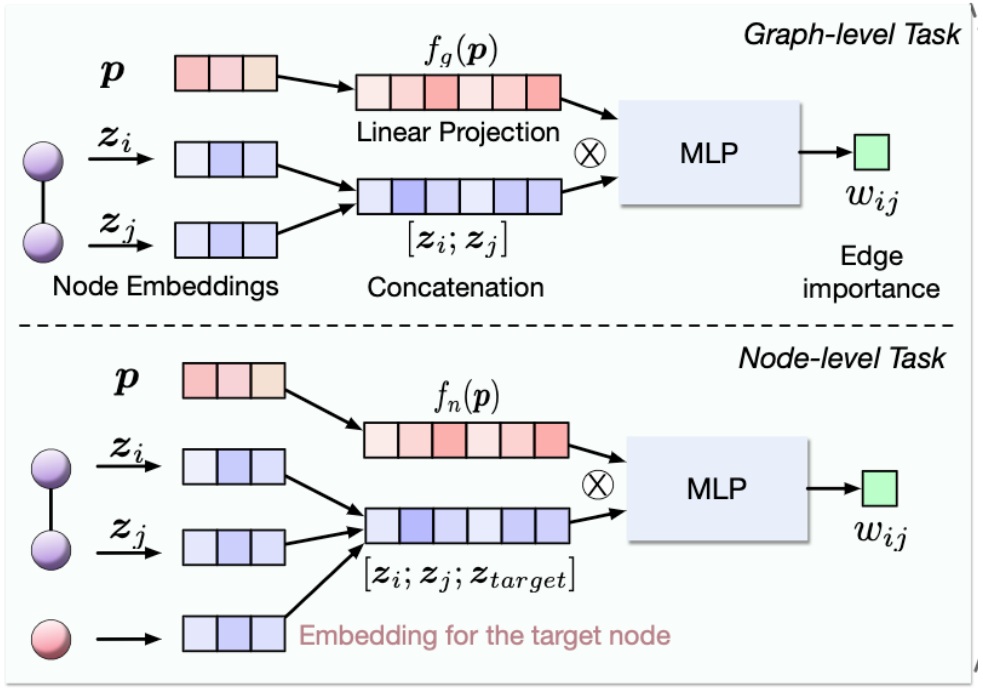
$$\boldsymbol{p} \in \mathbb{R}^d$$

Maximize condition-vector–masked mutual information

$$\max_\theta \mathrm{E}_{\boldsymbol{p}}[\mathbf{MI}(\boldsymbol{p} \otimes \mathcal{E}(G), \boldsymbol{p} \otimes \mathcal{E}(\mathcal{T}_\theta(\boldsymbol{p}, G)))]$$

The Embedding Explainer Architecture

- An MLP that computes a mask weight for each edge
- Add the target node embedding for node classification



$$w_{ij} = \mathbf{MLP}_g\left([\boldsymbol{z}_i; \boldsymbol{z}_j] \otimes \sigma(f_g(\boldsymbol{p}))\right)$$

$$w_{ij} = \mathbf{MLP}_n\left([\boldsymbol{z}_i; \boldsymbol{z}_j; \boldsymbol{z}_{target}] \otimes \sigma(f_n(\boldsymbol{p}))\right)$$

The Downstream Explainer

$$\mathcal{T}_{down} : \mathscr{D} \times \mathbb{R}^d \to \mathbb{R}^d$$

- A standard gradient-based approach

$$\boldsymbol{g} = \frac{\partial \max_{c \leq C} \mathcal{D}(\boldsymbol{z})[c]}{\partial \boldsymbol{z}} \in \mathbb{R}^{1 \times d}$$

probabilities $\mathcal{D}(\boldsymbol{z}) \in [0, 1]^C$ among all C classes

- Normalize gradients to get the condition vector

$$\boldsymbol{p} = \mathrm{ReLU}(\mathrm{norm}(\boldsymbol{g}^T))$$

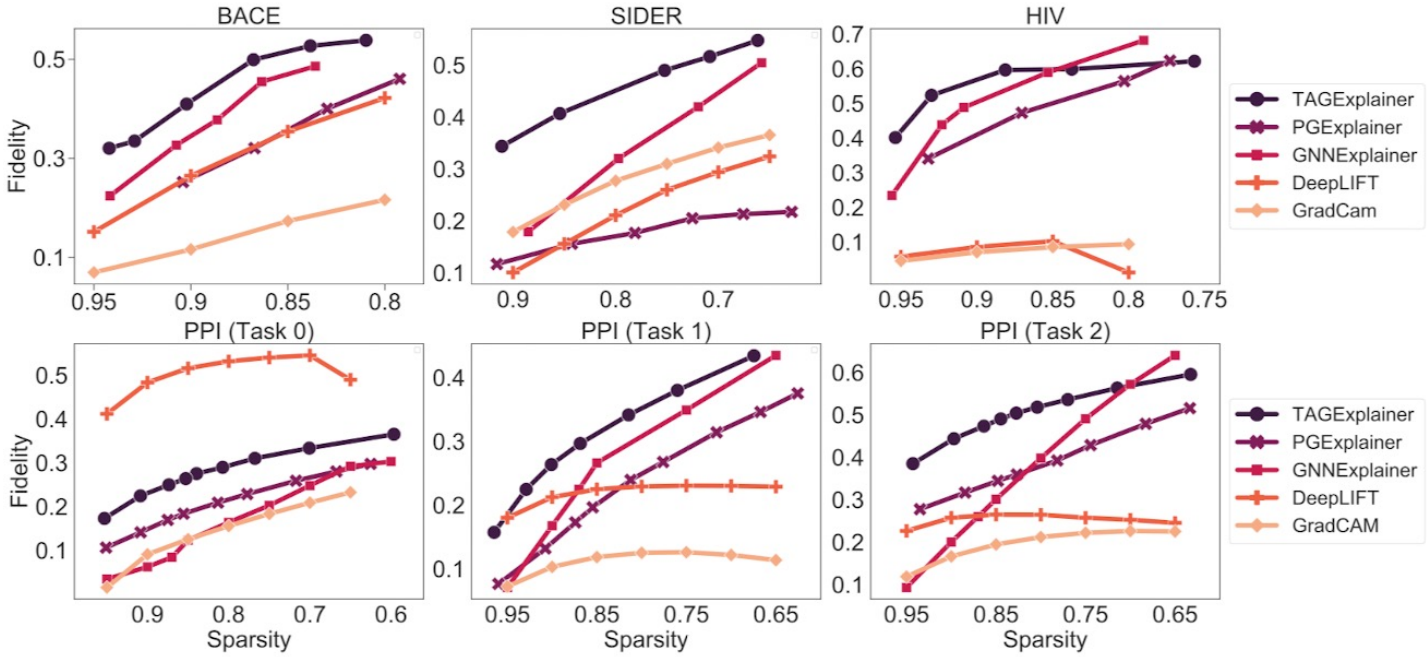
Experimental Results

- Does TAGE explanations have high fidelity?

	MoleculeNet					PPI
	HIV	BBBP	BACE	Sider	Total	
# of Graphs	41127	2039	1513	1427	–	24
Avg. # of Nodes	25.53	24.05	34.12	33.64	–	56,944
Avg. # of Edges	27.48	25.94	36.89	35.36	–	818,716
# of Tasks	1	1	1	27	227	121

$$Fidelity^{prob} = \frac{1}{N}\sum_{i=1}^{N}\left[f(G_i)_{c_i} - f(G_i^{\mathbf{1}-\boldsymbol{m}_i})_{c_i}\right],$$

$$Sparsity = \frac{1}{N}\sum_{i=1}^{N}|\boldsymbol{m}_i|/|V_i|,$$



Experimental Results

- Multi-task generalization comparison

Tables contain fidelity scores at the same sparsity level

Eval on	PGExplainer (trained on) BACE	HIV	BBBP	SIDER	TAGE w/o downstream
BACE	**0.252 ±0.340**	0.007 ±0.251	0.026 ±0.022	-0.151 ±0.330	**0.378 ±0.293**
HIV	-0.001 ±0.197	**0.473 ±0.404**	0.013 ±0.029	-0.060 ±0.356	**0.595 ±0.321**
BBBP	0.001 ±0.237	-0.056 ±0.226	**0.182 ±0.169**	-0.252 ±0.440	**0.193 ±0.161**
SIDER	0.012 ±0.219	-0.009 ±0.212	0.003 ±0.029	**0.444 ±0.391**	**0.521 ±0.278**

Eval on	PGExplainer (trained on) Task 0	Task 1	Task 2	Task 3	Task 4	TAGE w/o downstream
Task 0	**0.184 ±0.3443**	-0.005 ±0.268	0.033 ±0.335	0.034 ±0.310	0.018 ±0.194	**0.271 ±0.385**
Task 1	0.046 ±0.447	**0.197 ±0.380**	0.043 ±0.314	0.008 ±0.297	0.021 ±0.183	**0.300 ±0.415**
Task 2	0.028 ±0.434	0.001 ±0.283	**0.345 ±0.458**	0.024 ±0.320	0.097 ±0.320	**0.499 ±0.480**
Task 3	0.075 ±0.364	-0.015 ±0.219	0.036 ±0.317	**0.262 ±0.418**	0.040 ±0.221	**0.289 ±0.427**
Task 4	0.035 ±0.413	-0.021 ±0.238	0.223 ±0.438	0.075 ±0.374	**0.242 ±0.373**	**0.330 ±0.442**

Experimental Results

- Does it achieve efficient multi-task explanation?

Table 5: Comparison of computational time cost among three learning-based GNN explainers on the PPI dataset. The left two columns record time cost breakdown for T downstream tasks. The fourth column estimates the total time cost for explaining all 121 tasks of PPI. The last row shows the speedup times compared to GNNExplainer and PGExplainer, respectively.

Time cost	Training (s)	Inference (s)	Total time (T=1) (s)	Est. total for 121 tasks
GNNExplainer	20040.1*T	–	20040.1	28 d
PGExplainer	7117.0*T	**427.2*T**	7604.2	10.7 d
TAGE	**1405.3**	582.7*T	**1988.0**	**0.83 d**
Speedup	**14.3*T× / 5.1*T×**	– / 0.73×	**10.1× / 3.8×**	**33.7× / 12.9×**

How practical is the multi-task setting?

Q & A

Reference

- Wang, X., Wu, Y., Zhang, A., He, X., & Chua, T. S. (2021). Towards multi-grained explainability for graph neural networks. *Advances in Neural Information Processing Systems*, *34*, 18446-18458.
- Xie, Y., Katariya, S., Tang, X., Huang, E., Rao, N., Subbian, K., & Ji, S. (2022). Task-Agnostic Graph Explanations. *arXiv preprint arXiv:2202.08335*.